Security Assessment Findings report for testphp.vulnweb.com

Date: 16/03/2024
Project: 001

Table of Contents

Confidentiality Statement

This document is the exclusive property of VulnWeb and Harigovind. This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or in part, in any form, requires consent of both parties.
I 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. I prioritized the assessment to identify the weakest security controls an attacker would exploit. I recommend conducting similar assessments on an annual basis by internal or third-party assessors to ensure the continued success of the controls.

Contact Information

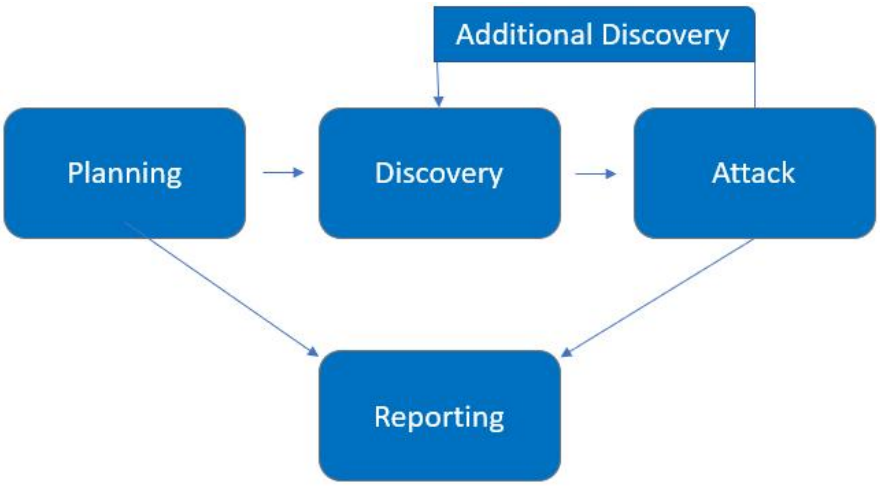
Name	Title	Contact Information
Vulnweb		
John Adams	Information Security (CISO)	Office: (555) 555-5555 Email: john.smith@vulnweb.com
Jim Peter	IT Manager	Office: (555) 555-5555 Email: jim.smith@vulnweb.com
Joe Smith	Network Engineer	Office: (555) 555-5555 Email: joe.smith@vulnweb.com
Pentester		
Harigovind	Pentester	Office: +910987654321 Email: hari@pentest.com

Assessment Overview

From March 11th, 2024 to March 15th, 2024, Vulnweb engaged my company to evaluate the security posture of its infrastructure compared to current industry best practices that included an external penetration test. 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:

- Planning – Customer goals are gathered and rules of engagement obtained.
- Discovery – Perform scanning and enumeration to identify potential vulnerabilities, weak areas, and exploits.
- Attack – Confirm potential vulnerabilities through exploitation and perform additional discovery upon new access.
- Reporting – Document all found vulnerabilities and exploits, failed attempts, and company strengths and weaknesses.



Assessment Components

External Penetration Test

An external penetration test emulates the role of an attacker attempting to gain access to an internal network without internal resources or inside knowledge. I attempted to gather sensitive information through open-source intelligence (OSINT), including employee information, historical breached passwords, and more that can be leveraged against external systems to gain internal network access. I also performed scanning and enumeration to identify potential vulnerabilities in hopes of exploitation.

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	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
High	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
Moderate	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

Scope

Assessment	Details
External penetration test	http://testphp.vulnweb.com

Executive Summary

During March 12 to March 15, a sample Vulnerability test was conducted on the website http://testphp.vulnweb.com. The tools used for scanning are Nuclei, OpenVAS, Burpsuite, Gobuster and pentesting was done on these vulnerabilities to check whether it is valid. By leveraging a series of attacks, I found critical level vulnerabilities that allowed full internal network access to testphp database. It is highly recommended that vulnweb address these vulnerabilities as soon as possible as the vulnerabilities are easily found through basic reconnaissance and exploitable without much effort. Here is a short report of the vulnerability scan.

Total records generated	Critical	High Severity	Medium Severity	Low Severity
9	2	5	2	0

Attack Summary

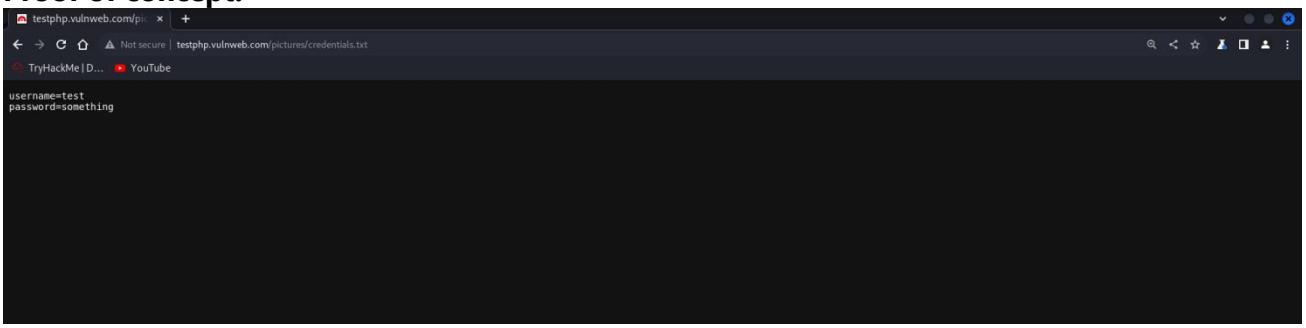
Sl. no	Action	Recommendation	Severity
1	Obtained login credentials from http://testphp.vulnweb.com/pictures/credentials.txt	Delete the webpages which contain user credentials	High
2	The website is vulnerable to clickjacking. It is possible to add the link of the webpage to the <iframe> section	Add http-security-headers	Medium
3	The login page is vulnerable to SQL injection	Implement Input Validation and Sanitization:	Critical
4	The login page is vulnerable to bruteforce attack	Limit the number of login attempts	High
5	Insecure Connection	Change the webpage to “https” from ”http”	Medium
6	Sensitive Information Disclosure	Remove the user credentials	Critical
7	Sensitive Information Disclosure	Remove the webpage contents	High
8	Sensitive Information Disclosure	Implement access controls	High
9	Sensitive Information Disclosure	Implement access controls	High

Penetration test findings

1) Obtained Login credentials (High)

Description	It was possible to find the login credentials of a user from http://testphp.vulnweb.com/pictures/credentials.txt. This user credentials can be later used to login to any webpages associated with it.
Impact	High
System	http://testphp.vulnweb.com/pictures/credentials.txt

Proof of concept:

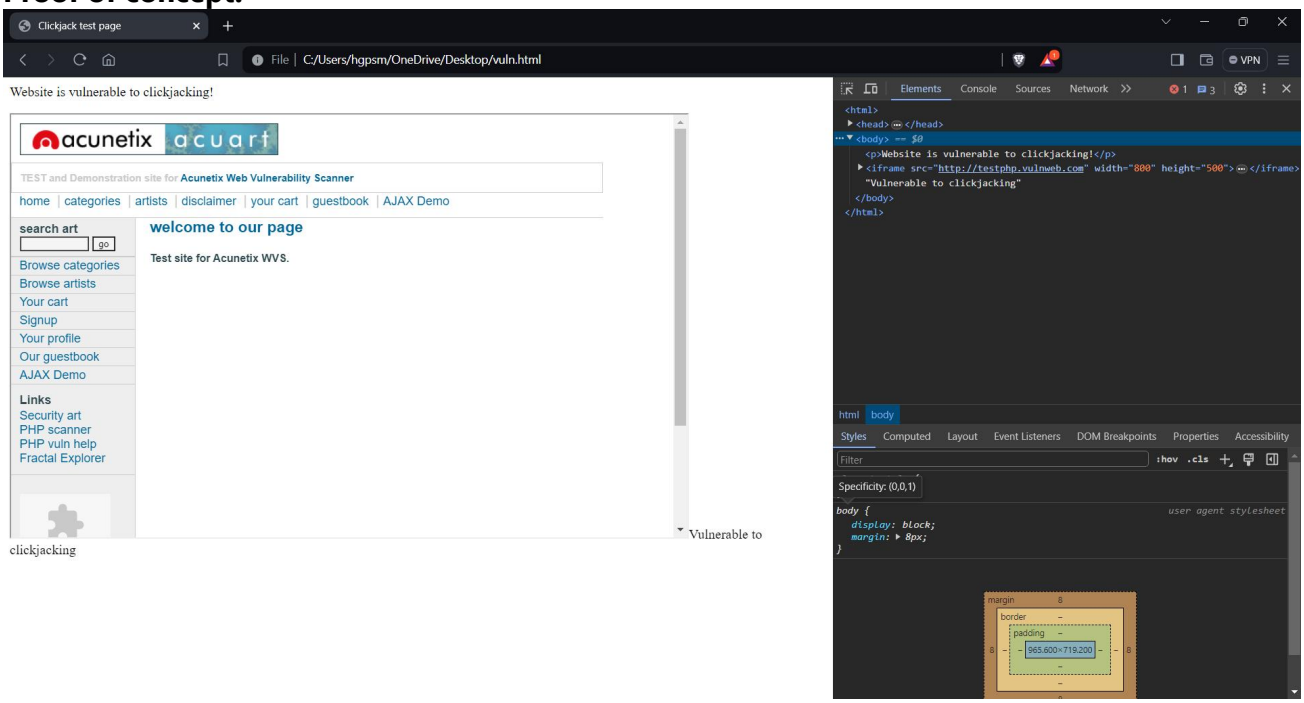


Recommendation:- Remove the webpage or the contents.

2) Clickjacking (Medium)

Description	The website is **Vulnerable to clickjacking**(*an attack that tricks a user into clicking a webpage element which is invisible or disguised as another element*)**.** This can cause users to unwittingly download malware, visit malicious web pages, provide credentials or sensitive information, transfer money or purchase products online. The webpage was added to the "<iframe>" section of a test code and it was able to show the results successfully.
Impact	Medium
System	http://testphp.vulnweb.com/

Proof of concept:



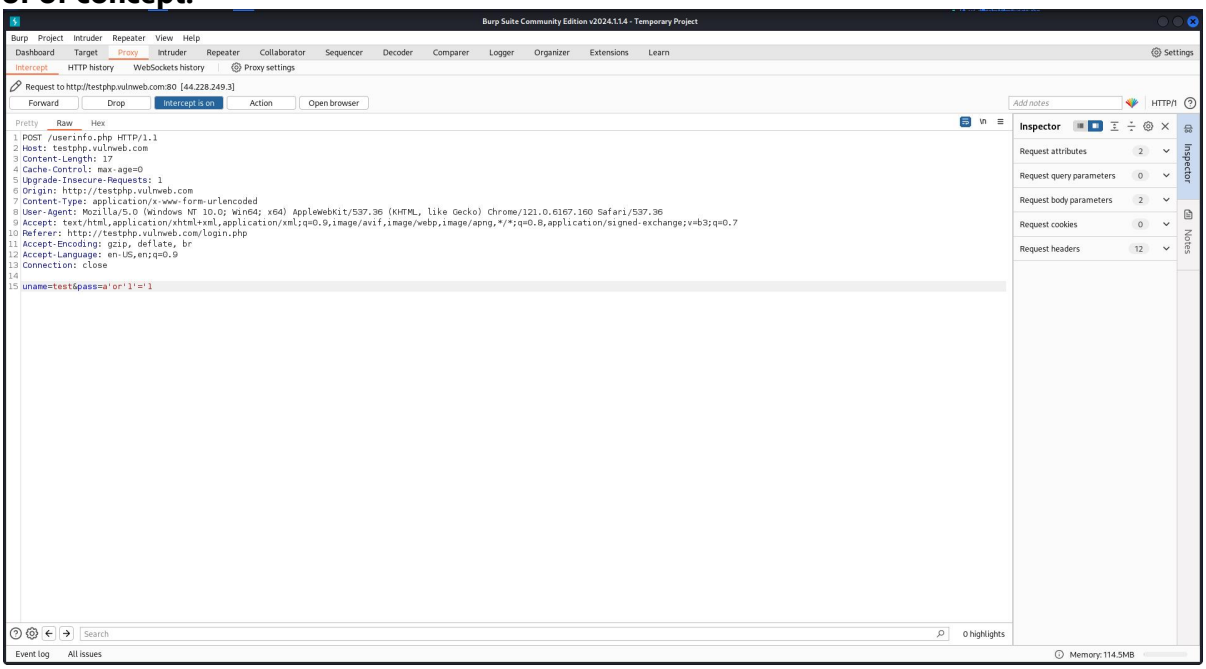
Recommendation:-

Implement X-frame-options header (Used to control whether a page can be placed in an *<iframe>*)
The syntax for adding xframe in Apache:
1) Edit the httpd.conf file.
2) Locate the section for your website or a virtual host.
3) Add the following line inside the section:
Header always set X-Frame-Options "DENY"

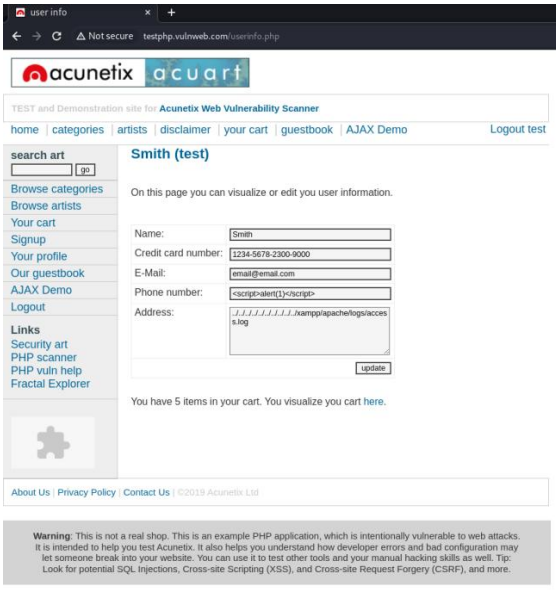
3) SQL injection (Critical)

Description	**The login page is vulnerable to SQL injection**(*a common attack vector that uses malicious SQL code for backend database manipulation to access information that was not intended to be displayed*)**.** It is possible to bypass the login page with the sql command *a' or '1'='1*. When the login page is vulnerable to sql injection other users will be able to login with just usernames and without their password. This can result in unauthorized access to sensitive data, such as: Passwords, credit card number and personal information.
Impact	Critical
System	http://testphp.vulnweb.com/login.php

Proof of concept:



The login request is captured and sql commands are given in the place of password



Login successful

Recommendation:-

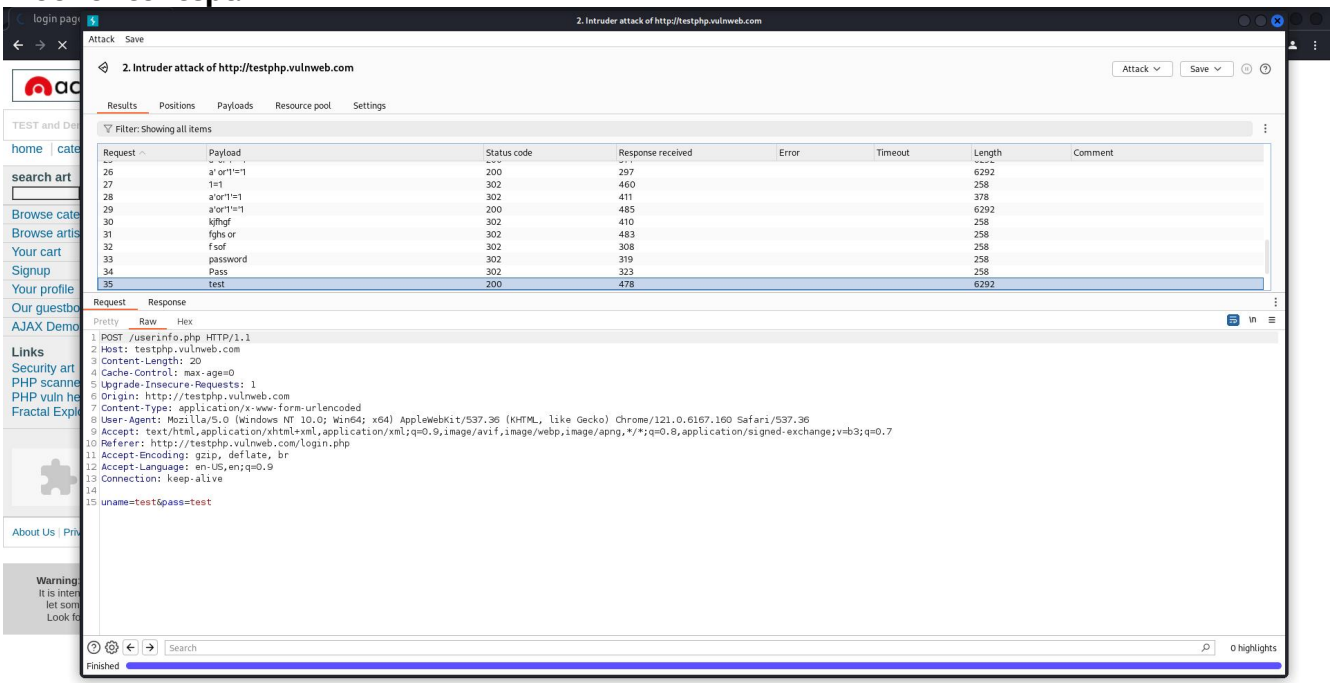
Implement input validation and parameterized queries including prepared statements.

1. Prepared Statements and Parameterization:
   Queries are built with placeholders (?) instead of directly embedding user input.
   User input is provided separately as parameters.
   The database engine safely handles the query, preventing malicious code execution.
2. Input Validation and Sanitization:
   Always validate and sanitize user input:
   Remove or escape special characters that could be used for injection (e.g., quotes, semicolons).
   Enforce allowed data types and formats.
3. Least Privilege Principle:
   Grant database users only the minimum permissions required.
   This minimizes potential damage if an attacker gains unauthorized access.
4. Stored Procedures (cautiously):
   Stored procedures can be secure if implemented carefully.
   Store the complete logic within the database.
   Pass sanitized parameters to the stored procedure.

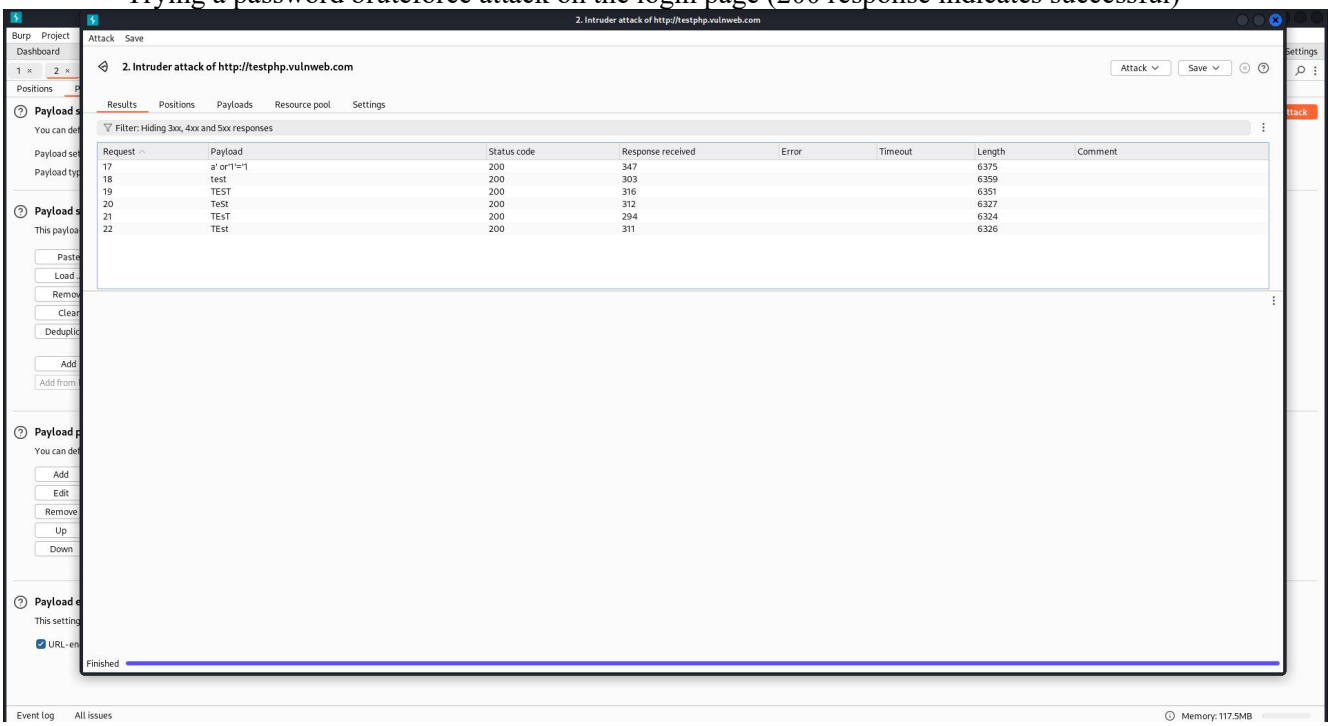
4) The login page is vulnerable to bruteforce attack (High)

Description	The login page is vulnerable to bruteforce attack (*a trial-and-error method used by application programs to decode login information and encryption keys to use them to gain unauthorized access to systems*). It is possible to perform a password bruteforce on the login page.
Impact	High
System	http://testphp.vulnweb.com/login.php

Proof of concept:



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



List of successful passwords

Recommendation:-

Implement number of login limit per ip address. If the number of login attempts from a particular ip address exceeds the limit block the ip address from logging in.

- Track Login Attempts:
- Store the number of failed attempts for a specific user or IP address in a database or temporary storage.
- Enforce Limits:
- Check the number of attempts before allowing another login try.
- Lockouts:
- Implement temporary or permanent lockouts after exceeding the limit.

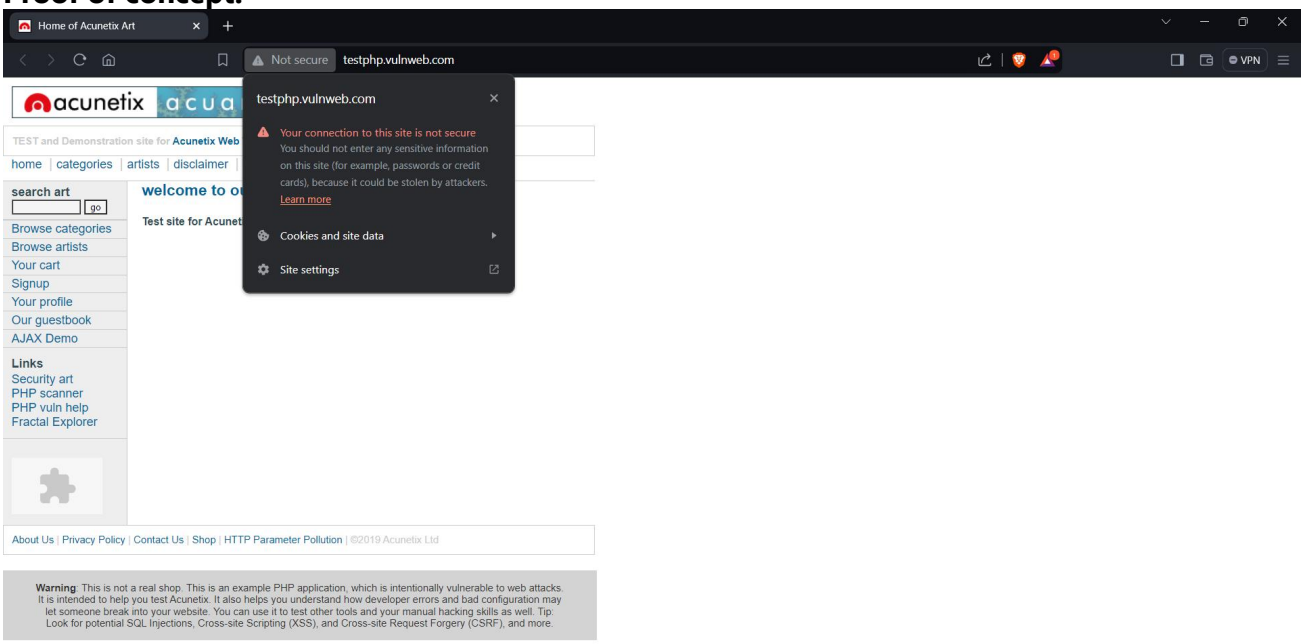
Important Considerations:

- Balance Security and Usability:
  Set a reasonable limit (e.g., 3-5 attempts) to prevent brute-force attacks while not inconveniencing genuine users.
- Lockout Timers:
  Implement temporary lockouts with a timer (e.g., 15 minutes) to allow legitimate users to recover from typos or forgotten passwords.
- Clear Error Messages:
  Inform users about exceeding login attempts and the lockout duration.
- Security Best Practices:
  Never store passwords in plain text. Always use strong hashing algorithms.
- Consider CAPTCHAs:
  These can add an extra layer of protection against automated bots.

5) Insecure connection (Medium)

Description	If a website uses "*http*" instead of "*https*", all requests an responses can be read by anyone who is monitoring the session. "*http*" messages are plaintext, which means unauthorized parties can easily access and read them over the internet. In contrast, "*https*" transmits all data in encrypted form.
Impact	Medium
System	http://testphp.vulnweb.com/

Proof of concept:



Recommendation:-

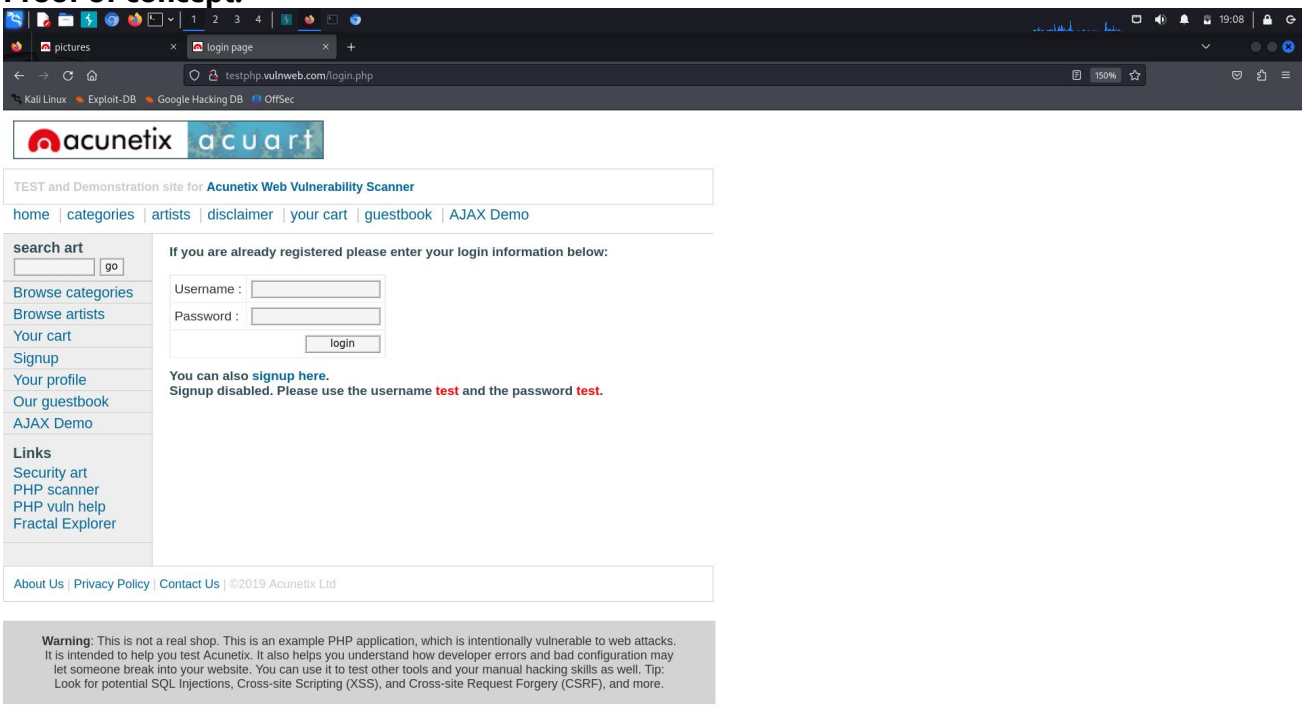
Change the webpage to "*https*".

1. Obtain an SSL Certificate: Purchase one from your hosting provider or a trusted CA.
2. Install and Configure SSL: Follow your hosting provider's instructions for SSL installation on your web server.
3. Enable HTTPS Redirection: Configure your server to redirect all HTTP traffic to HTTPS.
4. Update Internal Links: Modify all internal links within your website code to use HTTPS URLs.

6) Sensitive Information Disclosure(Critical)

Description	The login credentials are visible to other users with username=”test” password=”test”. It is visible to everyone who visits the signup page.
Impact	Critical
System	http://testphp.vulnweb.com/login.php

Proof of concept:

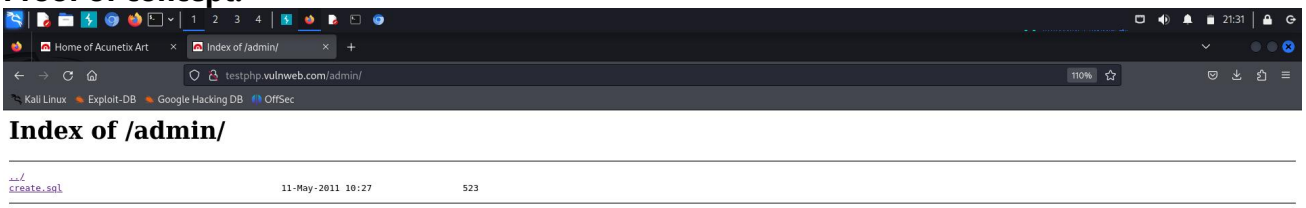


Recommendation:- Remove the user credentials
Ask the user to reset the password

7) Sensitive Information Disclosure(High)

Description	The contents of the admin directory are visible. This file contains sensitive information about the org.
Impact	High
System	http://testphp.vulnweb.com/admin

Proof of concept:

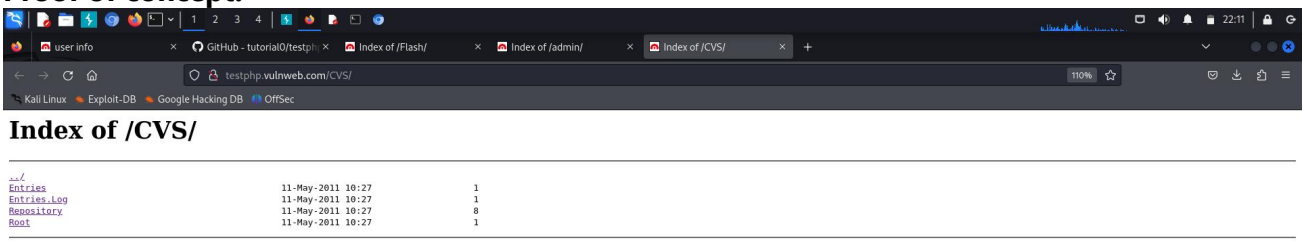


Recommendation:- Hide Sensitive contents from this web page.
Implement access controls.

8) Sensitive Information Disclosure(High)

Description	The contents of the CVS directory are visible. This file contains log history and other sensitive information about the org.
Impact	High
System	http://testphp.vulnweb.com/CVS

Proof of concept:

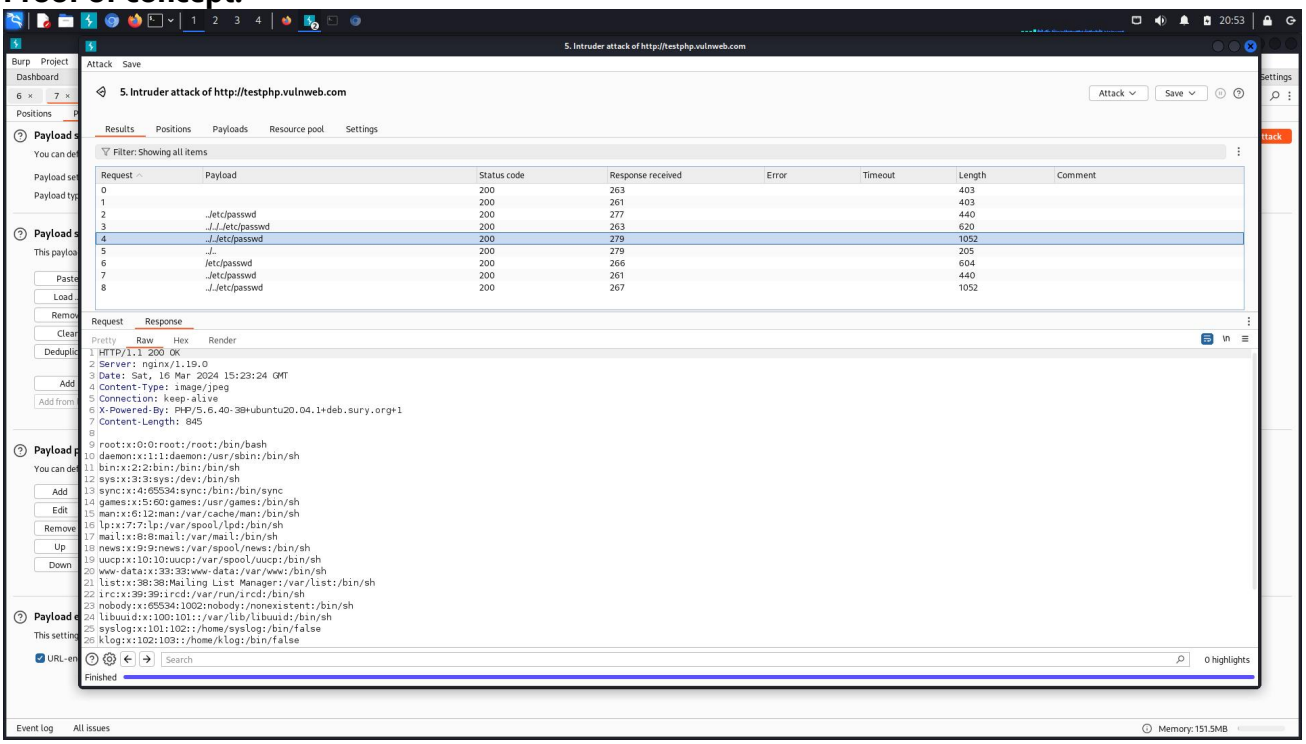


Recommendation:- Hide Sensitive contents from this web page.
Implement access controls

9) Sensitive Information Disclosure(High)

Description	It is possible obtain the login credentials of users from http://testphp.vulnweb.com/showimage.php?file=./pictures/1.jpg by capturing the packets and using the wordlist *../../etc/passwd*
Impact	High
System	http://testphp.vulnweb.com/showimage.php?file=./pictures/1.jpg

Proof of concept:

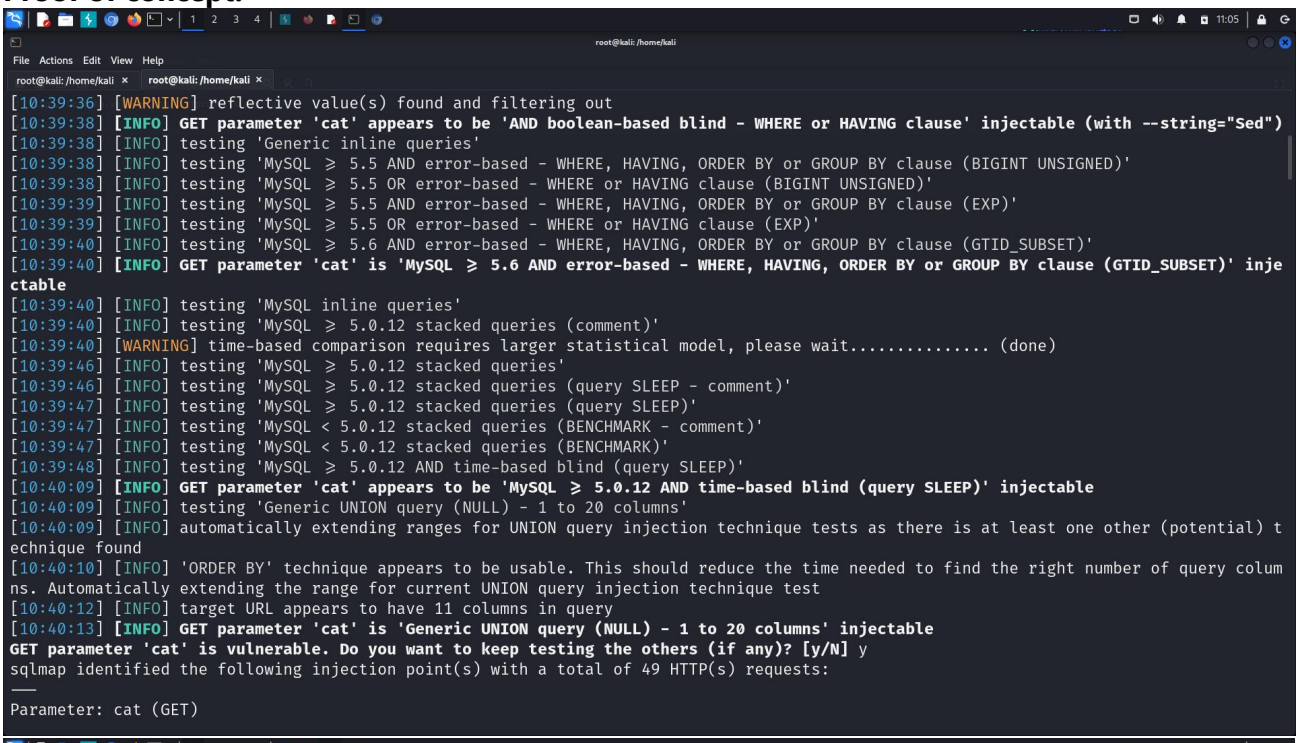


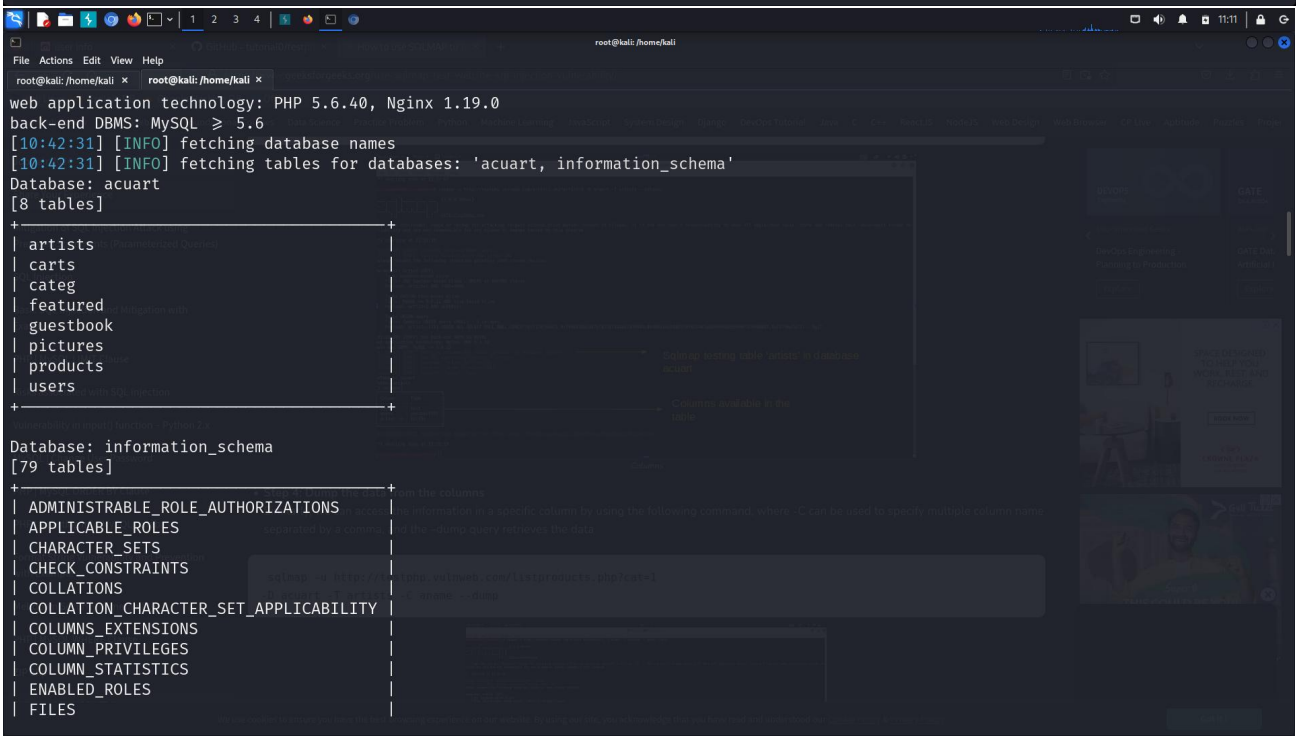
Recommendation:- Hide Sensitive contents from this web page or delete the web page. Implement access controls

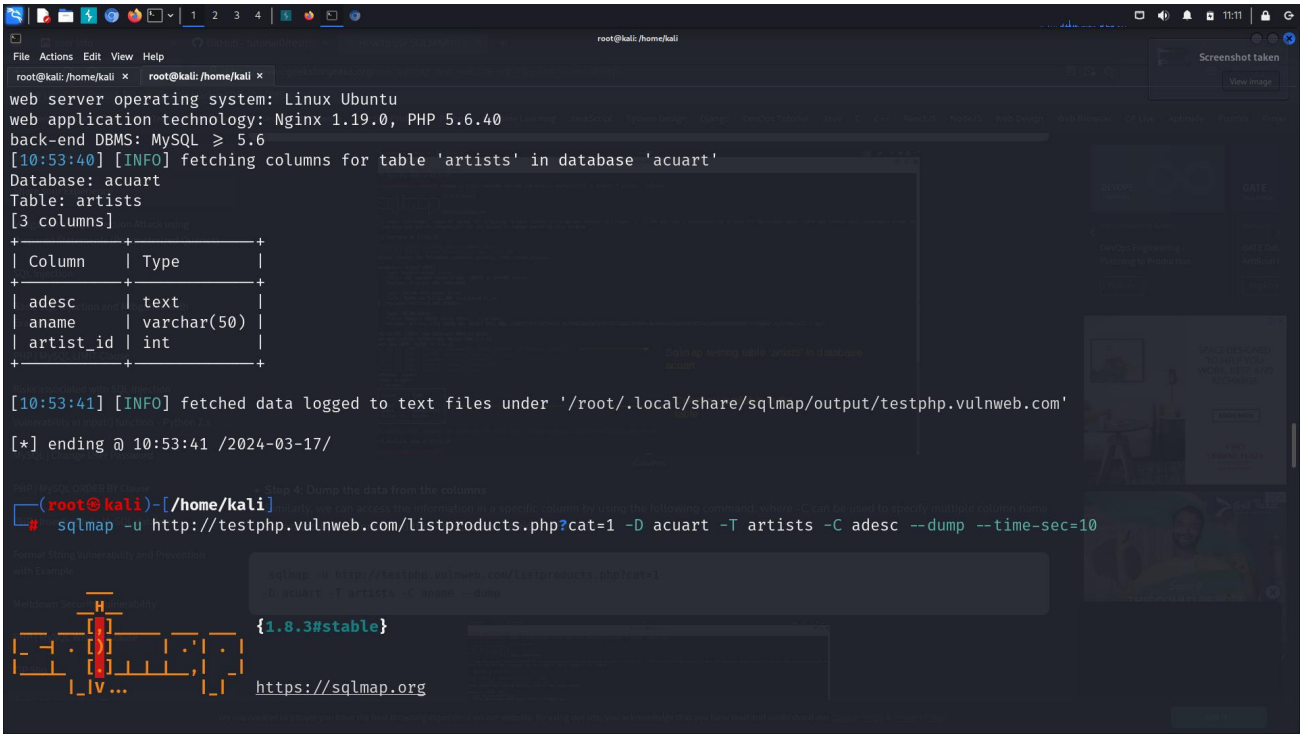
10) SQL injection(Critical)

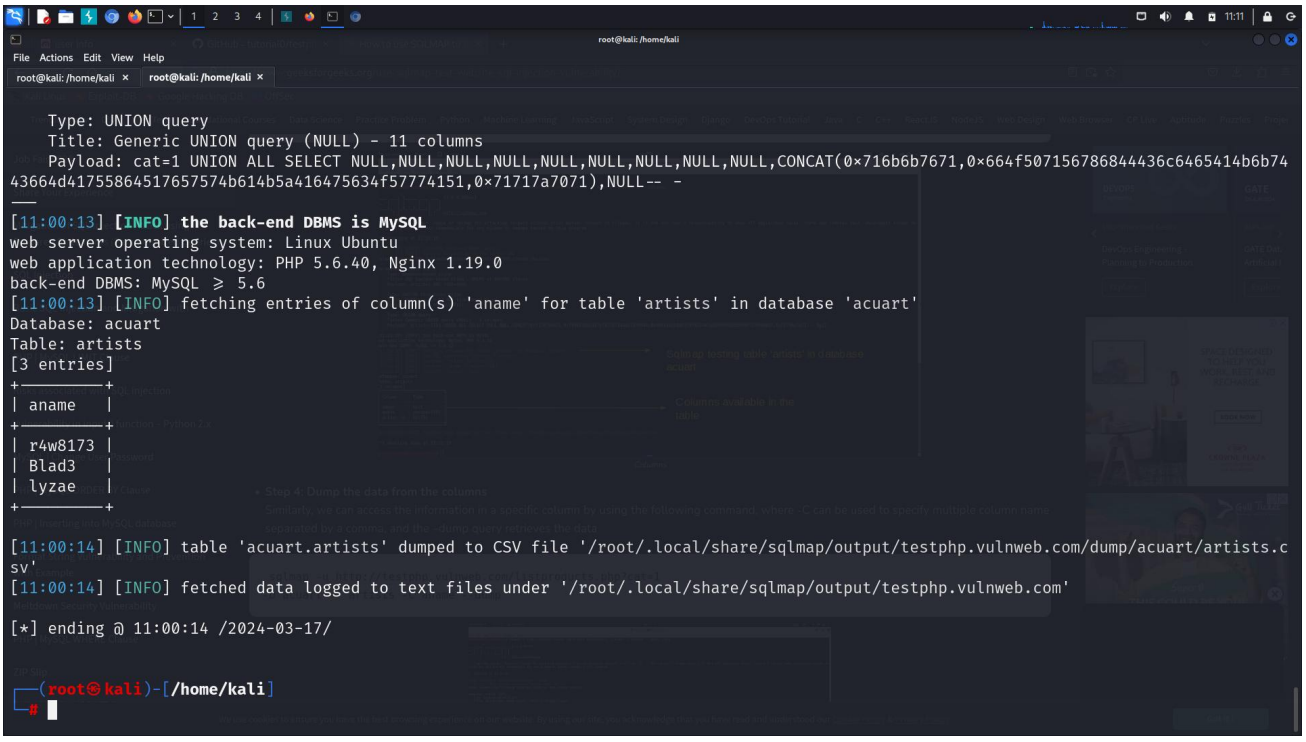
Description	The webpage http://testphp.vulnweb.com/product.php?pic=1 is vulnerable to SQL injection. It is possible to view all the contents of other pages. It displays "*artist ids*", "*artist names*" and much more information.
Impact	High
System	http://testphp.vulnweb.com/product.php?pic=1

Proof of concept:









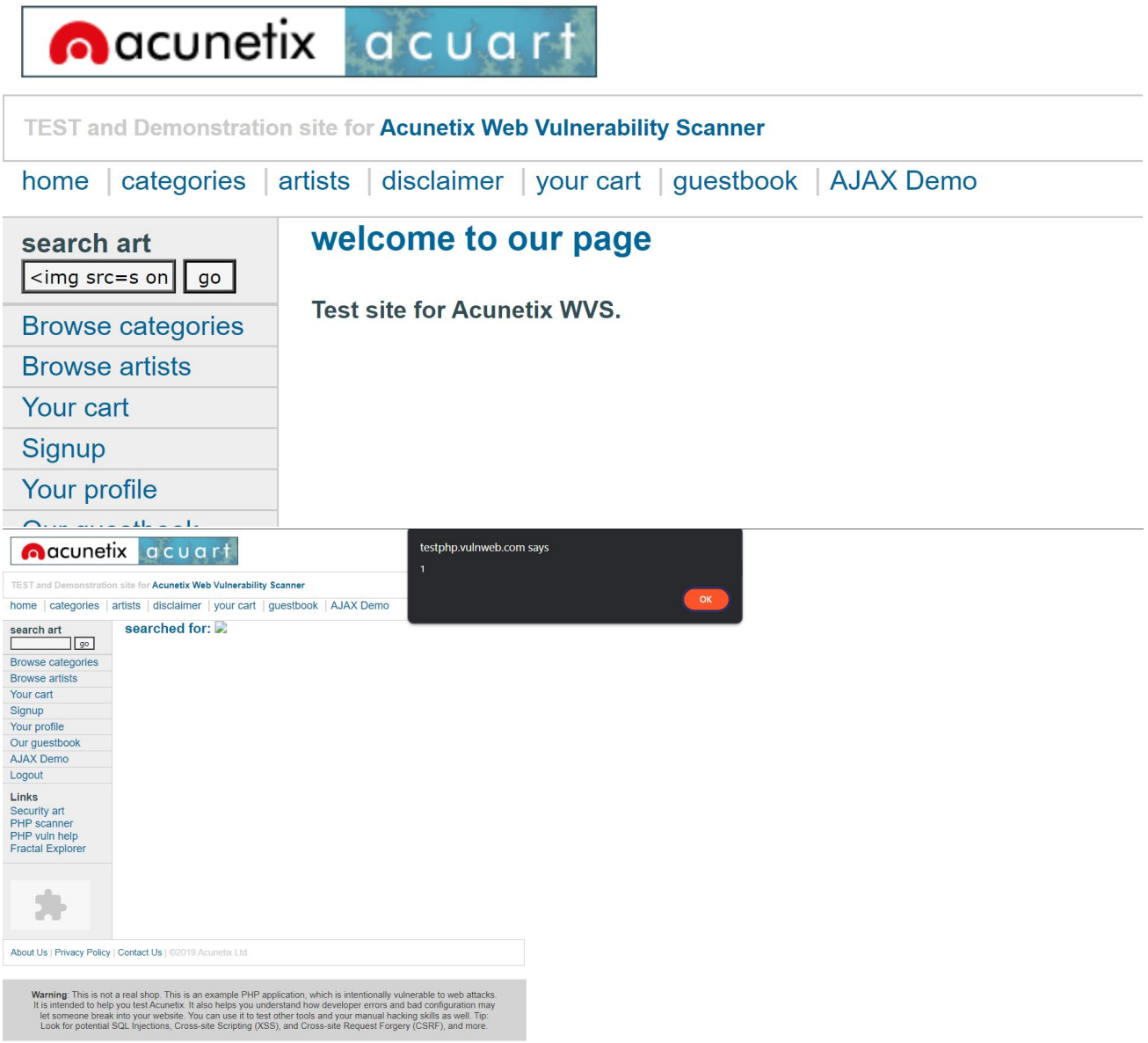
Recommendation:-

- Use prepared statements (parameterized queries): Prepared statements separate the SQL code from the user input.
- Validate and sanitize inputs: Rigorously examine all user inputs before incorporating them into SQL queries.
- Escape special characters
- Minimize database privileges
- Encrypt Data

11) Reflected Cross-site scripting in search bar(Critical)

Description	An attacker could inject malicious code into the search bar description. When another user sees that description and clicks the link, the attacker's code could run in the victim's browser. This code could steal the user's login information, redirect them to malicious sites, or disrupt the website itself. The command "*<img src=s onerror=alert(1)>*" was used.
Impact	High
System	http://testphp.vulnweb.com/

Proof of concept:



Recommendation:-

Input Validation and Filtering: Examine all user-supplied data upon receiving it.

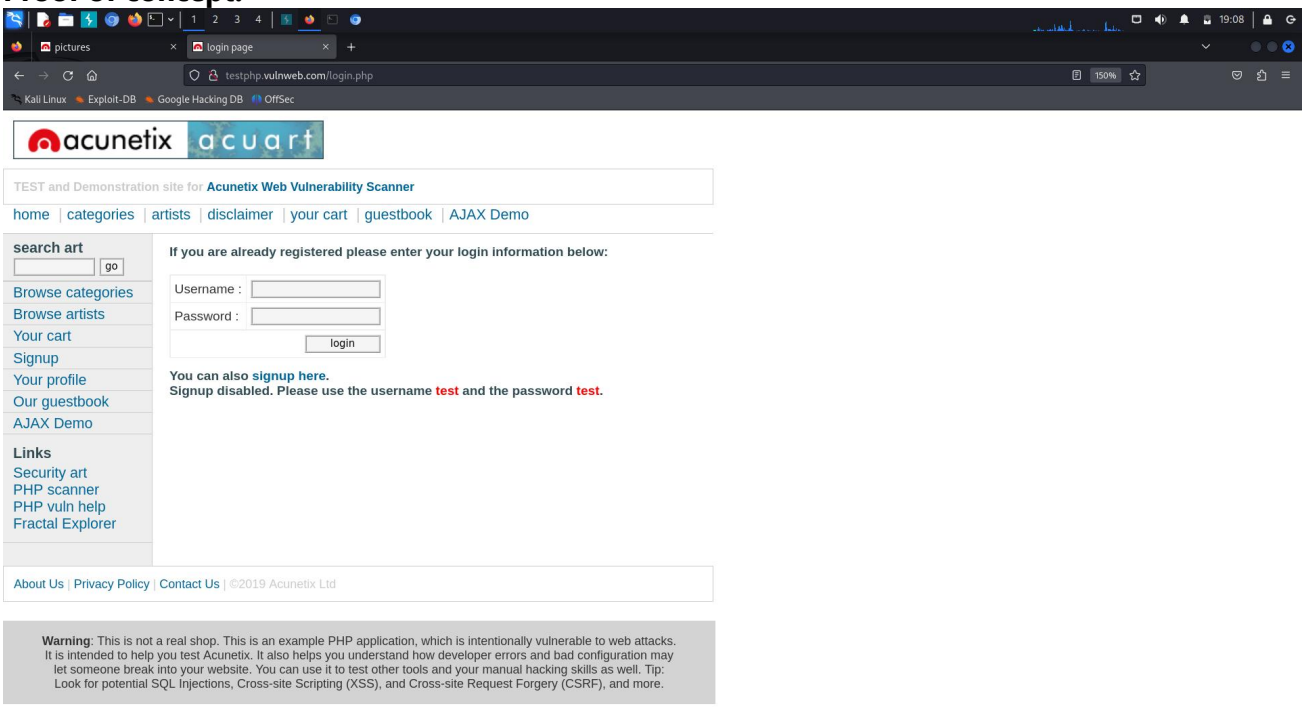
Output Encoding: Before displaying any user controlled data, encode it appropriately based on the context (HTML, URL, JavaScript, etc...)

Web Application Firewall: It can provide an additional layer of security. It can be configured to block malicious requests containing potential XSS payloads.

12) Weak Password Policy(High)

Description	The webpage follows a weak password policy. If a strong password policy is not implemented, it can make the webpage easily vulnerable to attacks.
Impact	High
System	http://testphp.vulnweb.com/login.php

Proof of concept:



Recommendation:-

- ✧ Create a password with minimum 8 characters.
- ✧ The password should contain special characters, numbers and symbols.
- ✧ Implement a password strength meter that gives users real-time feedback on the strength of their chosen password
- ✧ Store passwords securely using a hashing algorithm.
- ✧ Implement Multi-Factor Authentication which asks the user to enter a one-time-password which is sent to their registered mobile number or e-mail id.

Security Weakness

Weak Password Policy

I successfully performed password guessing attacks against testphp login forms. It doesn't have a strong password policy. Users are even able to create password with just 4 characters.

Unrestricted Login Attempts

During the assessment,I performed multiple brute-force attacks against login forms. For all logins, unlimited attempts were allowed, which permitted an eventual successful login.

Missing Multi-factor Authentication

Multi-factor Authentication is missing in the login page. It enables the attacker to login to the account with only username and password.