**COMPREHENSIVE VAPT ANALYSIS ON WEBSITE (http://testphp.vulnweb.com/)**

The domain of the Project

Cybersecurity - Network and Web Application Security

Under the guidance of

Mr. Nishchay Gaba (Penetration Tester)

By

Mr. Darshan Moradiya (B.Tech)

Period of the project

September 2024 to February 2025

SURE TRUST

PUTTAPARTHI, ANDHRA PRADESH

**DECLARATION**

The project titled **“*COMPREHENSIVE VAPT ANALYSIS ON WEBSITE (http://testphp.vulnweb.com/)*”** has been mentored by **Mr. Nishchay Gaba** and organized by SURE Trust from September 2024 to February 2025**.** This initiative aims to benefit educated unemployed rural youth by providing hands-on experience in industry-relevant projects, thereby enhancing employability.

I, **Mr. Darshan Moradiya,** hereby declare that I have solely worked on this project under the guidance of my mentor. This project has significantly enhanced my practical knowledge and skills in the domain.

# ***Table of Contents***

[***Table of Contents*** 2](#_Toc195116695)

[**1.** **Document control** 4](#_Toc195116696)

[**1.1** **Statement of confidential** 4](#_Toc195116697)

[**2.** **Disclaimer** 5](#_Toc195116698)

[**3.** **Introduction** 6](#_Toc195116699)

[**3.1** **Objectives** 6](#_Toc195116700)

[**3.2** **Scope of security assessment** 6](#_Toc195116702)

[**3.3** **Limitations** 6](#_Toc195116703)

[**4.** **Executive Summary** 7](#_Toc195116705)

[**4.1** **Summary of findings** 7](#_Toc195116706)

[**5.** **Scope of security assessment** 8](#_Toc195116707)

[**6.** **Testing Approach** 8](#_Toc195116708)

[**7.** **DOMAIN1 - http://testphp.vulnweb.com/** (Format) 10](#_Toc195116709)

[**8.** **Conclusion** 76](#_Toc195116710)

1. **Document control**
   1. **Statement of confidential**

**Confidentiality Notice:**

This document contains **sensitive and proprietary information** related to the security posture of the assessed web application and associated infrastructure. It is intended **solely for the use of authorized personnel** explicitly designated by [Client/Organization Name].

**Unauthorized Access, Disclosure, or Distribution** of this document, in whole or in part, is **strictly prohibited** and may result in legal action. Recipients of this document are obligated to:

1. Maintain the confidentiality of all information contained herein.
2. Use the information **exclusively** for remediation and security enhancement purposes.
3. Store the document securely, with access limited to individuals directly involved in addressing the identified vulnerabilities.

**Non-Disclosure Agreement (NDA):**

By accepting this document, you acknowledge and agree to:

* **Not share, reproduce, or transmit** this report via unsecured channels (e.g., email, public cloud storage).
* **Immediately report** any unauthorized disclosure to [Client/Organization Name] at [Security Team Contact Email/Phone].

**Destroy or return** all copies of this document upon

1. **Disclaimer**

The information provided in this report is based on findings from the security assessment conducted on **[25 Feb 2025]**.

1. **No Warranties**:
   * The authors of this report make **no representations or warranties**, express or implied, regarding the **accuracy, completeness, or reliability** of the information contained herein. Vulnerabilities or risks may exist that were not identified during the assessment.
2. **Limitation of Liability**:
   * **[Organization Name]** and its affiliates shall **not be liable** for any direct, indirect, incidental, or consequential damages arising from the use of, reliance on, or interpretation of this report or its contents.
3. **Scope Limitations**:
   * This report addresses only the vulnerabilities identified during the assessment period. It does **not guarantee** the absence of future vulnerabilities or the invulnerability of the system.
4. **Third-Party Reliance**:
   * Findings are specific to the assessed environment and timeframe. **Do not extrapolate** results to other systems or future states without further evaluation.
5. **Evolving Threats**:
   * Cybersecurity threats are dynamic. Regular assessments are required to maintain security posture.
6. **Introduction**
   1. **Objectives**

The primary objective of this VAPT is to identify vulnerabilities within the Website that could be exploited by malicious actors. The assessment aims to provide actionable recommendations to enhance the security posture of the organization.

* 1. **Scope of security assessment**

|  |  |
| --- | --- |
| **DEVICES** | **TOTAL IPS** |
| **DOMAIN** | **http://testphp.vulnweb.com/** |

Table 1 - Scop List

* 1. **Limitations**

The assessment depends on provided data, excluding physical security and social engineering. Time, scope, and dynamic network changes constrain findings. It may not detect all vulnerabilities, potentially including false positives or negatives, and is bound by legal and ethical guidelines.

1. **Executive Summary**
   1. **Summary of findings**

**Executive Summary:**

A comprehensive vulnerability assessment was conducted on two websites, testphp.vulnweb.com and career.abb.com, to identify security flaws and recommend mitigation strategies. The assessment revealed several critical, medium, and low-severity vulnerabilities across both websites. The identified vulnerabilities pose risks such as unauthorized data access, session hijacking, phishing, and potential system compromise. Immediate remediation is advised for high-severity findings to minimize security risks.

**Overall Risk:**

* *testphp.vulnweb.com*: High – Requires immediate attention due to SQL Injection and XSS vulnerabilities.
* *career.abb.com*: Medium – Open Redirect and Information Disclosure vulnerabilities need to be addressed promptly.

**Recommendations and Best Practices**

* Prioritize patching high-risk vulnerabilities immediately.
* Conduct regular vulnerability scans and penetration tests.
* Implement a robust Web Application Firewall (WAF) to prevent common attacks.
* Adhere to OWASP best practices for secure coding.
* Continuously monitor application logs for suspicious activity and implement an incident response plan.

1. **Scope of security assessment**

Each finding has been assigned a severity rating of CRITICAL, HIGH, MEDIUM, LOW. The rating is based on an assessment of the priority with which each finding should be viewed and the potential impact each has on the confidentiality, integrity, and availability of the Client's data.

|  |  |
| --- | --- |
| **Severity** | **Definition** |
| **Critical 9.0 - 10.0** | Vulnerabilities that score in the critical range usually have most of the following characteristics: • Exploitation of the vulnerability likely results in root-level compromise of servers or infrastructure devices. • Exploitation is usually straightforward, in the sense that the attacker does not need any special authentication credentials or knowledge about individual victims, and does not need to persuade a target user, for example via social engineering, into performing any special functions. • It is advised that you patch or upgrade as soon as possible, unless you have other mitigating measures in place |
| **High 7.0 - 8.9** | Vulnerabilities that score in the high range usually have some of the following characteristics: • The vulnerability is difficult to exploit. • Exploitation could result in elevated privileges. • Exploitation could result in a significant data loss or downtime |
| **Medium 4.0 - 6.9** | Vulnerabilities that score in the medium range usually have some of the following characteristics: • Vulnerabilities that require the attacker to manipulate individual victims via social engineering tactics. • Denial of service vulnerabilities that are difficult to set up. • Exploits that require an attacker to reside on the same local network as the victim. • Vulnerabilities where exploitation provides only very limited access. • Vulnerabilities that require user privileges for successful exploitation. |
| **Low 0.1 - 3.9** | Vulnerabilities in the low range typically have very little impact on an organization's business. Exploitation of such vulnerabilities usually requires local or physical system access. Vulnerabilities in third party code that are unreachable from Atlassian code may be downgraded to low severity. |

*Table 2. Risk Impact Definition*

1. **Testing Approach**

**Black Box Assessment** - A Black-box penetration test is a penetration testing service that aims to find & exploit vulnerabilities in a system as an outsider. The security expert is provided with no information about the target system prior to the testing except the target URL and access similar to an end-user. This means the tester has no access to source code (other than publicly available code), internal data, structure & design of the application before the testing.

1. **DOMAIN1 -** [**http://testphp.vulnweb.com/**](http://testphp.vulnweb.com/) (Format)

**Vulnerability Name:**

**Description:**

**CVSS:**

**Affected IPs:**

**Affected Port:**

**CVE-ID:**

**Technical Impact:**

**Mitigation:**

**Reference:**

**POC:**

DOMAIN 1

Executive Summary

Critical

1. **Vulnerability Name: SQL Injection (SQLi)**

* **Description: Attackers inject malicious SQL queries into input fields (e.g., login forms, search bars) to manipulate or extract database contents.**
* **CVSS: 9.0 (Critical) – AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H**
* **Affected URL:**[**http://testphp.vulnweb.com/listproducts.php?cat=3**](http://testphp.vulnweb.com/listproducts.php?cat=3) **,** [**http://testphp.vulnweb.com/artists.php?artist=1**](http://testphp.vulnweb.com/artists.php?artist=1) **,** [**http://testphp.vulnweb.com/listproducts.php?artist=1**](http://testphp.vulnweb.com/listproducts.php?artist=1)[**http://testphp.vulnweb.com/product.php?pic=1**](http://testphp.vulnweb.com/product.php?pic=1)

[**http://testphp.vulnweb.com/login.php**](http://testphp.vulnweb.com/login.php)

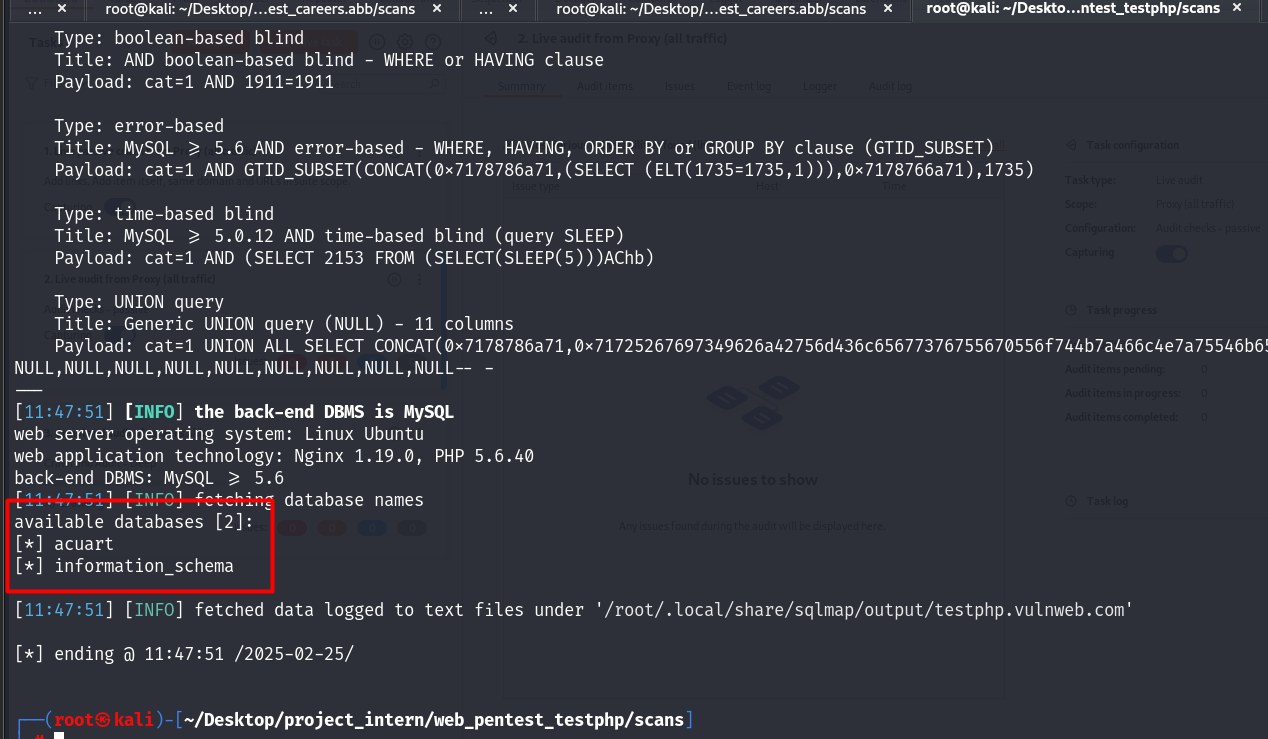
[**http://testphp.vulnweb.com/search.php?search=**](http://testphp.vulnweb.com/search.php?search=searchterm)

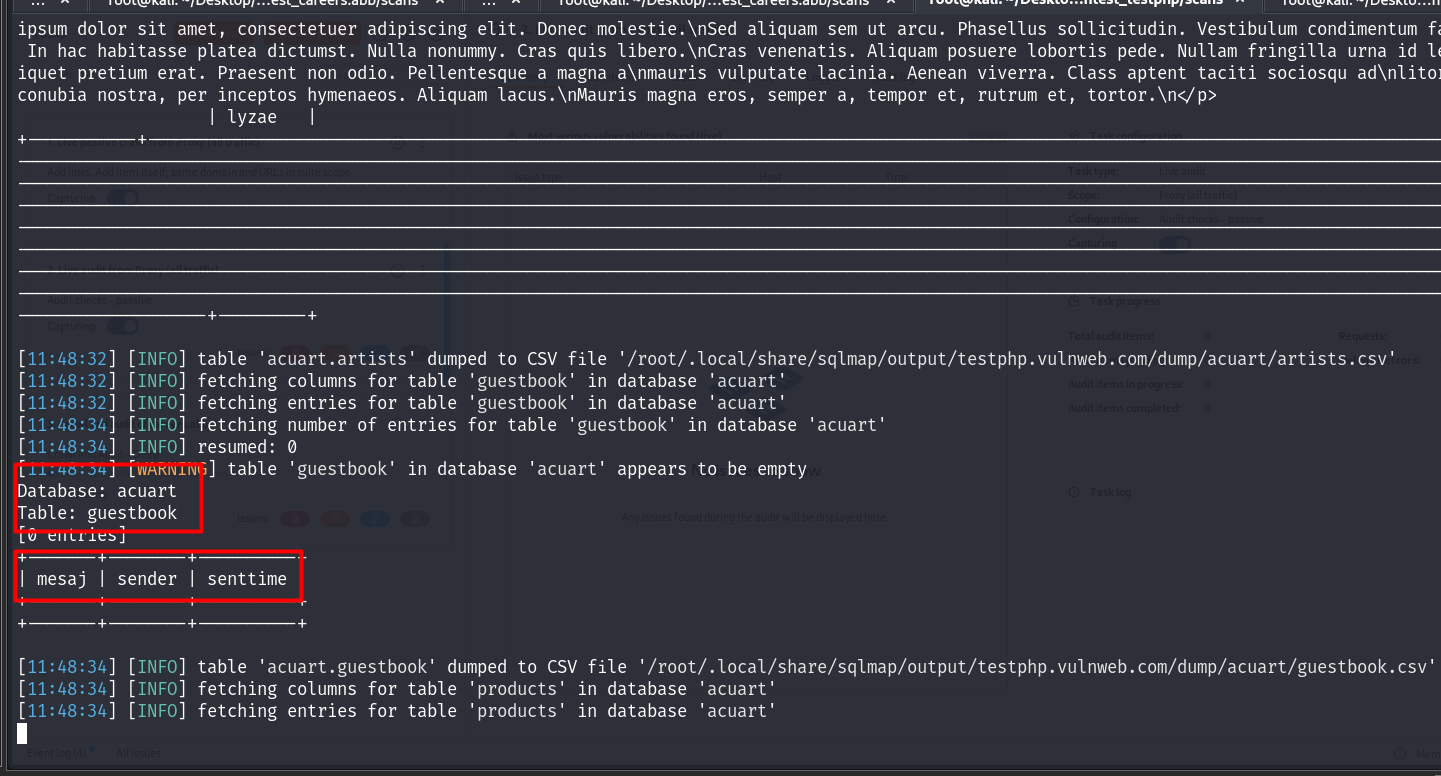
[**http://testphp.vulnweb.com/search.php?test=**](http://testphp.vulnweb.com/search.php?test=)

[**http://testphp.vulnweb.com/userinfo.php?uid=**](http://testphp.vulnweb.com/userinfo.php?uid=1%27%20OR%20%271%27=%271%27%20--)

[**http://testphp.vulnweb.com/guestbook.php**](http://testphp.vulnweb.com/guestbook.php)

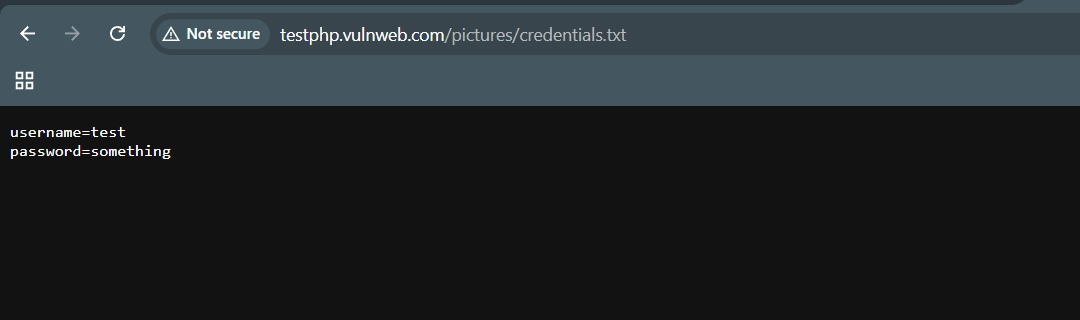
* **CVE-ID: N/A**
* **Technical Impact:**
  + **Unauthorized access to sensitive data (passwords, credit cards).**
  + **Database corruption or deletion (e.g., DROP TABLE users).**
  + **Remote Code Execution (RCE) if database functions allow OS command execution.**
* **Mitigation:**
  1. **Use parameterized queries (e.g., PreparedStatement in Java, PDO in PHP).**
  2. **Validate inputs using allowlist regex (e.g., ^[a-zA-Z0-9]+$).**
  3. **Deploy a WAF (e.g., ModSecurity) to block SQLi payloads.**
* **Reference:**[**OWASP SQL Injection**](https://owasp.org/www-community/attacks/SQL_Injection)
* **OWASP TOP 10: A03:2021 – Injection**
* **POC:**

****

****

1. **Vulnerability Name: Obtained Login Credentials**

* **Description: Credentials exposed in logs, code repositories, or public databases.**
* **CVSS: 9.0 (Critical) – AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H**
* **Affected URL:**[**http://testphp.vulnweb.com/pictures/credentials.txt**](http://testphp.vulnweb.com/pictures/credentials.txt)
* **CVE-ID: N/A**
* **Technical Impact:**
  + **Full account compromise and lateral movement within the network.**
* **Mitigation:**
  1. **Rotate credentials immediately after exposure.**
  2. **Use secret scanning tools (e.g., GitGuardian, TruffleHog).**
  3. **Monitor logs for unauthorized access attempts.**
* **Reference:**[**OWASP Credential Stuffing**](https://owasp.org/www-community/attacks/Credential_stuffing)
* **OWASP TOP 10: A07:2021 – Identification and Authentication Failures**
* **POC:**

****

1. **Vulnerability Name: Outdated PHP Version (EOL)**

* **Description: The server uses PHP 5.6.40, which is End-of-Life (EOL) and no longer receives security patches.**
* **CVSS: 9.8 (Critical) – AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H**
* **Affected URL:**[**http://testphp.vulnweb.com/**](http://testphp.vulnweb.com/)
* **CVE-ID:**
  + **CVE-2019-11043: Remote Code Execution in PHP-FPM.**
  + **CVE-2019-11044: Buffer Overflow in PHP-FPM.**
* **Technical Impact:**
  + **Remote Code Execution (RCE) via unpatched PHP vulnerabilities.**
  + **Denial-of-Service (DoS) attacks.**
* **Mitigation:**
  1. **Upgrade to PHP 7.4+ or 8.x (supported versions).**
  2. **Apply security patches if upgrading is not feasible.**
  3. **Disable dangerous PHP functions (e.g., exec, system).**
* **Reference:**[**PHP 5.6 EOL Notice**](https://www.php.net/eol.php)
* **OWASP TOP 10: A06:2021 – Vulnerable and Outdated Components**
* **POC:**

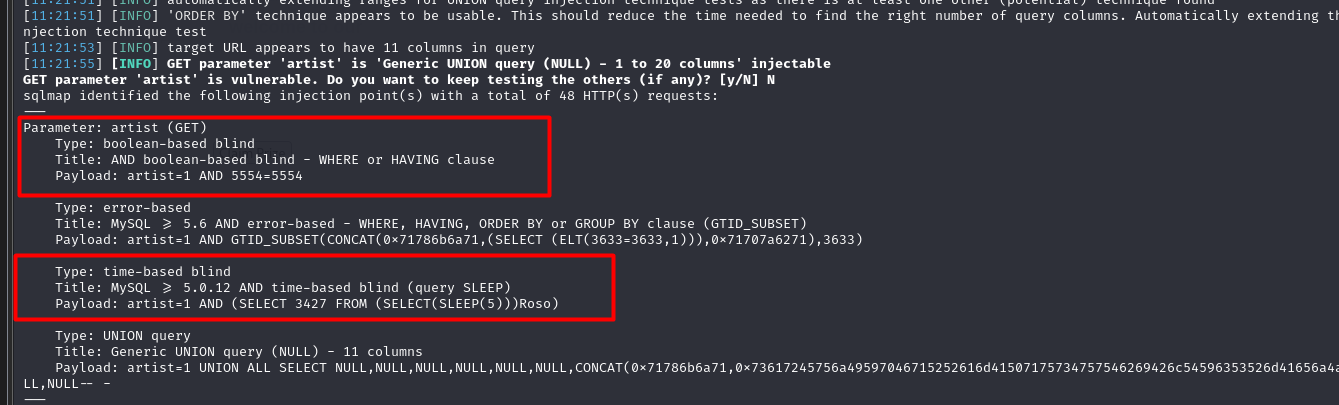


High

1. **Vulnerability Name: Blind SQL Injection**

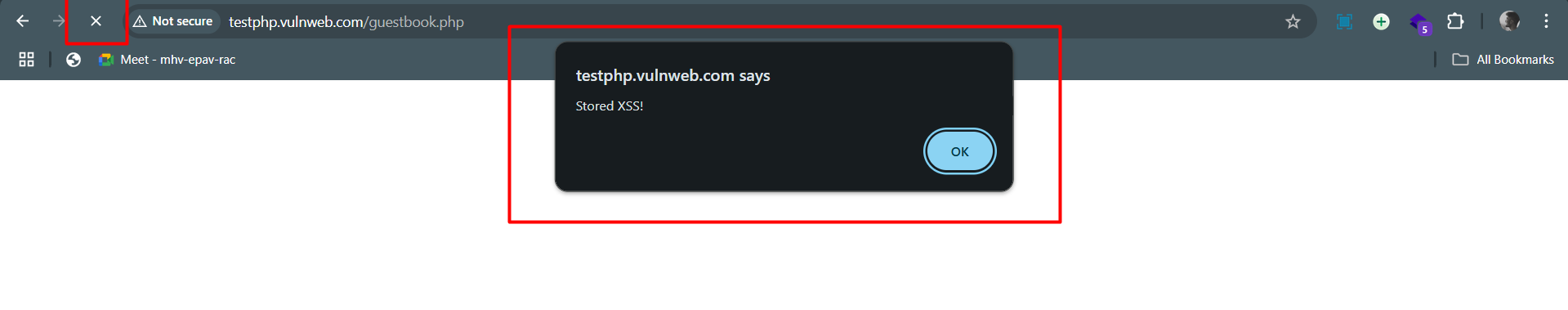
* **Description: Attackers infer database structure/content via boolean or time-based responses (e.g., AND 1=1, SLEEP(5)).**
* **CVSS: 8.5 (High) – AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H**
* **Affected URL:**[**http://testphp.vulnweb.com/listproducts.php?cat=1**](http://testphp.vulnweb.com/listproducts.php?cat=1)
* **CVE-ID: N/A**
* **Technical Impact:**
  + **Stealthy extraction of sensitive data (e.g., admin credentials).**
  + **Denial-of-Service (DoS) via resource-heavy queries.**
* **Mitigation:**
  1. **Use stored procedures with strict input validation.**
  2. **Implement rate limiting to block repetitive malicious requests.**
  3. **Log and monitor for abnormal query patterns (e.g., SLEEP()).**
* **Reference:**[**OWASP Blind SQLi**](https://owasp.org/www-community/attacks/Blind_SQL_Injection)
* **OWASP TOP 10: A03:2021 – Injection**
* **POC:**

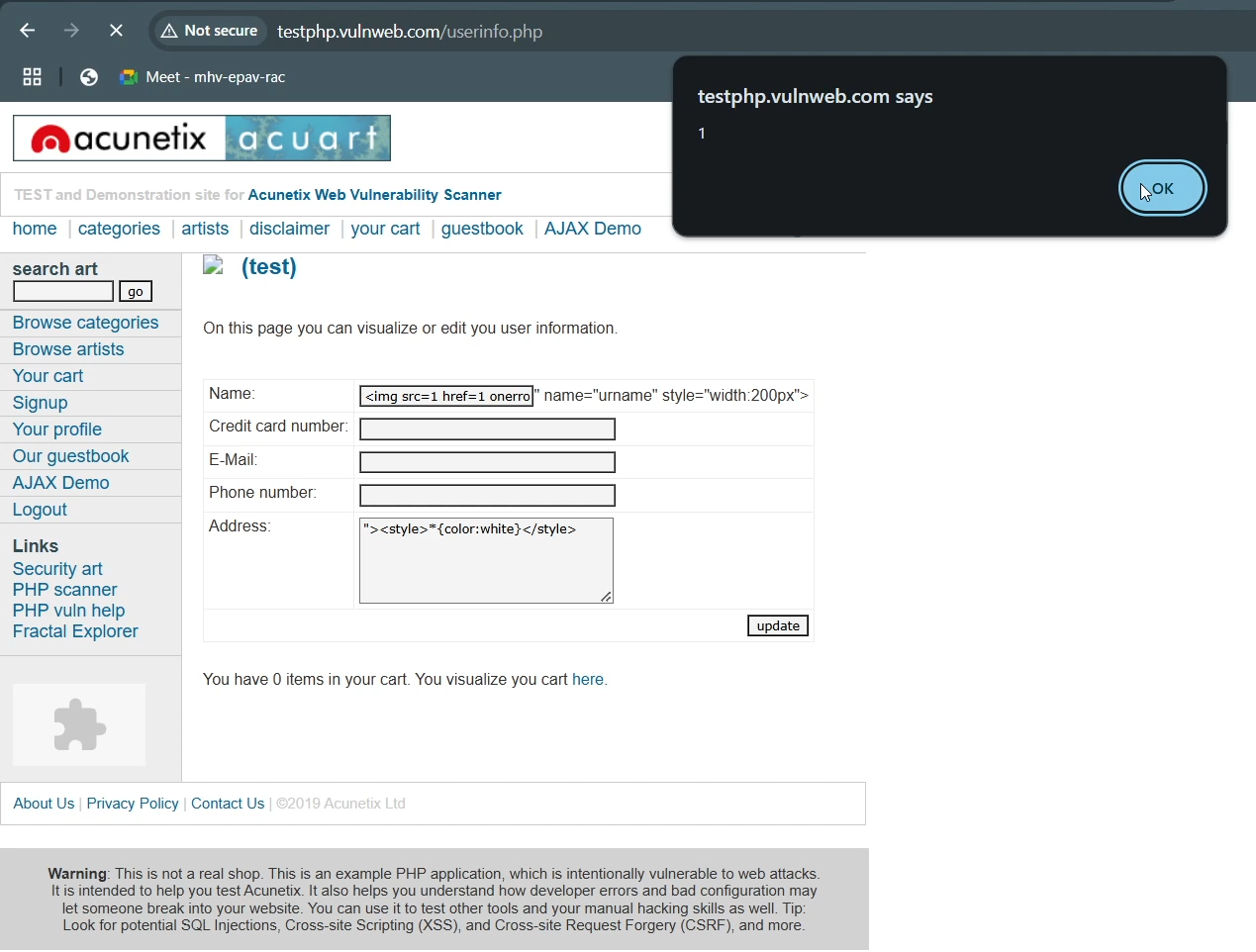
****



1. **Vulnerability Name: XSS – Stored (Persistent)**

* **Description: Malicious scripts stored in the application (e.g., comments, profiles) execute for all users accessing the page.**
* **CVSS: 8.0 (High) – AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:L/A:N**
* **Affected URL:**[**http://testphp.vulnweb.com/guestbook.php**](http://testphp.vulnweb.com/guestbook.php)
* **CVE-ID: N/A**
* **Technical Impact:**
  + **Persistent phishing attacks (e.g., fake login forms).**
  + **Theft of CSRF tokens or session cookies.**
* **Mitigation:**
  1. **Sanitize user inputs with DOMPurify or OWASP Java Encoder.**
  2. **Use CSP with nonces to allow only trusted scripts.**
  3. **Regularly audit user-generated content for malicious scripts.**
* **Reference:**[**OWASP Stored XSS**](https://owasp.org/www-community/attacks/xss/)
* **OWASP TOP 10: A03:2021 – Injection**
* **POC:**

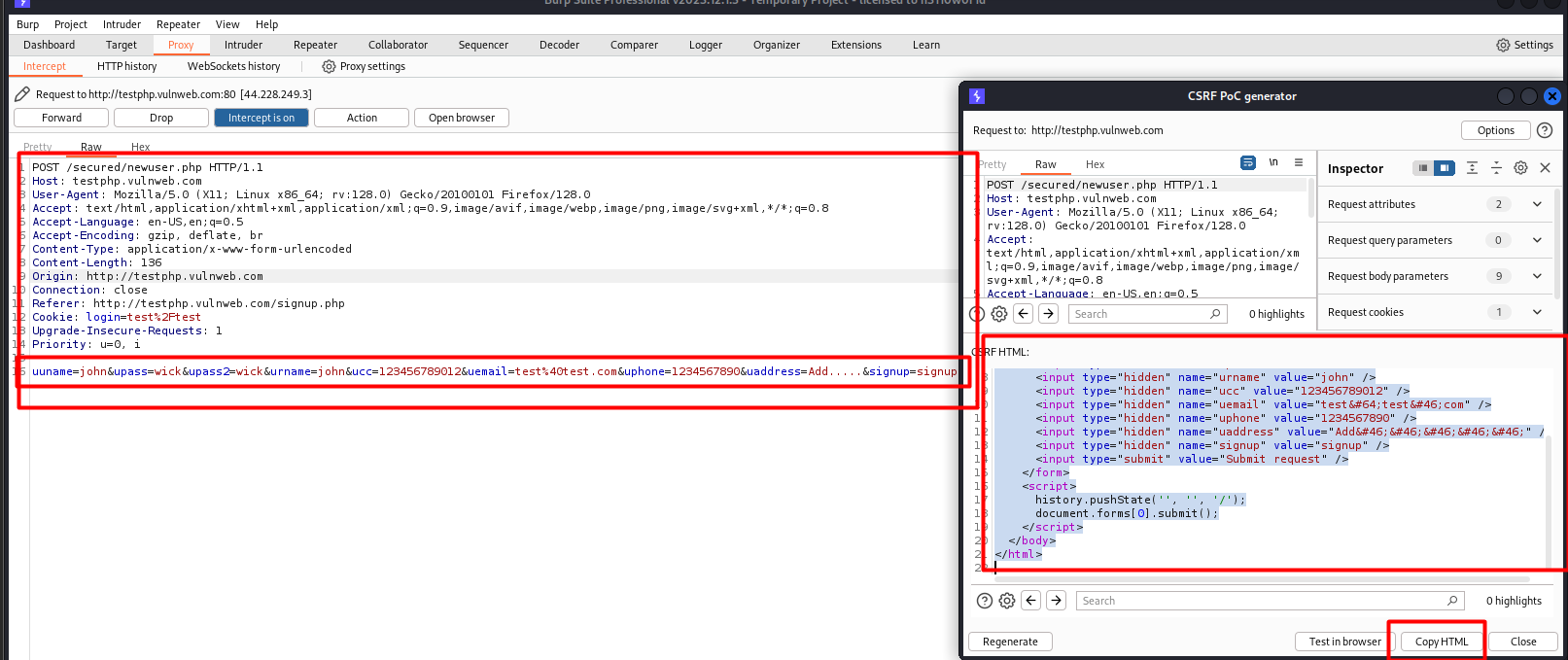
****

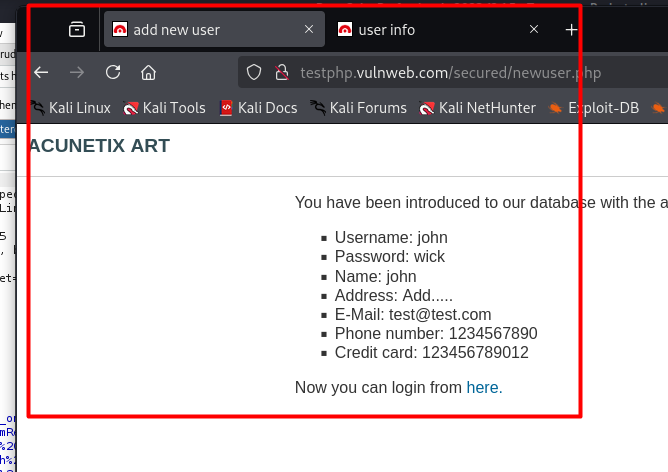
****

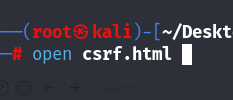
1. **Vulnerability Name: Cross-Site Request Forgery (CSRF)**

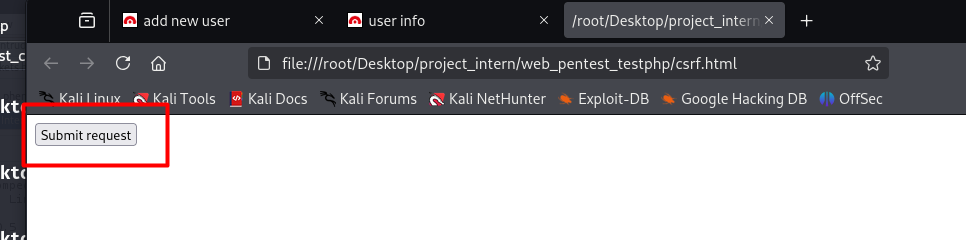
* **Description: Attackers trick users into executing unintended actions (e.g., password change, fund transfer).**
* **CVSS: 7.5 (High) – AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N**
* **Affected URL:**[**http://testphp.vulnweb.com/secured/newuser.php**](http://testphp.vulnweb.com/secured/newuser.php)
* **CVE-ID: N/A**
* **Technical Impact:**
  + **Unauthorized financial transactions.**
  + **Account takeover via email/password change.**
* **Mitigation:**
  1. **Implement anti-CSRF tokens (synchronizer tokens).**
  2. **Use SameSite=Lax for session cookies.**
  3. **Validate the Origin and Referer headers.**
* **Reference:**[**OWASP CSRF**](https://owasp.org/www-community/attacks/csrf)
* **OWASP TOP 10: A01:2021 – Broken Access Control**

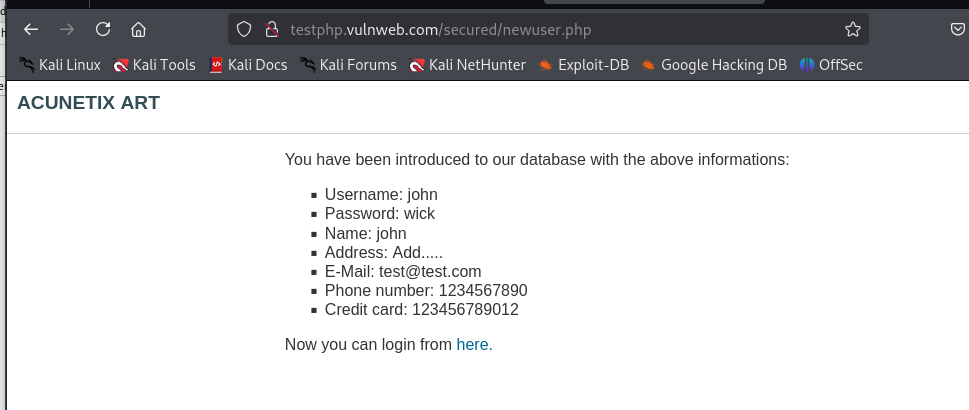
**POC:**

****

****

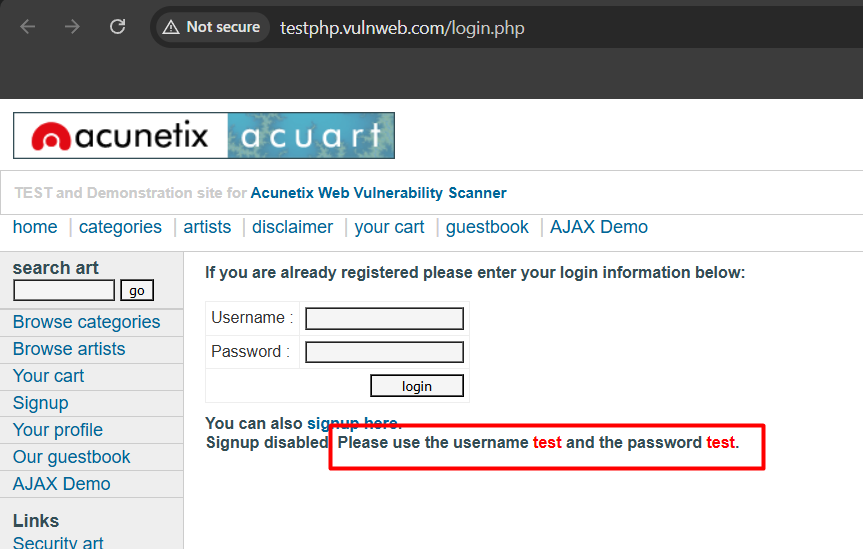
****

****

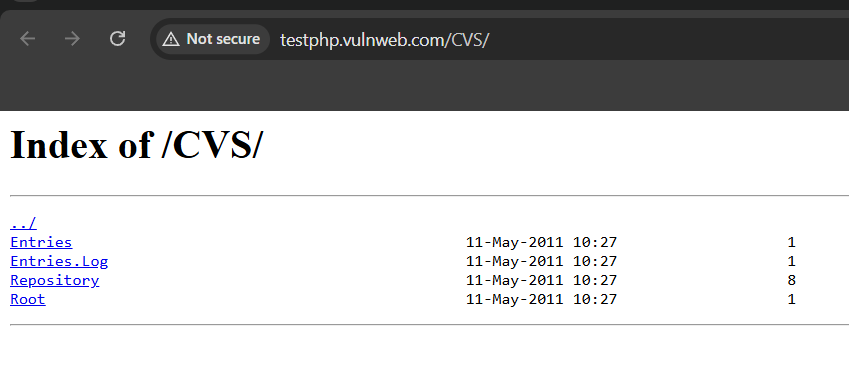
****

1. **Vulnerability Name: Sensitive Information Disclosure**

* **Description: Error messages or misconfigurations expose internal data (e.g., stack traces, credentials).**
* **CVSS: 7.5 (High) – AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N**
* **Affected URL: http://testphp.vulnweb.com/login.php**
* **CVE-ID: N/A**
* **Technical Impact:**
  + **Database credentials leaked, leading to full system compromise.**
* **Mitigation:**
  1. **Replace debug messages with generic errors (e.g., "An error occurred").**
  2. **Sanitize logs to remove sensitive data.**
  3. **Separate development, staging, and production environments.**
* **Reference:**[**OWASP Info Exposure**](https://owasp.org/www-community/vulnerabilities/Information_exposure)
* **OWASP TOP 10: A05:2021 – Security Misconfiguration**
* **POC:**

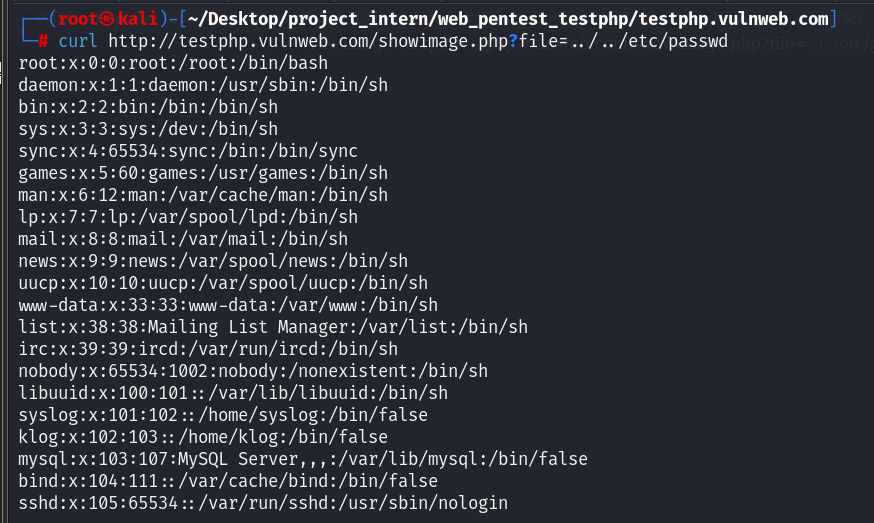
****

****

****

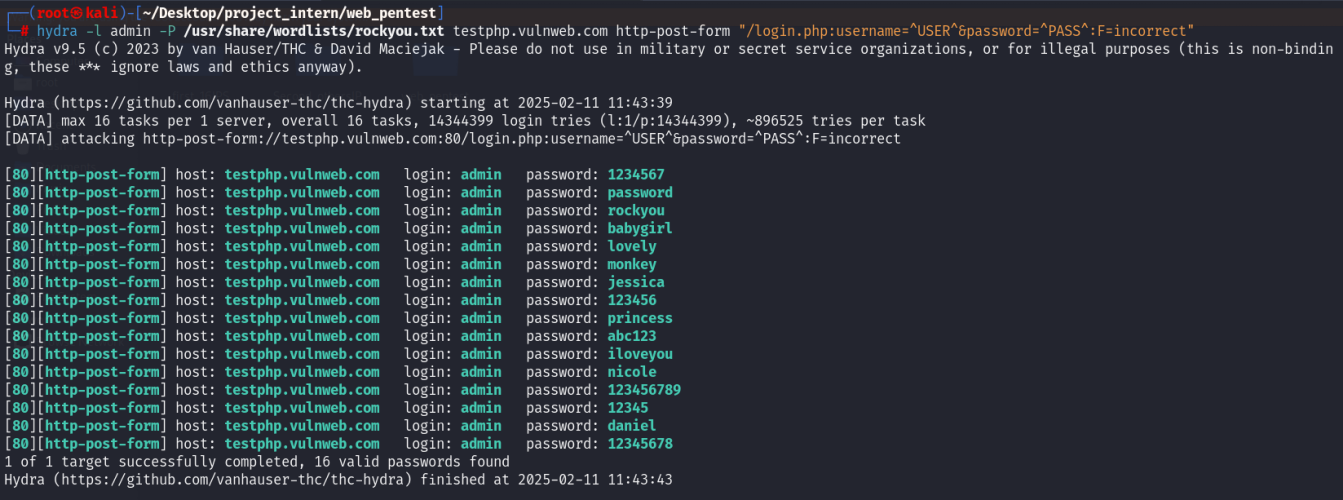
1. **Vulnerability Name: Local File Inclusion (LFI)**

* **Description: Attackers manipulate file paths (e.g., ../../etc/passwd) to read sensitive local files.**
* **CVSS: 7.5 (High) – AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N**
* **Affected URL:**[**http://testphp.vulnweb.com/showimage.php?file=**](http://testphp.vulnweb.com/showimage.php?file=.%2Fpictures%2F1.jpg)**../../etc/passwd**
* **CVE-ID: N/A**
* **Technical Impact:**
  + **Exposure of system files (e.g., /etc/shadow, .env).**
  + **Source code leaks leading to further vulnerabilities.**
* **Mitigation:**
  1. **Use allowlists for allowed file paths.**
  2. **Normalize and validate user-supplied file paths.**
  3. **Run the application with least-privilege user permissions.**
* **Reference:**[**OWASP LFI**](https://owasp.org/www-community/attacks/Path_Traversal)
* **OWASP TOP 10: A01:2021 – Broken Access Control**
* **POC:**



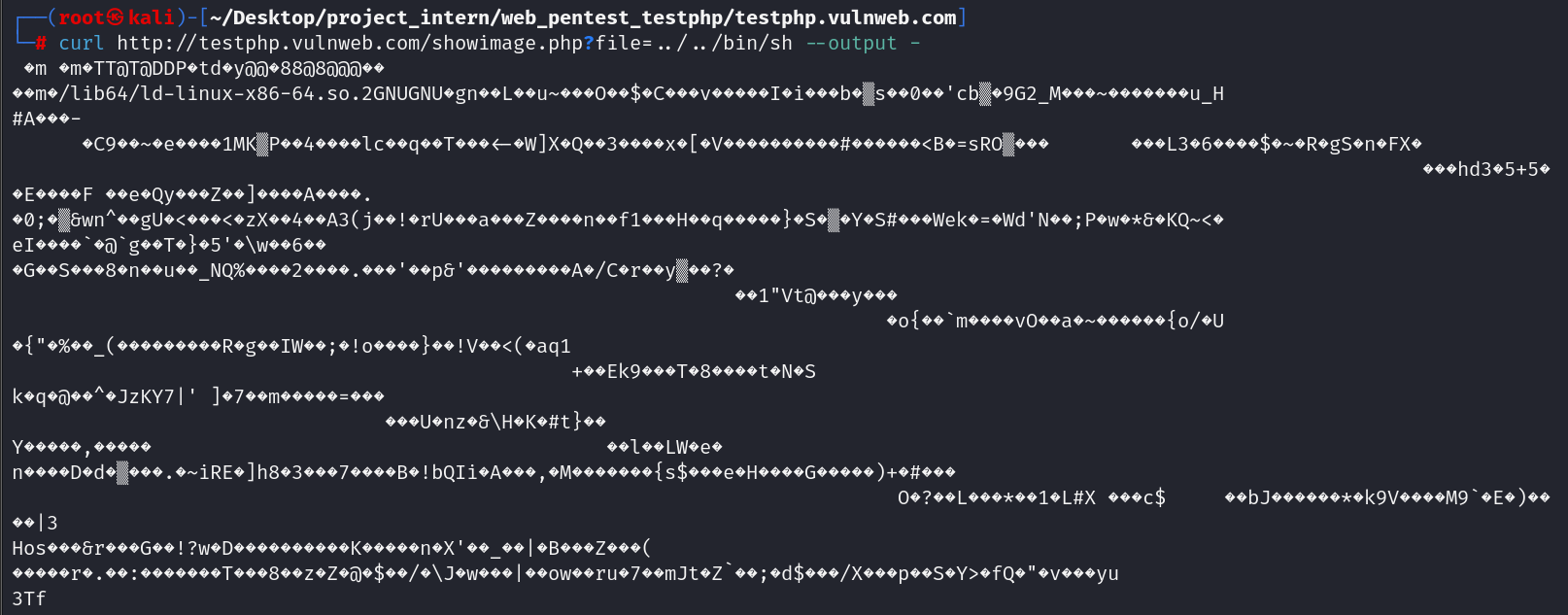
1. **Vulnerability Name: Broken Authentication (No Rate Limit)**

* **Description: No restrictions on login attempts, enabling brute-force attacks.**
* **CVSS: 7.5 (High) – AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N**
* **Affected URL: testphp.vulnweb.com/login.php**
* **CVE-ID: N/A**
* **Technical Impact:**
  + **Account takeover via credential stuffing.**
  + **Lockout of legitimate users via DoS.**
* **Mitigation:**
  1. **Enforce rate limiting (e.g., 5 attempts/minute).**
  2. **Implement account lockouts after 10 failed attempts.**
  3. **Use CAPTCHA for repeated login attempts.**
  4. **Use Threshold policy**
* **Reference:**[**OWASP Broken Authentication**](https://owasp.org/www-project-top-ten/2017/A2_2017-Broken_Authentication)
* **OWASP TOP 10: A07:2021 – Identification and Authentication Failures**
* **POC:**

****

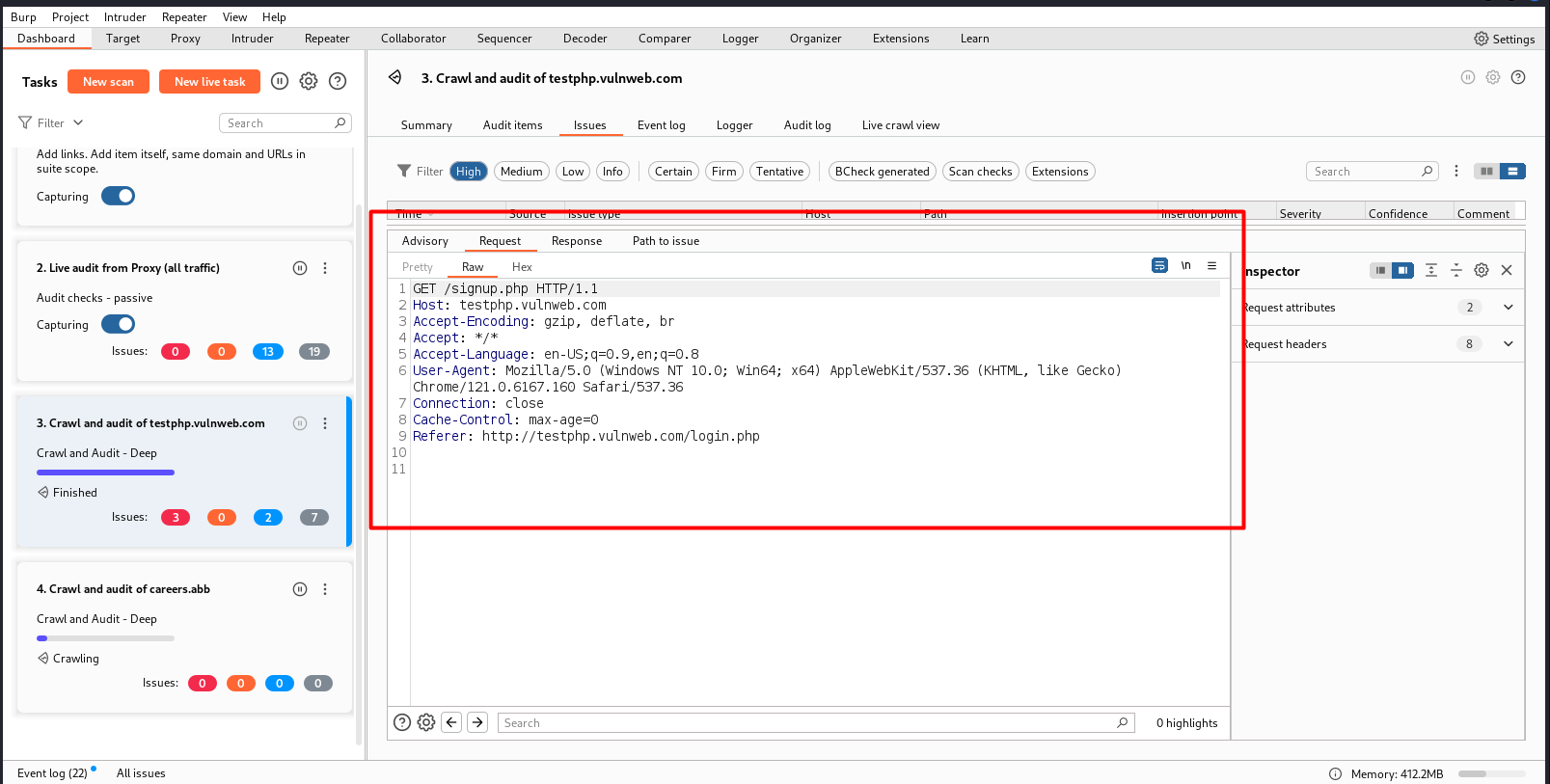
1. **Vulnerability Name: Path Traversal**

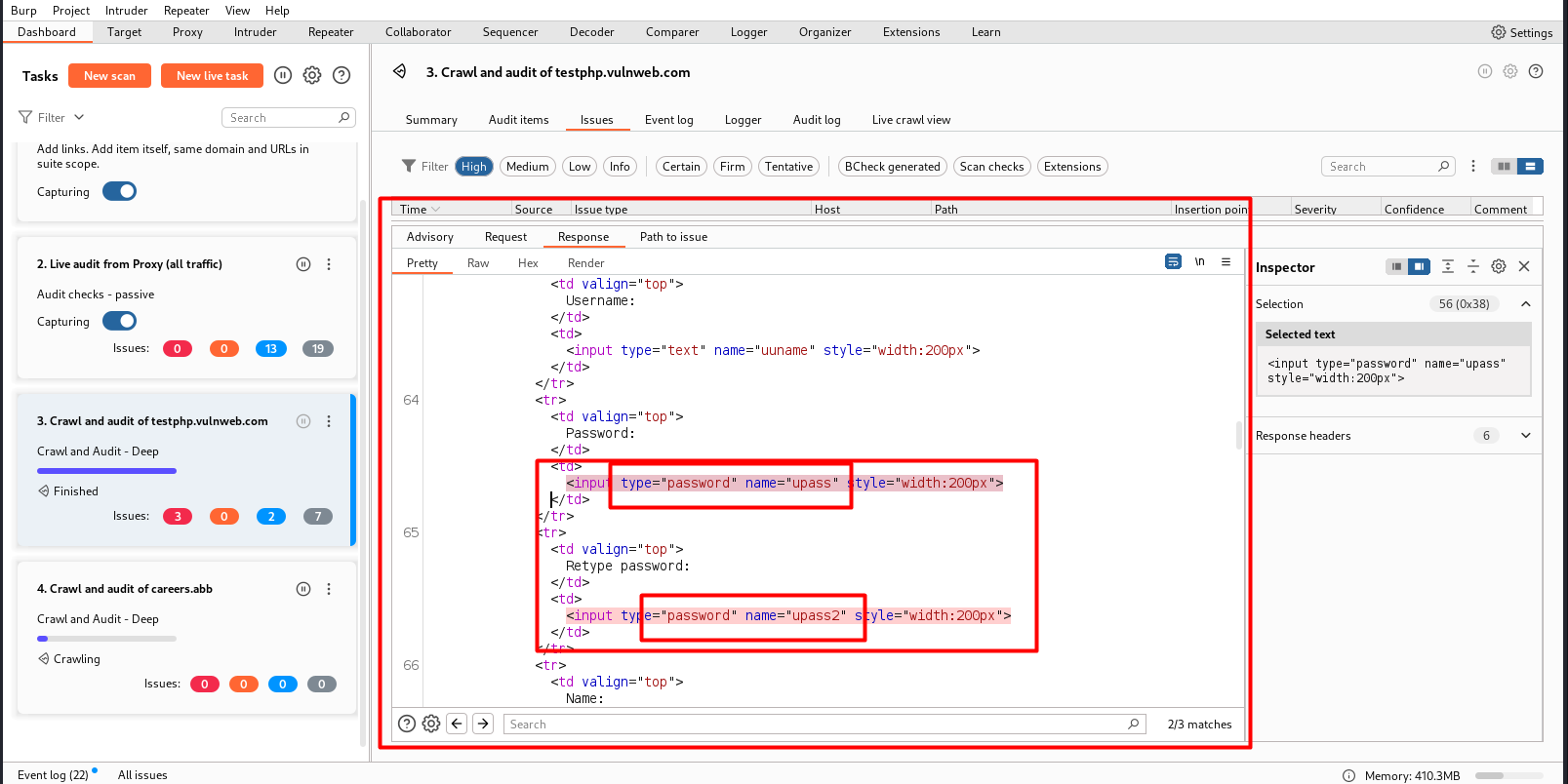
* **Description: Attackers bypass directory restrictions to access unauthorized files.**
* **CVSS: 7.5 (High) – AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N**
* **Affected URL: http://testphp.vulnweb.com/showimage.php?file=../../bin/sh**
* **CVE-ID: N/A**
* **Technical Impact:**
  + **Exposure of configuration files, backups, or logs.**
  + **Privilege escalation via SSH key leaks.**
* **Mitigation:**
  1. **Use allowlists for valid file names.**
  2. **Normalize paths using realpath() or Path.GetFullPath().**
  3. **Run the app in a chroot jail or container.**
* **Reference:**[**OWASP Path Traversal**](https://owasp.org/www-community/attacks/Path_Traversal)
* **OWASP TOP 10: A01:2021 – Broken Access Control**
* **POC:**



1. **Vulnerability Name: Cleartext Submission of Password**

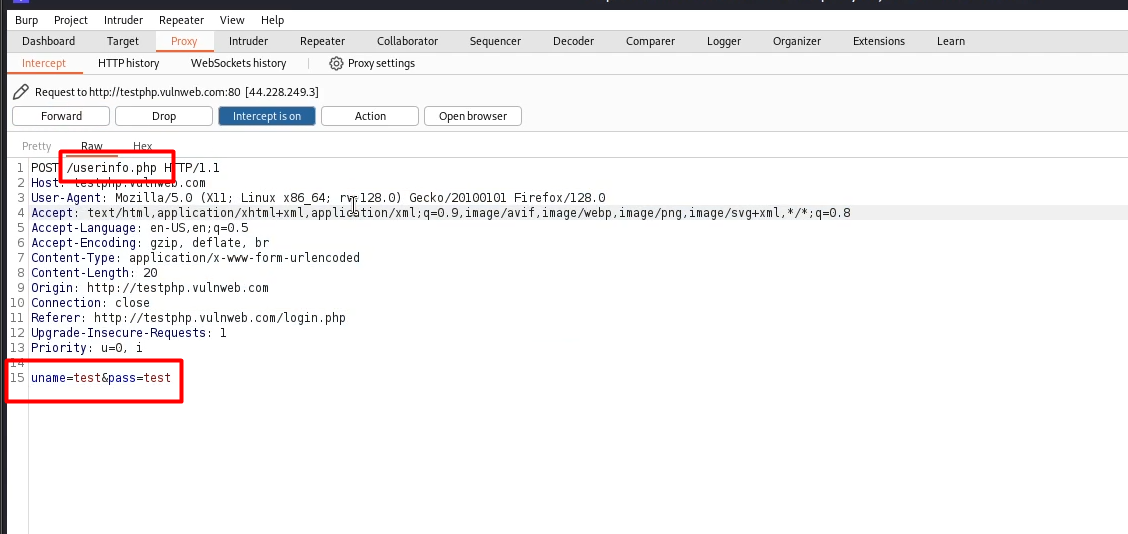
* **Description: Passwords transmitted over unencrypted HTTP.**
* **CVSS: 7.5 (High) – AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N**
* **Affected URL:**[**http://testphp.vulnweb.com/userinfo.php**](http://testphp.vulnweb.com/userinfo.php)**,** [**http://testphp.vulnweb.com/secured/newuser.php**](http://testphp.vulnweb.com/secured/newuser.php)
* **CVE-ID: N/A**
* **Technical Impact:**
  + **Credential theft via network sniffing (e.g., public Wi-Fi).**
* **Mitigation:**
  1. **Enforce HTTPS with 301 redirects.**
  2. **Implement HSTS headers (Strict-Transport-Security: max-age=31536000).**
  3. **Hash passwords client-side (e.g., using bcrypt) before transmission.**
* **Reference:**[**OWASP Sensitive Data Exposure**](https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure)
* **OWASP TOP 10: A02:2021 – Cryptographic Failures**
* **POC:**

****

****

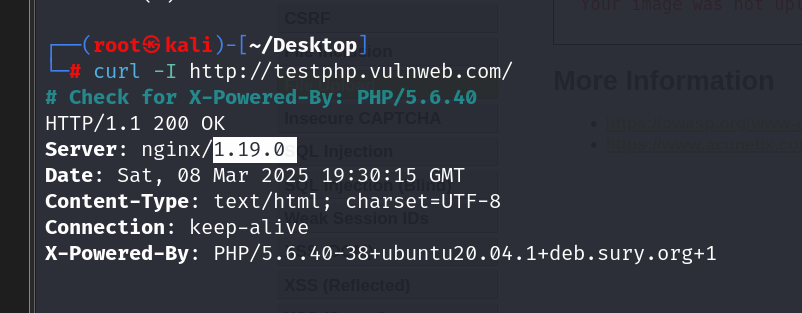
1. **Vulnerability Name: Username/Password in Plaintext**

* **Description: Credentials stored unencrypted in databases or logs.**
* **CVSS: 7.5 (High) – AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N**
* **Affected URL:**[**http://testphp.vulnweb.com/userinfo.php**](http://testphp.vulnweb.com/userinfo.php)
* **CVE-ID: N/A**
* **Technical Impact:**
  + **Credential theft if the database is breached.**
* **Mitigation:**
  1. **Hash passwords using bcrypt/Argon2 with unique salts.**
  2. **Encrypt databases at rest (e.g., AES-256).**
  3. **Use secrets management tools (e.g., HashiCorp Vault).**
* **Reference:**[**OWASP Password Storage**](https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html)
* **OWASP TOP 10: A02:2021 – Cryptographic Failures**
* **POC:**

****

1. **Vulnerability Name: Outdated nginx Server**

* **Description: The server runs nginx 1.19.0, which has known vulnerabilities.**
* **CVSS: 7.5 (High) – AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N**
* **Affected URL:**[**http://testphp.vulnweb.com/**](http://testphp.vulnweb.com/)
* **CVE-ID:**
  + **CVE-2021-23017: Use-after-free in nginx resolver.**
  + **CVE-2020-12440: HTTP/2 memory exhaustion vulnerability.**
* **Technical Impact:**
  + **Server crashes (DoS) or memory corruption leading to RCE.**
* **Mitigation:**
  1. **Upgrade nginx to the latest stable version (1.25.x).**
  2. **Disable unused modules (e.g., HTTP/2 if not needed).**
* **Reference:**[**nginx Security Advisories**](https://nginx.org/en/security_advisories.html)
* **OWASP TOP 10: A06:2021 – Vulnerable and Outdated Components**
* **POC:**



Medium

1. **Vulnerability Name: Cross-Site Scripting (XSS) – Reflected**

* **Description: User-supplied input (e.g., URL parameters) is reflected in the response without sanitization, enabling script execution.**
* **Payload:  <script>alert('XSS')</script>**
* **CVSS: 6.5 (Medium) – AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N**
* **Affected URL:**[**http://testphp.vulnweb.com/listproducts.php?cat=**](http://testphp.vulnweb.com/listproducts.php?cat=)

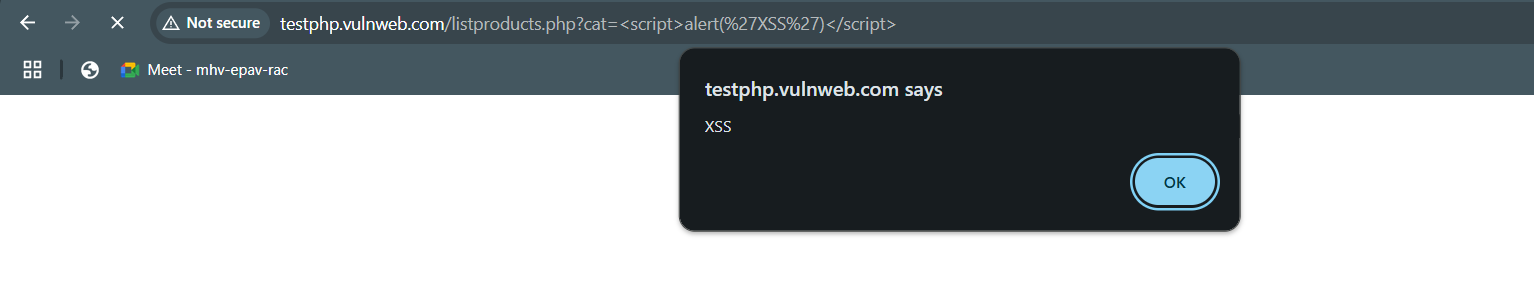
[**http://testphp.vulnweb.com/guestbook.php**](http://testphp.vulnweb.com/guestbook.php)

[**http://testphp.vulnweb.com/search.php?test=query**](http://testphp.vulnweb.com/search.php?test=query)

[**http://testphp.vulnweb.com/userinfo.php**](http://testphp.vulnweb.com/userinfo.php)

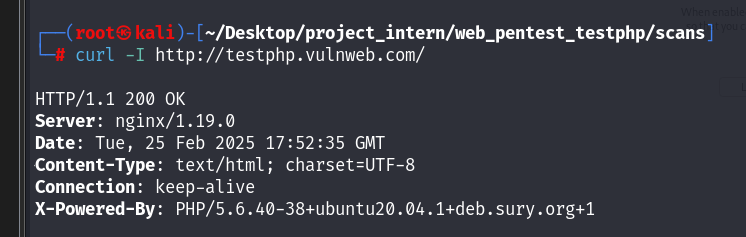
[**http://testphp.vulnweb.com/artists.php?artist=**](http://testphp.vulnweb.com/artists.php?artist=)

* **CVE-ID: N/A**
* **Technical Impact:**
  + **Session hijacking via cookie theft.**
  + **Redirect users to malicious sites (e.g., window.location='http://evil.com').**
* **Mitigation:**
  1. **Encode outputs using context-aware encoding (e.g., HTML, JavaScript).**
  2. **Implement Content Security Policy (CSP) headers (e.g., default-src 'self').**
  3. **Use frameworks like React/Angular that auto-escape content.**
* **Reference:**[**OWASP XSS**](https://owasp.org/www-community/attacks/xss/)
* **OWASP TOP 10: A03:2021 – Injection**
* **POC:**

****

1. **Vulnerability Name: PHP Version Disclosure**

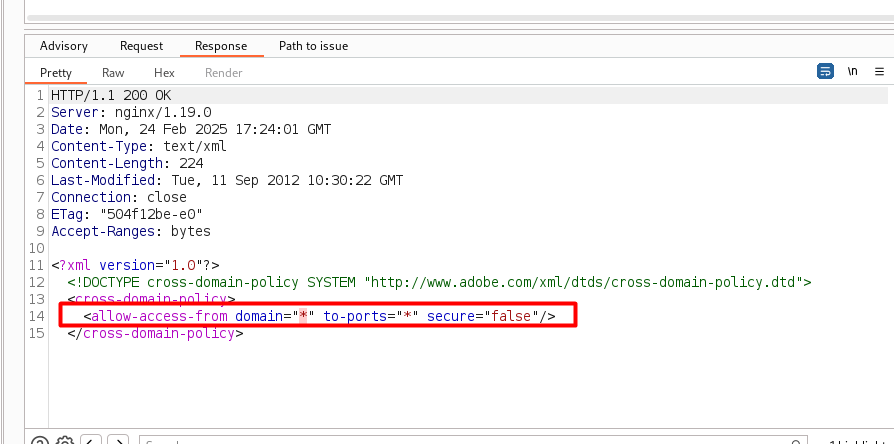
* **Description: Server leaks PHP version via headers or phpinfo() pages.**
* **CVSS: 5.3 (Medium) – AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N**
* **Affected URL: http://testphp.vulnweb.com**
* **CVE-ID: N/A**
* **Technical Impact:**
  + **Targeted exploitation of known PHP vulnerabilities.**
  + **Information disclosure for attack planning.**
* **Mitigation:**
  1. **Disable expose\_php in php.ini.**
  2. **Remove or restrict access to phpinfo.php.**
  3. **Use security headers like X-Content-Type-Options: nosniff.**
* **Reference:**[**OWASP Info Exposure**](https://owasp.org/www-community/vulnerabilities/Information_exposure)
* **OWASP TOP 10: A05:2021 – Security Misconfiguration**
* **POC:**

****

1. **Vulnerability Name: Flash Cross-Domain Policy**

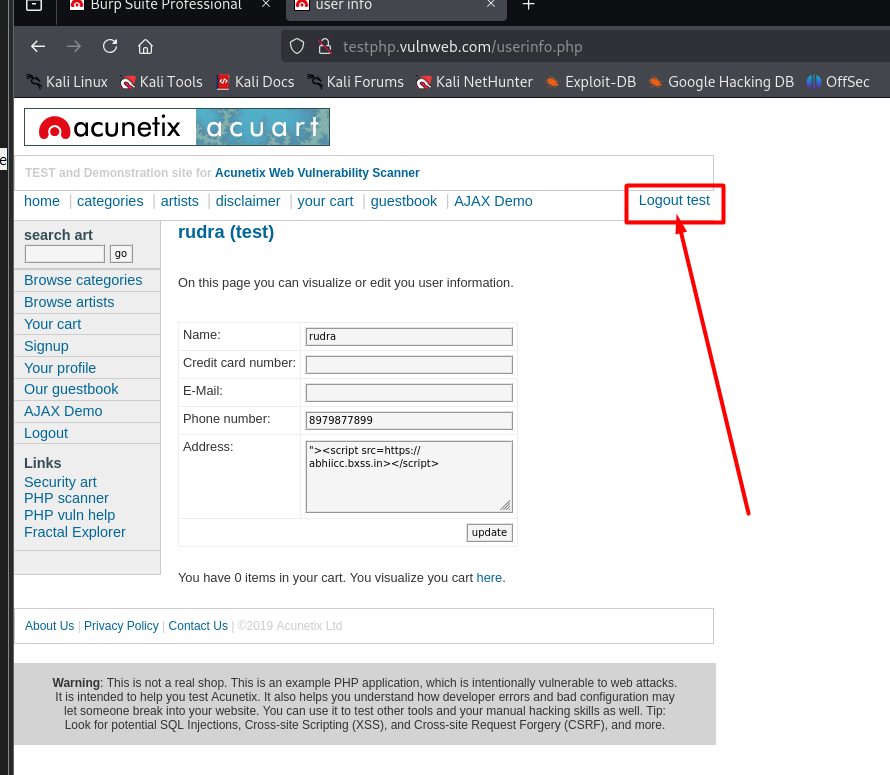
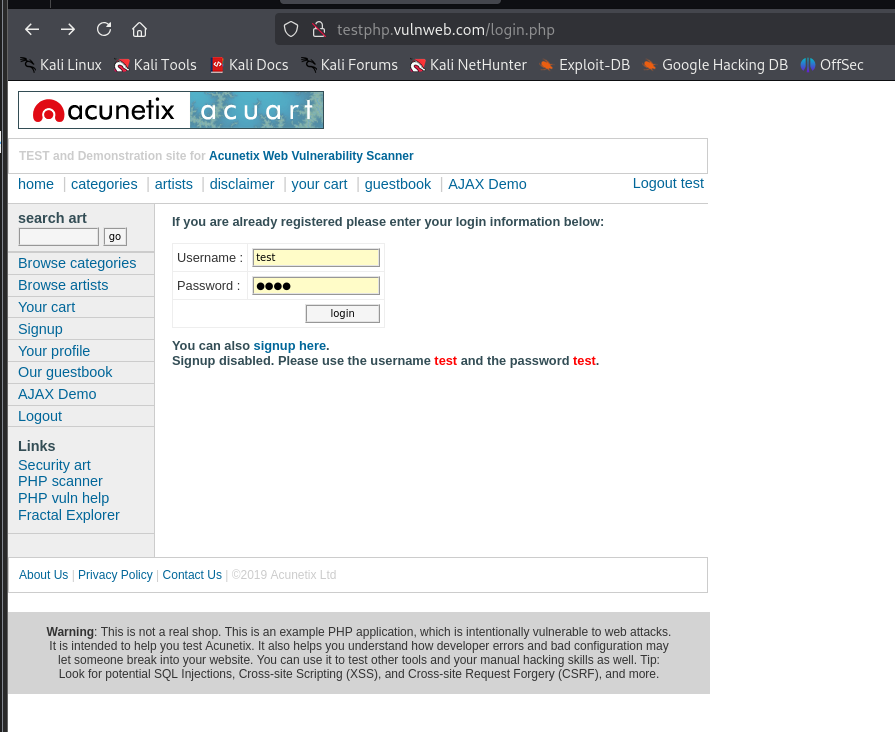
* **Description: Misconfigured crossdomain.xml allows arbitrary domains to access resources via Flash.**
* **CVSS: 6.5 (Medium) – AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N**
* **Affected URL:** [**http://testphp.vulnweb.com/crossdomain.xml**](http://testphp.vulnweb.com/crossdomain.xml)
* **CVE-ID: N/A**
* **Technical Impact:**
  + **Cross-domain data theft (e.g., stealing cookies or API keys).**
  + **Unauthorized actions via Flash-based requests.**
* **Mitigation:**
  1. **Restrict domains in crossdomain.xml (e.g., <allow-access-from domain="\*.example.com"/>).**
  2. **Disable Flash entirely; migrate to HTML5.**
  3. **Use Subresource Integrity (SRI) for external scripts.**
* **Reference:**[**Adobe Cross-Domain Policy**](https://helpx.adobe.com/flash/kb/flash-player-security-policy-cross-domain.html)
* **OWASP TOP 10: A05:2021 – Security Misconfiguration**
* **POC:**

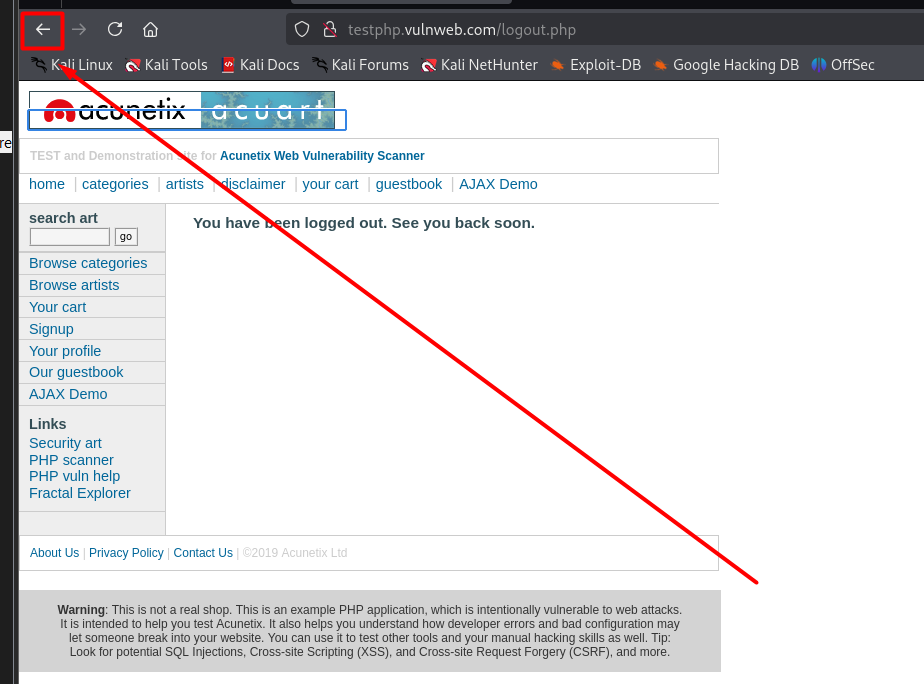
****

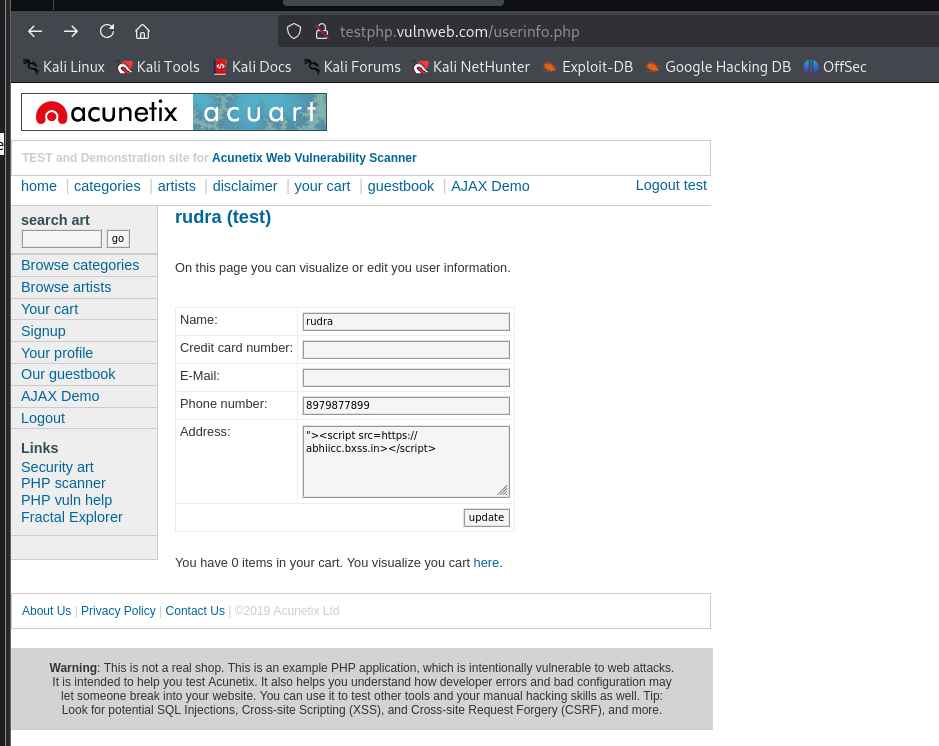
****

1. **Vulnerability Name: Improper Logout Management (no functionality access)**

* **Description: Sessions remain active after logout, allowing reuse of session tokens.**
* **CVSS: 5.5 (Medium) – AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N**
* **Affected URL:**[**http://testphp.vulnweb.com/login.php**](http://testphp.vulnweb.com/login.php)**,** [**http://testphp.vulnweb.com/logout.php**](http://testphp.vulnweb.com/logout.php)**,** [**http://testphp.vulnweb.com/userinfo.php**](http://testphp.vulnweb.com/userinfo.php)
* **CVE-ID: N/A**
* **Technical Impact:**
  + **Session hijacking via stolen cookies.**
* **Mitigation:**
  1. **Invalidate sessions server-side on logout.**
  2. **Regenerate session IDs post-login/logout.**
  3. **Set short session timeouts (e.g., 15 minutes).**
* **Reference:**[**OWASP Session Management**](https://owasp.org/www-project-top-ten/2017/A2_2017-Broken_Authentication)
* **OWASP TOP 10: A07:2021 – Identification and Authentication Failures**
* **POC:**

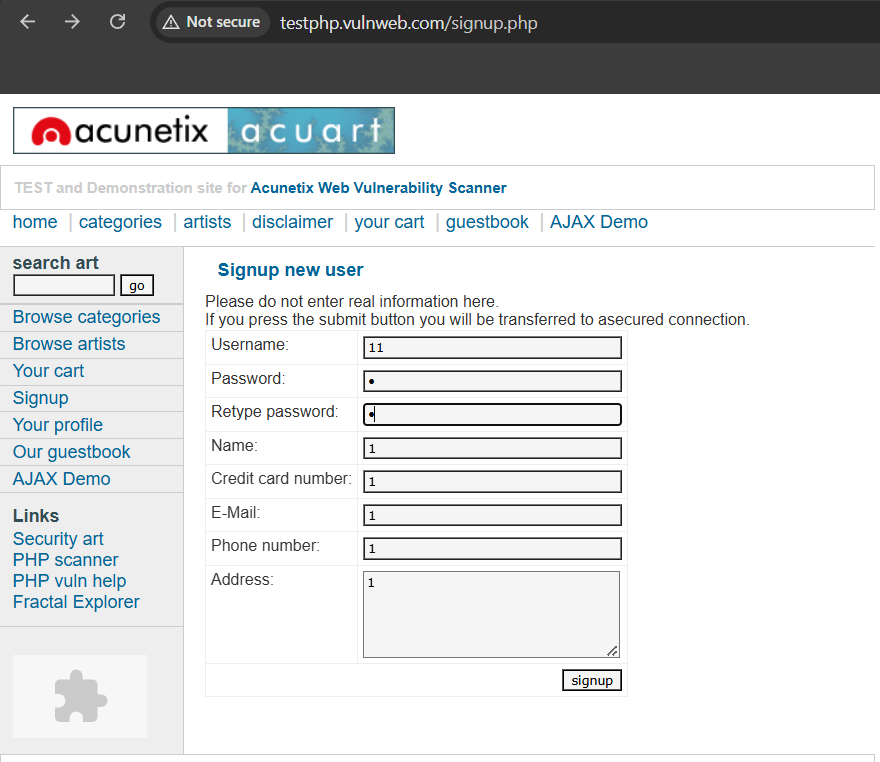
****

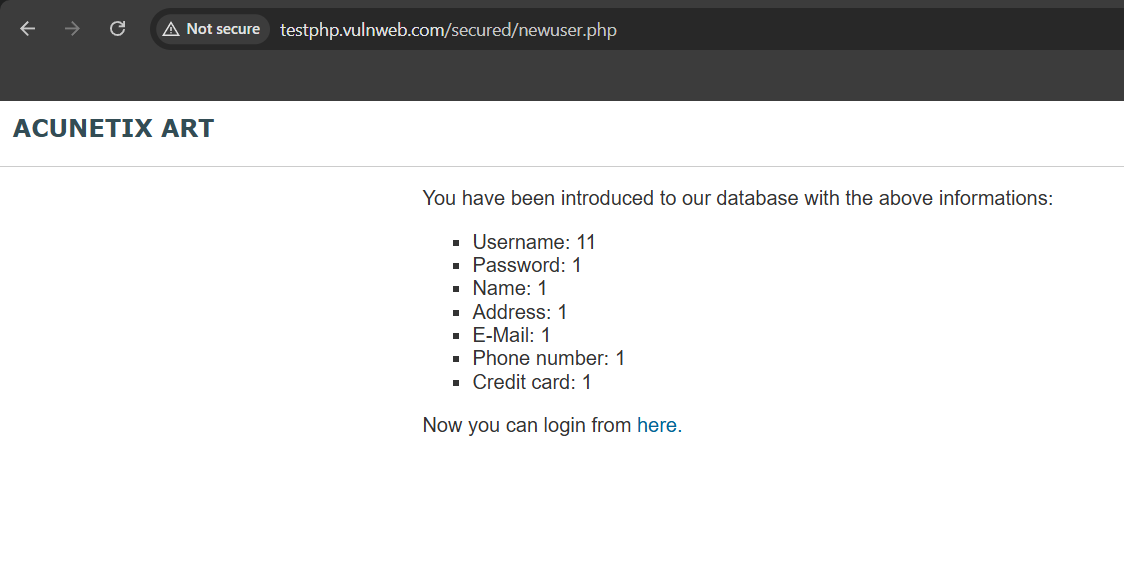
****

****

1. **Vulnerability Name: Weak Password Policy**

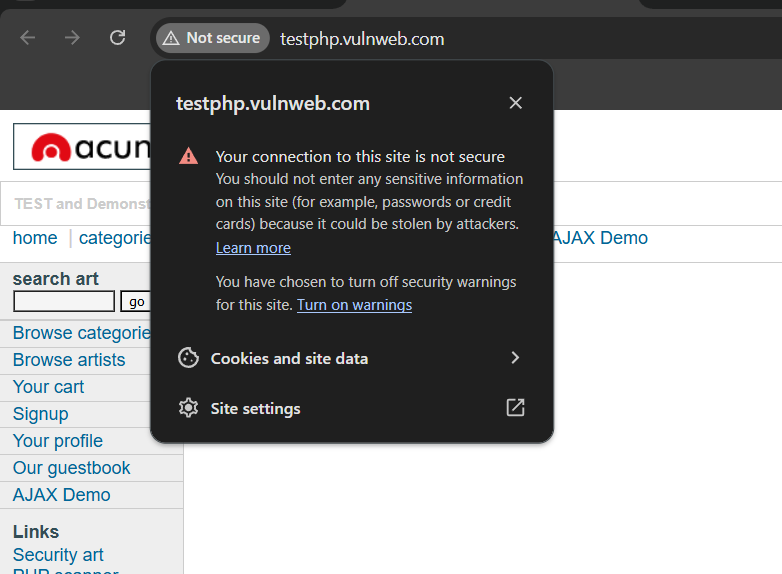
* **Description: No enforcement of password complexity or length.**
* **CVSS: 6.5 (Medium) – AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N**
* **Affected URL:**[**http://testphp.vulnweb.com/signup.php**](http://testphp.vulnweb.com/signup.php)
* **CVE-ID: N/A**
* **Technical Impact:**
  + **Brute-force attacks using weak passwords (e.g., "password123").**
* **Mitigation:**
  1. **Enforce 12+ characters with mix of uppercase, numbers, and symbols.**
  2. **Integrate with breached password databases (e.g., HaveIBeenPwned API).**
  3. **Require multi-factor authentication (MFA).**
* **Reference:**[**OWASP Password Policy**](https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html)
* **OWASP TOP 10: A07:2021 – Identification and Authentication Failures**
* **POC:**

****

****

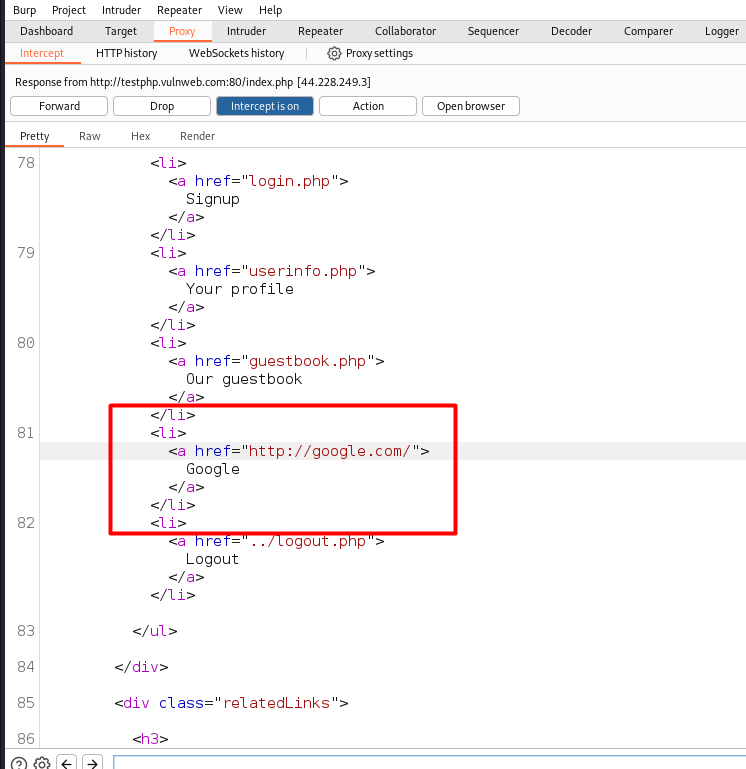
1. **Vulnerability Name: Insecure Connection (HTTP)**

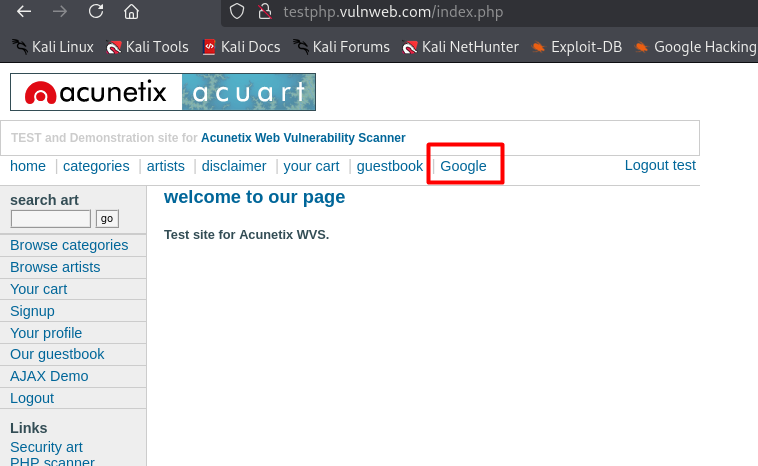
* **Description: The site uses HTTP instead of HTTPS, exposing data in transit.**
* **CVSS: 5.3 (Medium) – AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N**
* **Affected URL: http://testphp.vulnweb.com**
* **CVE-ID: N/A**
* **Technical Impact:**
  + **Session cookie theft via MITM attacks.**
* **Mitigation:**
  1. **Redirect HTTP to HTTPS using 301 redirects.**
  2. **Deploy TLS 1.3 with modern ciphers (e.g., AES-GCM).**
  3. **Submit the domain to the HSTS preload list.**
* **Reference:**[**OWASP Transport Layer Protection**](https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure)
* **OWASP TOP 10: A02:2021 – Cryptographic Failures**
* **POC:**

****

1. **Vulnerability Name: Response Manipulation**

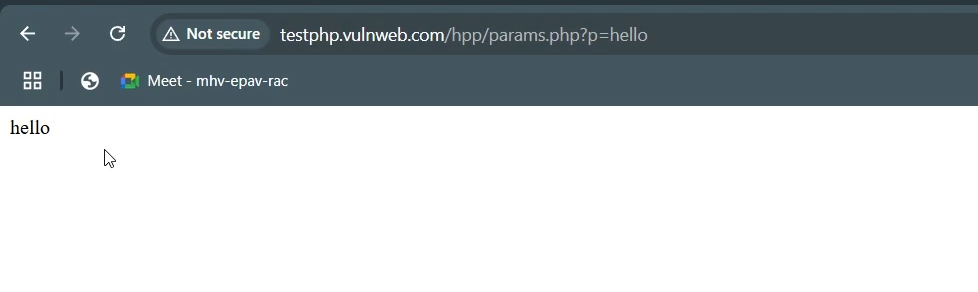
* **Description: Server responses can be altered to inject malicious content.**
* **CVSS: 6.5 (Medium) – AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N**
* **Affected URL:**[**http://testphp.vulnweb.com/index.php**](http://testphp.vulnweb.com/index.php)
* **CVE-ID: N/A**
* **Technical Impact:**
  + **Phishing via fake login pages injected into responses.**
* **Mitigation:**
  1. **Validate responses using HMAC signatures.**
  2. **Implement Subresource Integrity (SRI) for scripts/stylesheets.**
  3. **Add security headers like X-Content-Type-Options: nosniff.**
* **Reference:**[**OWASP Content Spoofing**](https://owasp.org/www-community/attacks/Content_Spoofing)
* **OWASP TOP 10: A03:2021 – Injection**
* **POC:**

****

****

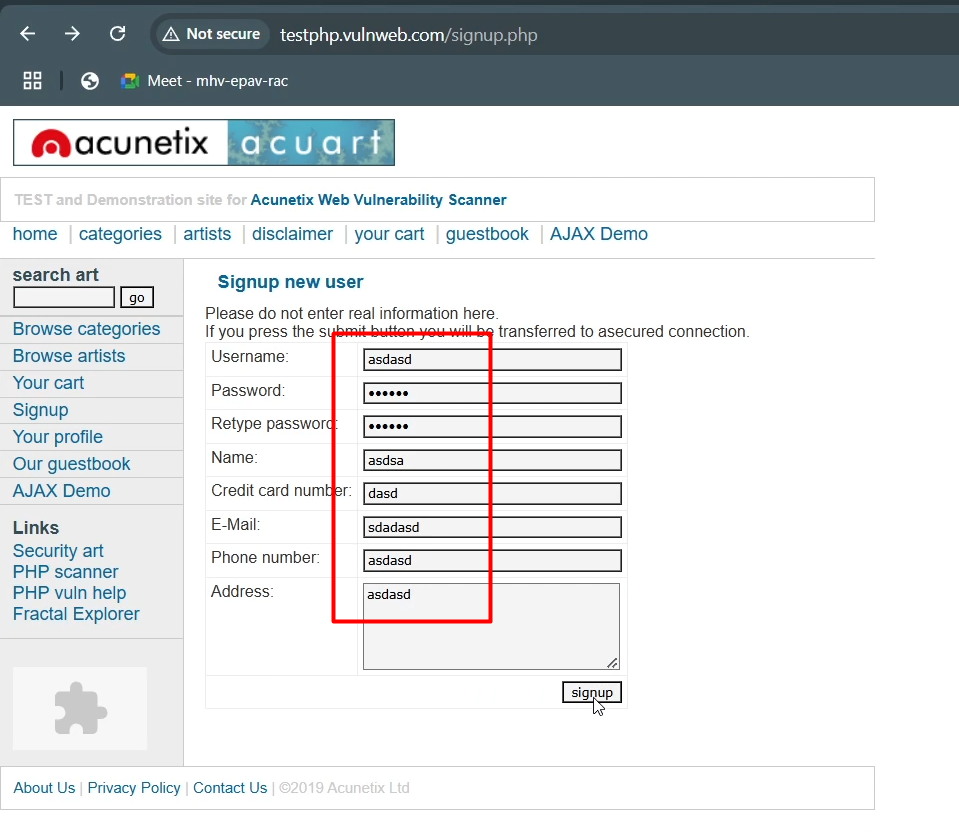
1. **Vulnerability Name: Text Injection**

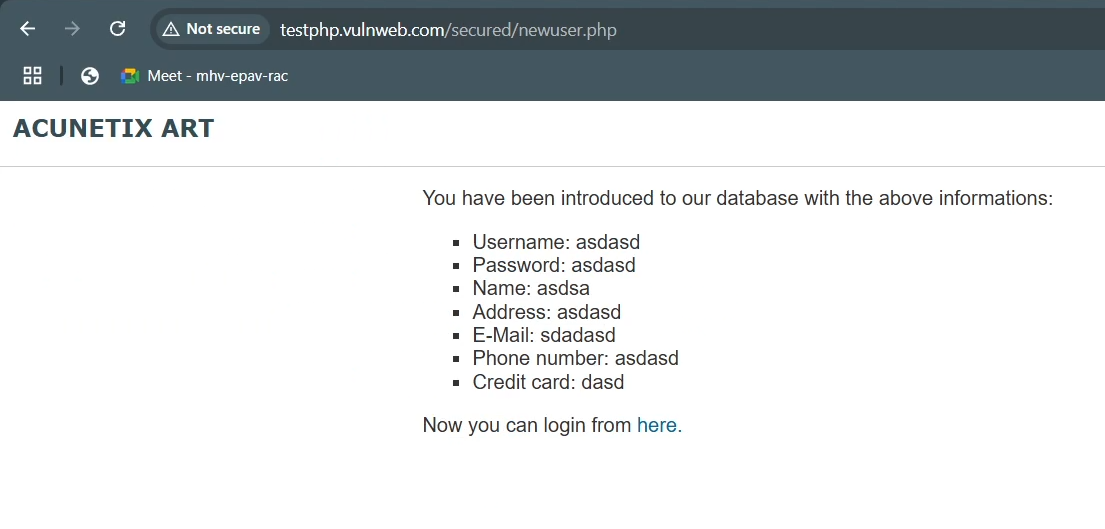
* **Description: User input is reflected without sanitization, allowing arbitrary text injection.**
* **CVSS: 5.3 (Medium) – AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N**
* **Affected URL:**[**http://testphp.vulnweb.com/hpp/params.php?p=**](http://testphp.vulnweb.com/hpp/params.php?p=)
* **CVE-ID: N/A**
* **Technical Impact:**
  + **Phishing or defacement (e.g., fake system alerts).**
* **Mitigation:**
  1. **Encode user input for HTML, URL, and JavaScript contexts.**
  2. **Use regex to block malicious patterns (e.g., admin, password).**
  3. **Log and audit user inputs for injection attempts.**
* **Reference:**[**OWASP XSS**](https://owasp.org/www-community/attacks/xss/)
* **OWASP TOP 10: A03:2021 – Injection**
* **POC:**

****

1. **Vulnerability Name: Improper Input Validation**

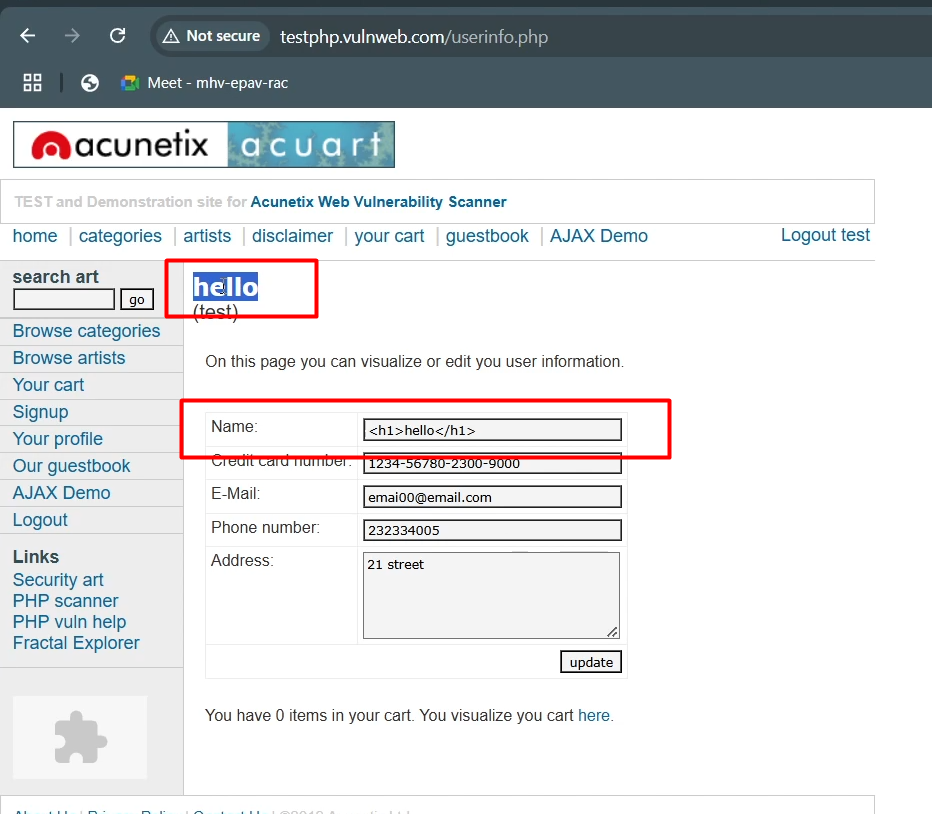
* **Description: Missing validation for input types, lengths, or formats.**
* **CVSS: 6.5 (Medium) – AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N**
* **Affected URL:**[**http://testphp.vulnweb.com/signup.php**](http://testphp.vulnweb.com/signup.php)
* **CVE-ID: N/A**
* **Technical Impact:**
  + **Buffer overflows, command injection, or data corruption.**
* **Mitigation:**
  1. **Validate input types (e.g., enforce int for numeric fields).**
  2. **Restrict input lengths (e.g., 50 characters for usernames).**
  3. **Use OWASP ESAPI for input sanitization.**
* **Reference:**[**OWASP Input Validation**](https://owasp.org/www-community/Improper_Input_Validation)
* **OWASP TOP 10: A03:2021 – Injection**
* **POC:**

****

****

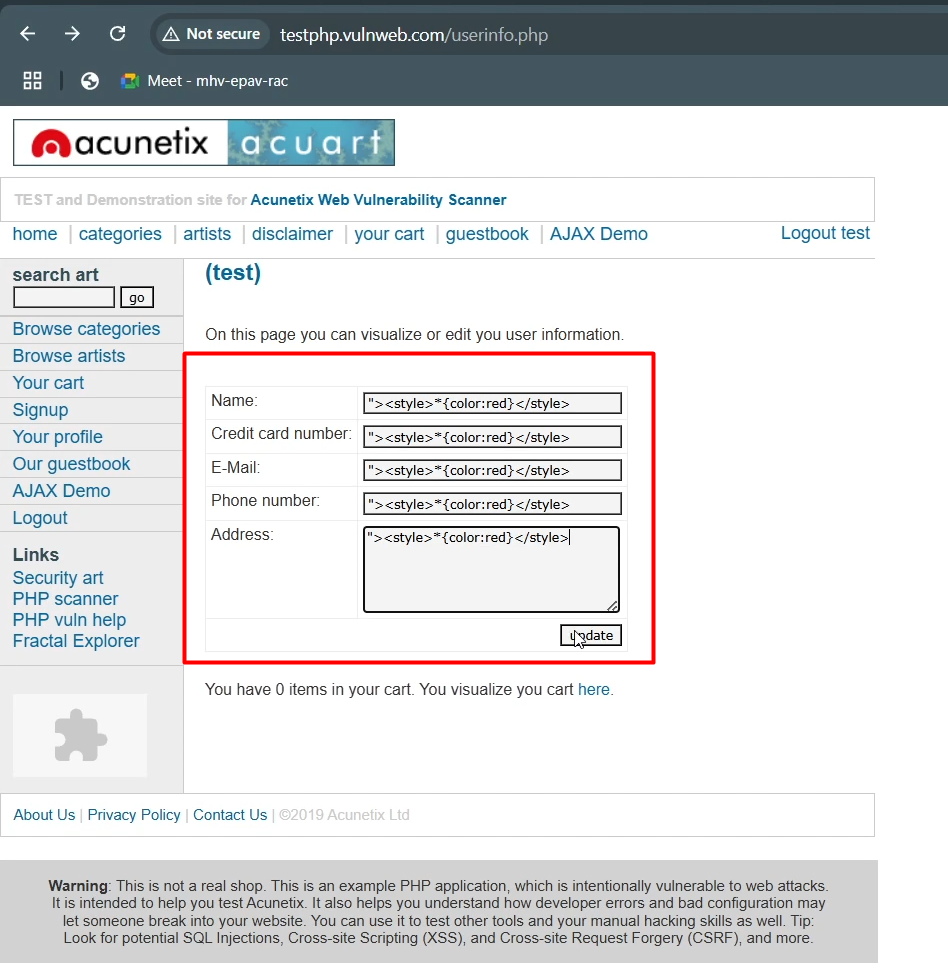
1. **Vulnerability Name: HTML Injection**

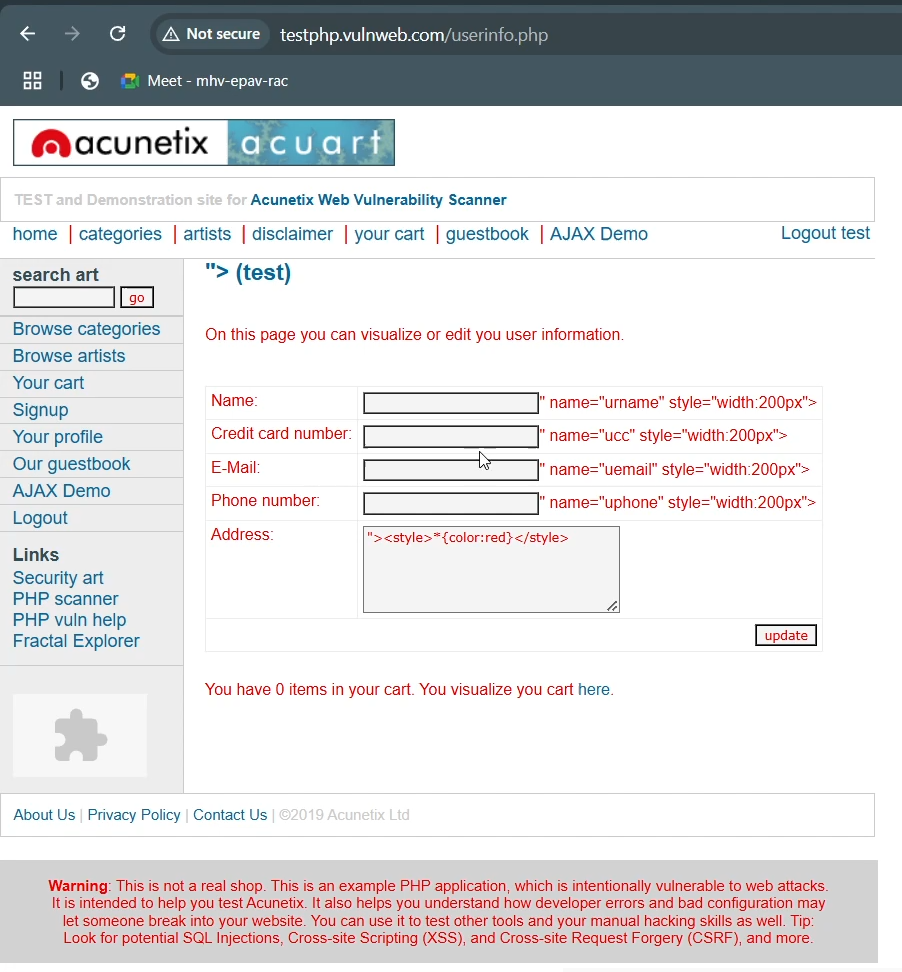
* **Description: User input rendered as raw HTML, allowing UI manipulation.**
* **CVSS: 6.5 (Medium) – AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N**
* **Affected URL:**[**http://testphp.vulnweb.com/userinfo.php**](http://testphp.vulnweb.com/userinfo.php)
* **CVE-ID: N/A**
* **Technical Impact:**
  + **Defacement (e.g., injecting fake news or banners).**
* **Mitigation:**
  1. **Sanitize HTML with libraries like DOMPurify.**
  2. **Use CSP headers to block inline scripts (unsafe-inline).**
  3. **Leverage framework protections (e.g., React’s JSX escaping).**
* **Reference:**[**OWASP XSS**](https://owasp.org/www-community/attacks/xss/)
* **OWASP TOP 10: A03:2021 – Injection**
* **POC:**

****

1. **Vulnerability Name: CSS Injection**

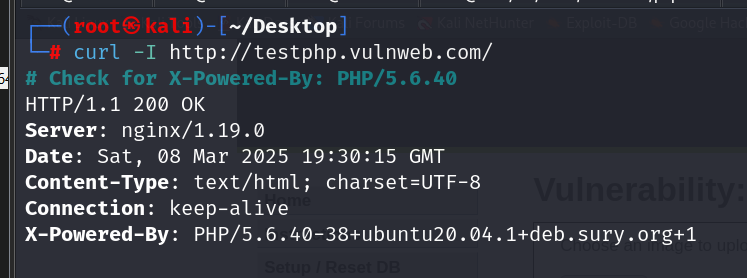
* **Description: User-controlled CSS modifies page styles (e.g., hiding buttons).**
* **CVSS: 4.3 (Medium) – AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:N**
* **Affected URL: http://testphp.vulnweb.com/userinfo.php**
* **CVE-ID: N/A**
* **Technical Impact:**
  + **UI redressing (e.g., hiding logout buttons).**
* **Mitigation:**
  1. **Sanitize CSS inputs (e.g., allow only color, font-size).**
  2. **Isolate user-generated content in sandboxed iframes.**
  3. **Use CSP style-src to restrict CSS sources.**
* **Reference:**[**OWASP XSS**](https://owasp.org/www-community/attacks/xss/)
* **OWASP TOP 10: A03:2021 – Injection**
* **POC:**

****

****

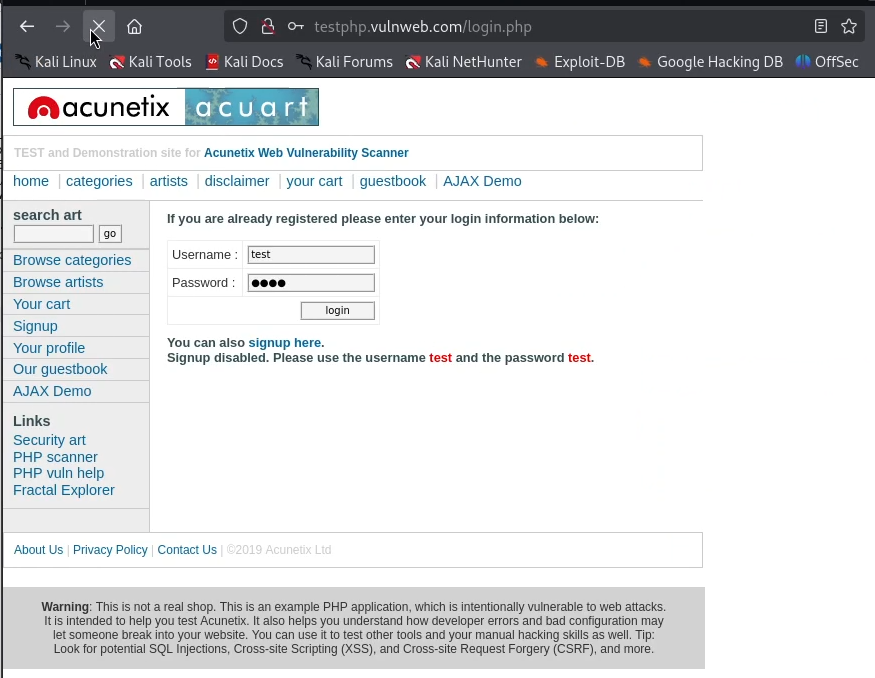
1. **Vulnerability Name: Missing X-Content-Type-Options Header**

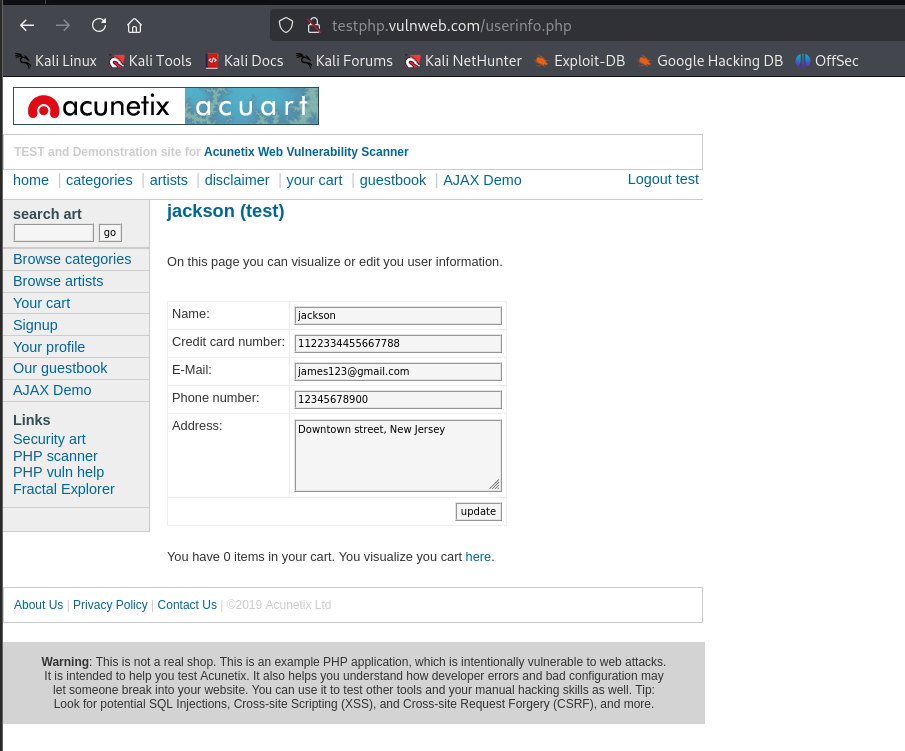
* **Description: The X-Content-Type-Options: nosniff header is missing, allowing browsers to interpret files as unintended MIME types.**
* **CVSS: 5.3 (Medium) – AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N**
* **Affected URL:**[**http://testphp.vulnweb.com/**](http://testphp.vulnweb.com/)
* **CVE-ID: N/A**
* **Technical Impact:**
  + **MIME sniffing attacks (e.g., rendering HTML as JavaScript).**
  + **Cross-site scripting (XSS) via uploaded files.**
* **Mitigation:**
  1. **Add X-Content-Type-Options: nosniff to HTTP responses.**
  2. **Validate MIME types for uploaded files.**
* **Reference:**[**MDN X-Content-Type-Options**](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options)
* **OWASP TOP 10: A05:2021 – Security Misconfiguration**
* **POC:**

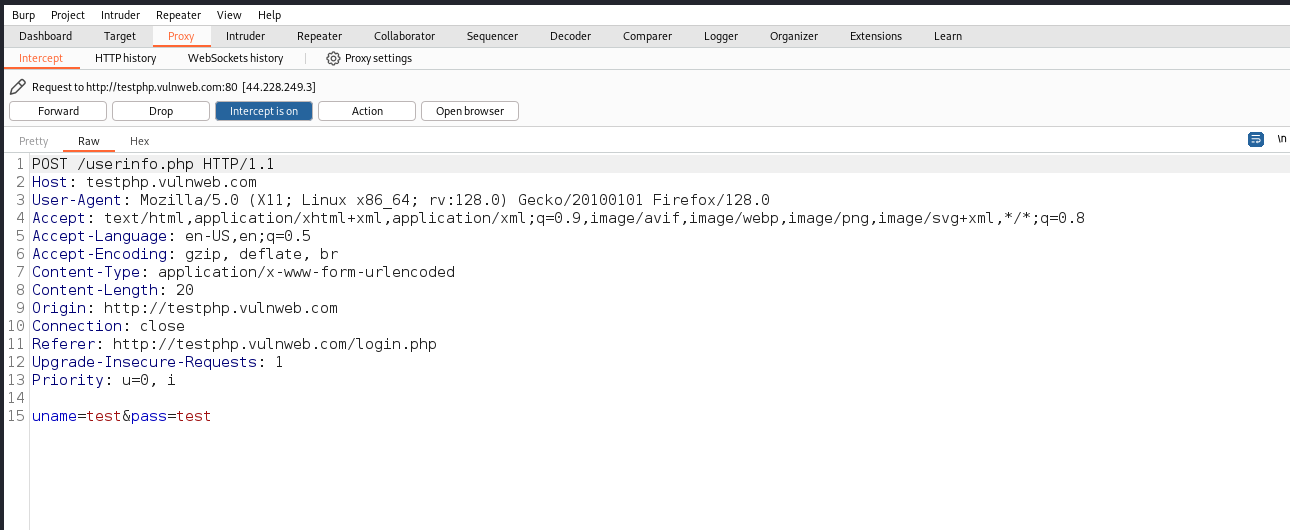


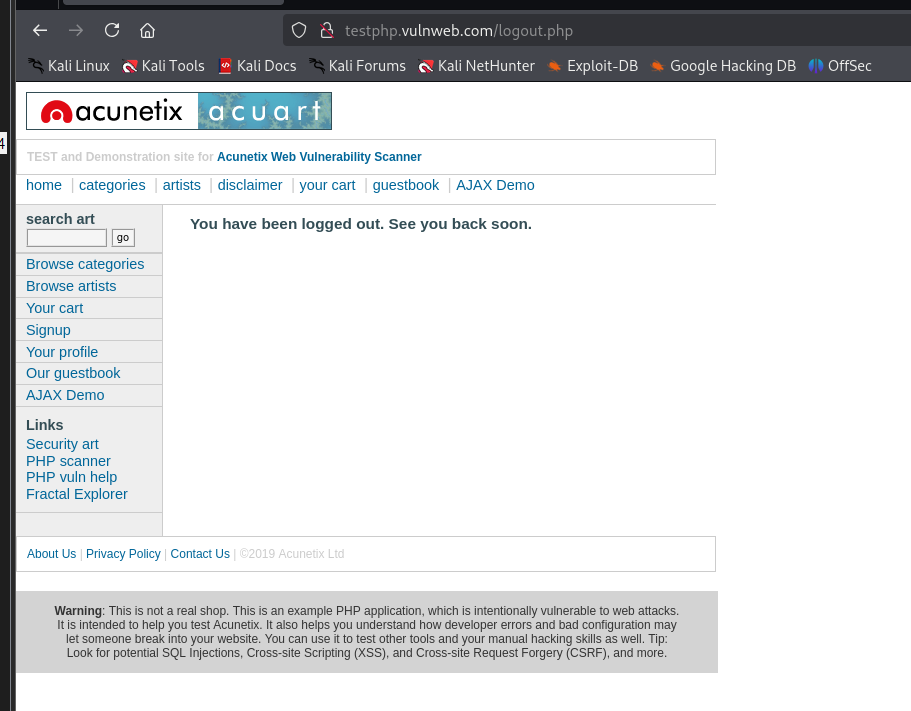
1. **Vulnerability Name: Improper Session Management**

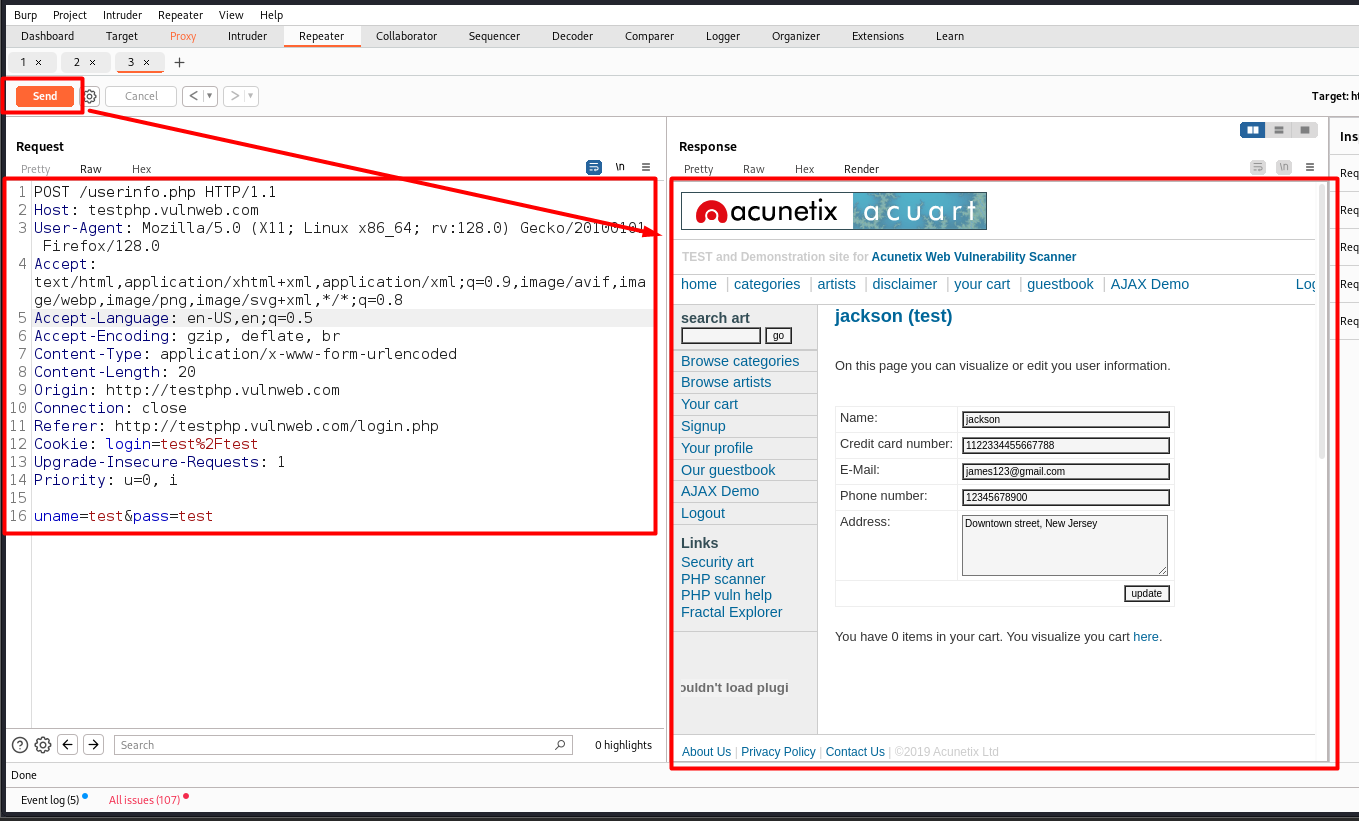
* **Description: Weak session expiration, fixation, or token rotation.**
* **CVSS: 6.5 (Medium) – AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N**
* **Affected URL:**[**http://testphp.vulnweb.com/login.php**](http://testphp.vulnweb.com/login.php)
* **CVE-ID: N/A**
* **Technical Impact:**
  + **Session hijacking via stolen or reused tokens.**
* **Mitigation:**
  1. **Set session cookies with Secure, HttpOnly, and SameSite attributes.**
  2. **Invalidate sessions after logout or inactivity.**
  3. **Use OAuth2/OpenID Connect for token management.**
* **Reference:**[**OWASP Session Management**](https://owasp.org/www-project-top-ten/2017/A2_2017-Broken_Authentication)
* **OWASP TOP 10: A07:2021 – Identification and Authentication Failures**
* **POC:**

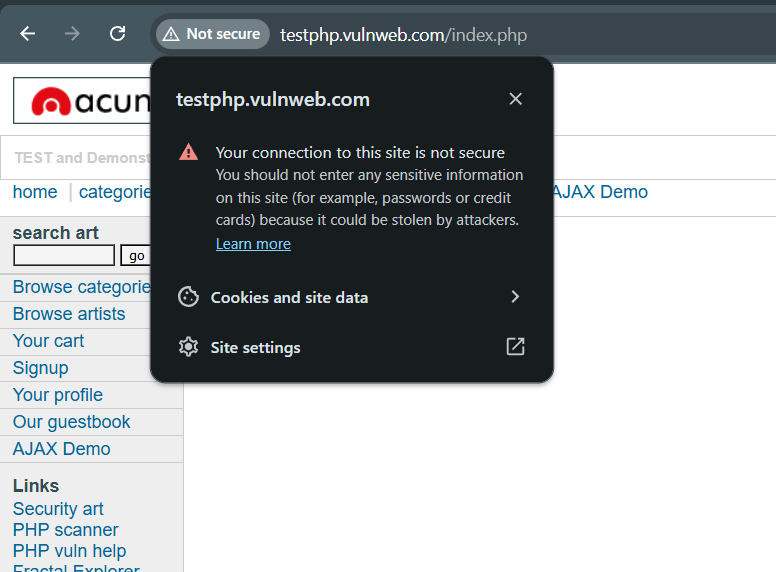
****

****

****

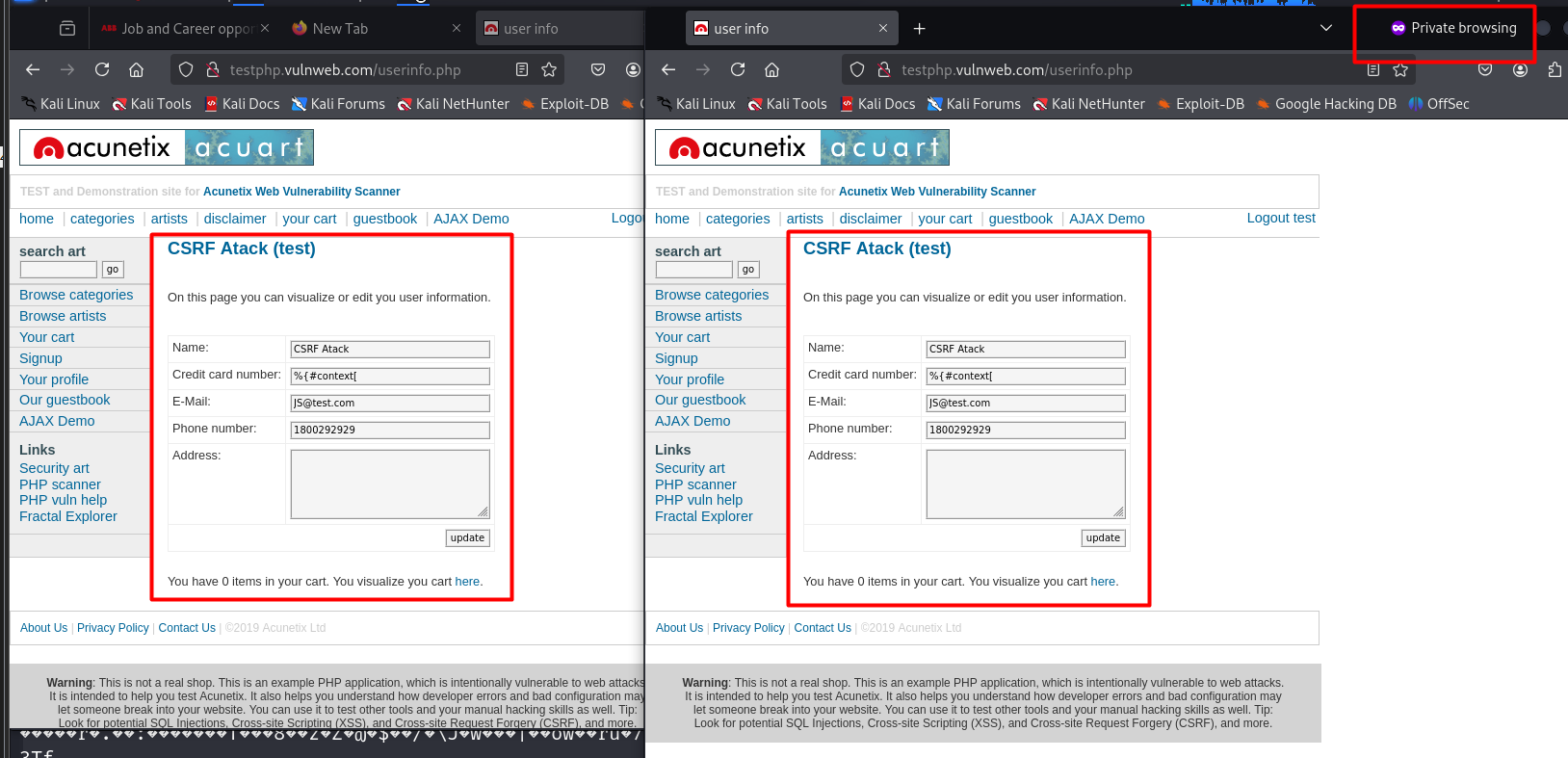
****

****



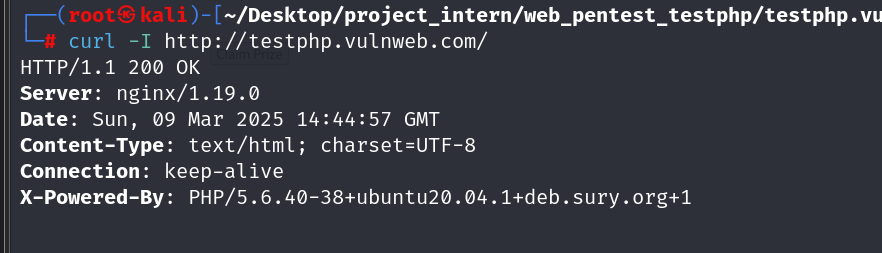
1. **Vulnerability Name: Concurrent Login Vulnerability**

* **Description: Multiple active sessions allowed for the same user.**
* **CVSS: 5.5 (Medium) – AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N**
* **Affected URL: http://testphp.vulnweb.com/userinfo.php**
* **CVE-ID: N/A**
* **Technical Impact:**
  + **Account sharing or hijacking via simultaneous logins.**
* **Mitigation:**
  1. **Enforce single-session-per-user policies.**
  2. **Notify users of new logins via email/SMS.**
  3. **Invalidate old sessions on new login.**
* **Reference:**[**OWASP Session Management**](https://owasp.org/www-project-top-ten/2017/A2_2017-Broken_Authentication)
* **OWASP TOP 10: A07:2021**
* **POC:**



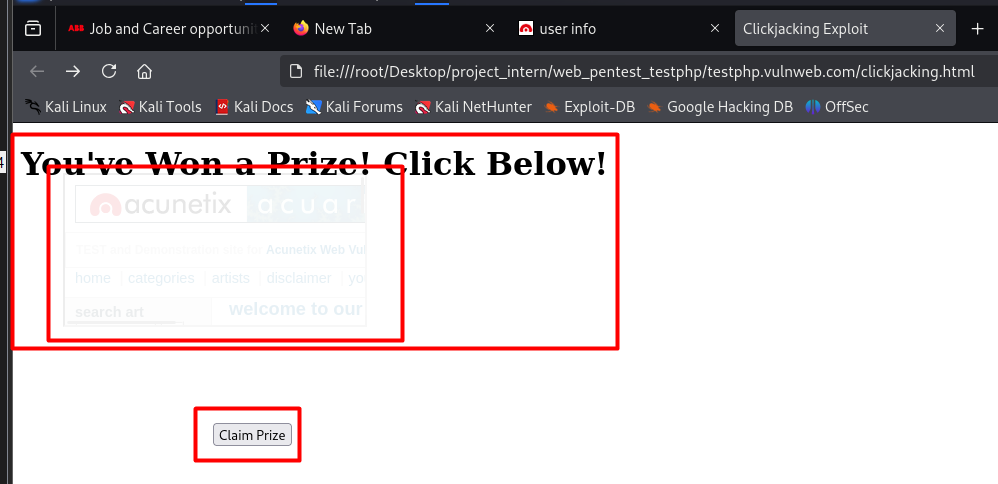
1. **Vulnerability Name: Security Headers Missing**

* **Description: Critical security headers are missing in HTTP responses, particularly on pages handling form submissions (e.g., login, contact forms). This exposes the application to attacks like clickjacking, MIME sniffing, and cross-site scripting (XSS).**
* **CVSS: 5.5 (Medium) – AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N**
* **Affected URL: http://testphp.vulnweb.com**
* **CVE-ID: N/A**
* **Technical Impact:**
  + **Clickjacking: Attackers can embed the page in an iframe to trick users into performing unintended actions (e.g., submitting forms).**
  + **XSS Exploitation: Missing Content-Security-Policy (CSP) allows execution of injected scripts.**
  + **MIME Sniffing: Browsers may misinterpret content types, leading to data leaks or code execution.**
  + **HTTP Downgrade Attacks: Missing Strict-Transport-Security (HSTS) enables MITM attacks over HTTP.**
* **Mitigation:**
  1. **Add Missing Headers:**
     + **Content-Security-Policy: Restrict sources for scripts/styles (e.g., default-src 'self').**
     + **Strict-Transport-Security: Enforce HTTPS (e.g., max-age=31536000; includeSubDomains).**
     + **X-Content-Type-Options: nosniff: Prevent MIME sniffing.**
     + **X-Frame-Options: DENY or SAMEORIGIN: Block clickjacking.**
     + **Referrer-Policy: Limit referrer data leakage (e.g., strict-origin-when-cross-origin).**
  2. **Automate Header Configuration:**
     + **Use server middleware (e.g., Apache/Nginx) or frameworks (e.g., Django, Express.js) to enforce headers globally.**
  3. **Regular Audits:**
     + **Scan headers using tools like**[**SecurityHeaders.com**](https://securityheaders.com/)**or**[**Observatory by Mozilla**](https://observatory.mozilla.org/)**.**
* **Reference:**
  + [**OWASP Secure Headers**](https://owasp.org/www-project-secure-headers/)
  + [**MDN HTTP Headers**](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers)
* **OWASP TOP 10: A05:2021 – Security Misconfiguration**
* **POC:**



1. **Vulnerability Name: Clickjacking**

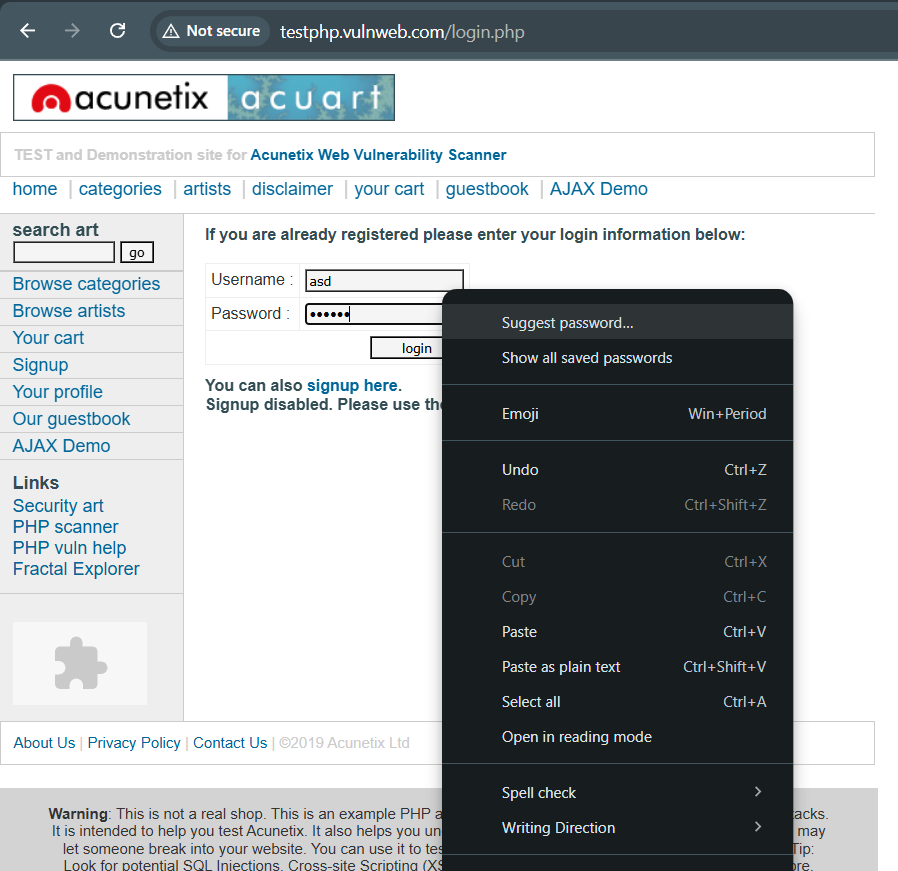
* **Description: The application lacks protections against being embedded in malicious iframes, allowing attackers to trick users into clicking hidden UI elements.**
* **CVSS: 5.3 (Medium) – AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:L/A:N**
* **Affected URL:**[**http://testphp.vulnweb.com/**](http://testphp.vulnweb.com/)
* **CVE-ID: N/A**
* **Technical Impact:**
  + **Unauthorized actions (e.g., transferring funds, changing settings) via deceptive overlays.**
* **Mitigation:**
  1. **Add the X-Frame-Options: DENY or SAMEORIGIN header.**
  2. **Implement Content-Security-Policy: frame-ancestors 'none'.**
  3. **Use JavaScript framebusting scripts (e.g., if (top != self) top.location = self.location;).**
* **Reference:**[**OWASP Clickjacking**](https://owasp.org/www-community/attacks/Clickjacking)
* **OWASP TOP 10: A05:2021 – Security Misconfiguration**
* **POC:**



Low

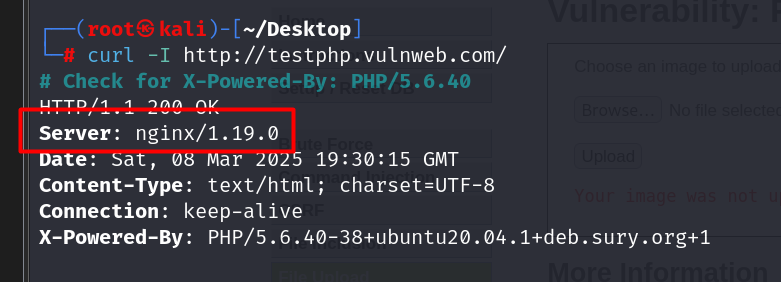
1. **Vulnerability Name: Right Click enable**

* **Description: The application allows users to right-click and view page source, potentially exposing sensitive data or logic.**
* **CVSS: 3.1 (Low) – AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N**
* **Affected URL:**[**http://testphp.vulnweb.com/login.php**](http://testphp.vulnweb.com/login.php)
* **CVE-ID: N/A**
* **Technical Impact:**
  + **Source code or sensitive data leakage (e.g., API keys, hidden endpoints).**
* **Mitigation:**
  1. **Avoid client-side secrets: Never hardcode credentials in HTML/JS.**
  2. **Obfuscate code: Use minification or tools like JavaScript Obfuscator.**
  3. **Educate users: Disabling right-click is not security; focus on server-side protections.**
* **Reference:**[**OWASP Client-Side Security**](https://owasp.org/www-community/controls/Client_Side_Security)
* **OWASP TOP 10: A05:2021 – Security Misconfiguration**
* **POC:**



1. **Vulnerability Name: Server Header Disclosure**

* **Description: The server leaks its version (nginx/1.19.0) in HTTP headers, aiding attackers in exploiting known vulnerabilities.**
* **CVSS: 3.7 (Low) – AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N**
* **Affected URL:**[**http://testphp.vulnweb.com/**](http://testphp.vulnweb.com/)
* **CVE-ID: N/A**
* **Technical Impact:**
  + **Reconnaissance for targeted attacks.**
* **Mitigation:**
  1. **Remove/obfuscate the Server header in nginx config:**
  2. **Use third-party modules like ngx\_headers\_more to customize headers.**
* **Reference:**[**OWASP Information Exposure**](https://owasp.org/www-community/attacks/Information_exposure)
* **OWASP TOP 10: A05:2021 – Security Misconfiguration**
* **POC:**



Info

1. **Conclusion**

The comprehensive assessment of http://testphp.vulnweb.com revealed multiple critical security vulnerabilities that could be exploited by attackers to gain unauthorized access, exfiltrate sensitive information, or compromise the system’s integrity and availability. Key issues include SQL Injection (including Blind SQLi), Stored XSS, CSRF, Local File Inclusion (LFI), Path Traversal, and Broken Authentication mechanisms. Additionally, the use of an outdated PHP version (5.6.40) introduces serious risks of Remote Code Execution (RCE) due to known CVEs.

High-severity exposures such as plaintext credential transmission and storage, absence of HTTPS, and leaked credentials via accessible files amplify the overall risk profile. Many of these vulnerabilities map directly to the OWASP Top 10, demonstrating a systemic lack of secure coding practices and input validation.

Immediate remediation actions are strongly advised, including patching outdated software, enforcing secure communication protocols (HTTPS), implementing strong authentication and session management controls, and sanitizing user inputs. Additionally, applying Web Application Firewalls (WAF), regular security audits, and security-aware development practices will significantly reduce attack surface and enhance overall resilience.