KLEAP

CYBERSECURITY

Web Application Security Report

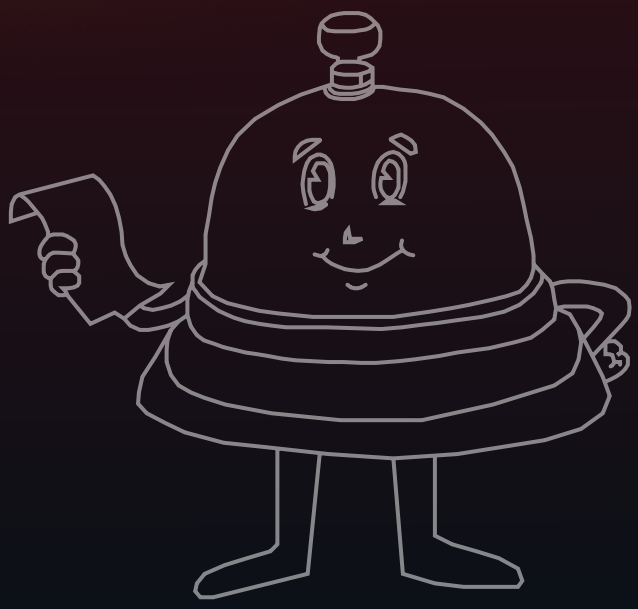


Table of Contents

Statement of Confidentiality

This pentest report contains confidential and proprietary information belonging to KLEAP Technologies Pvt. Ltd. and Client. It is intended solely for the use of the Client and KLEAP Technologies Pvt. Ltd. The information provided within this report should not be disclosed, distributed, or shared with any third parties without the explicit written consent of both KLEAP Technologies Pvt. Ltd. and Client. Any unauthorized use or disclosure of this information is strictly prohibited and may result in legal action.

Executive Summary

Client engaged KLEAP Technologies Pvt. Ltd. to perform penetration testing of the APIs. The primary goal of this API penetration testing project was to identify any potential areas of concern associated with the API in its current state and determine the extent to which the system may be breached by an attacker possessing a particular skill and motivation. The assessment was performed in accordance with the “best-in-class” practices as defined by ISECOM's Open Source Security Testing Methodology Manual (OSSTMM) and Open Web Application Security Project (OWASP).

KLEAP Technologies Pvt. Ltd. conducted the penetration testing during the period of March 20th, 2024 to April 11th, 2024. All testing activities were performed on the staging environment provided by the customer and completely isolated from the production data. While performing the testing activities, KLEAP Technologies Pvt. Ltd. emulated an external attacker without prior knowledge of the environment. To test the user-authenticated area and privilege escalation vulnerabilities, the customer supplied KLEAP Technologies Pvt. Ltd. credentials for several registered user and admin accounts.

Scoping and Time Limitations

Scoping during the engagement did not permit denial of service or social engineering across all testing components.

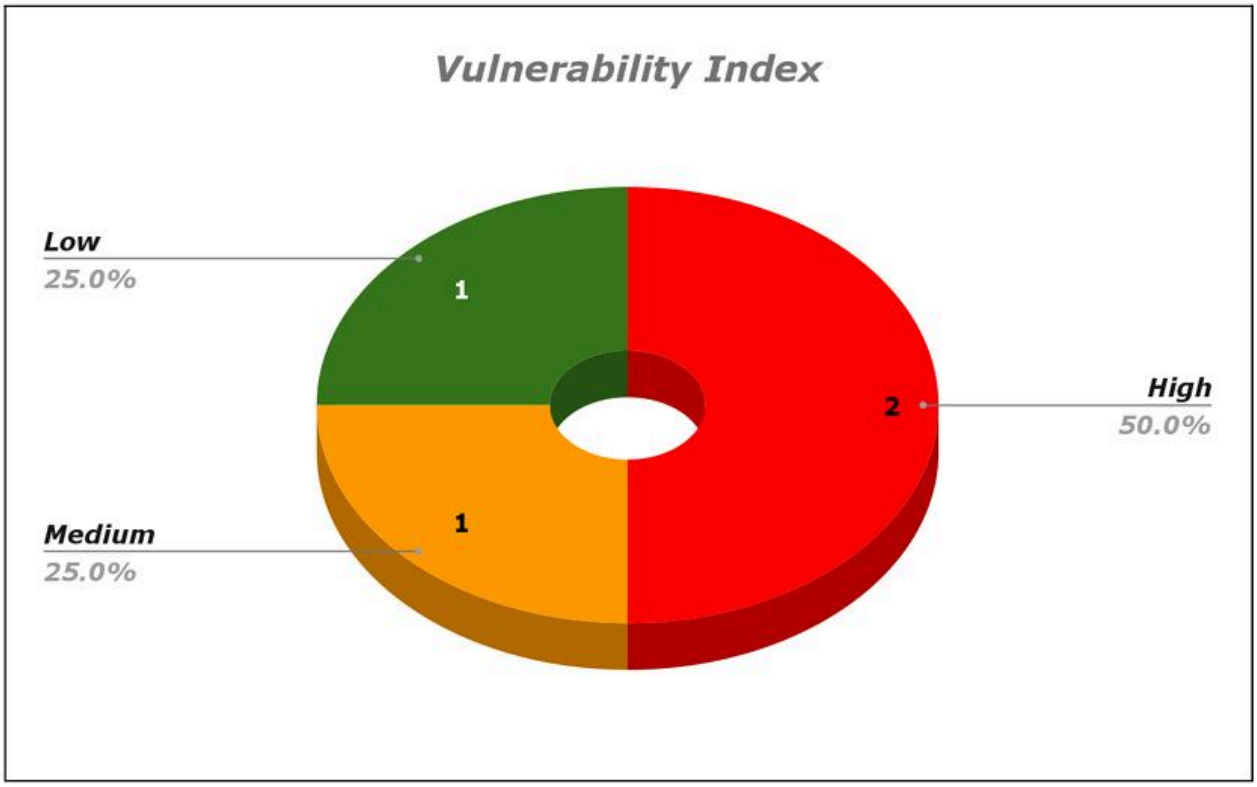
Time limitations were in place for testing. API penetration testing was permitted for seven (7) business days.

Testing Summary

(Overall Summary of the findings)

Scope	Critical	High	Medium	Low	Info	Total
Client AD	0	2	1	1	0	4

Table 1: Findings per asset



Vulnerability Index

Vulnerability Status

Sr. No.	Vulnerability	Severity	Status
1	SQL Injection	HIGH	OPEN
2	Cross-Site Scripting	HIGH	OPEN
3	Directory Traversal	MEDIUM	OPEN
4	Server Version Disclosure	LOW	OPEN

Recommendation

Based on the results of this assessment, KLEAP Technologies Pvt. Ltd. has the following high-level key recommendations.

Key Recommendation (Network)

Key Issue	During the security assessment of the web application, it was observed that the root cause of the vulnerabilities found in the application was mainly due to improper input validation, inadequate access controls, and missing or poor security configurations.
Recommendation	We recommend implementing proper input sanitization, access controls, and appropriate security configurations.

Scope Summary

In-Scope Assets

The following assets were considered explicitly in-scope for testing:

Assets In-Scope	Hostname / CIDR / IP
Web Application	http://testphp.vulnweb.com/

Out-of-Scope Assets

(If any)

Assets Out-of-Scope	Hostname / CIDR / IP
N/A	N/A

Methodology

The pentest methodology employed by KLEAP Technologies Pvt. Ltd. follows a systematic approach to assess the security posture of client systems.

Our Penetration Testing Methodology is based on following guidelines and standards:

- Penetration Testing Execution Standard (PTES)
- NIST SP 800-115
- Open Source Security Testing Methodology Manual (OSSTMM)
- SANS: Conducting a Penetration Test on an Organization
- OWASP Testing Guide
- OWASP Top 10 Application Security Risks

Engagement Phases

1. Reconnaissance

In this phase, the pentester gathers information about the target systems through passive reconnaissance and OSINT techniques. This includes identifying domain names, IP addresses, employee details, and any publicly available information. The goal is to gain a better understanding of the target's infrastructure, potential vulnerabilities, and attack surface.

2. Scanning and Enumeration

In this phase, the pentester conducts active scanning to identify live hosts, open ports, and services running on the target systems. Tools like Nmap, Nessus, or OpenVAS are used to perform network scans and identify potential entry points. The identified services are then enumerated to gather more information, such as software versions, configurations, and potential vulnerabilities. This phase helps in identifying potential weaknesses and areas of focus for further assessment.

3. Vulnerability Assessment

In this phase, the pentester performs a comprehensive vulnerability assessment using a combination of automated tools and manual techniques. Commercial or open-source vulnerability scanners are utilized to identify common vulnerabilities and misconfigurations. The scan results are manually reviewed to validate and prioritize the identified vulnerabilities based on their severity and potential impact. This phase helps in identifying specific vulnerabilities that can be exploited to gain unauthorized access or compromise the target systems.

4. Exploitation

In this phase, the pentester attempts to exploit the identified vulnerabilities to gain unauthorized access or escalate privileges. Ethical hacking techniques are utilized to simulate real-world attack scenarios while ensuring no harm is caused to the target systems. The pentester may use various tools, scripts, or custom exploits to exploit the identified vulnerabilities. The goal is to demonstrate the potential impact of the vulnerabilities and assess the effectiveness of the target's security controls.

5. Reporting

In this final phase, the pentester compiles all findings, categorizes them based on severity levels, and provides detailed explanations, proof-of-concept demonstrations, and prioritized recommendations for remediation. The report includes a summary of the pentest engagement, an overview of the methodology used, and a comprehensive analysis of the vulnerabilities discovered. It also includes actionable recommendations to mitigate the identified vulnerabilities and improve the overall security posture of the target systems. The report serves as a valuable resource for the client to understand the security risks and take appropriate measures to address them.

Vulnerability Classification & Severity

To categorize vulnerabilities according to a commonly understood vulnerability taxonomy, KLEAP Technologies Pvt. Ltd. uses the industry-standard Common Weakness Enumeration (CWE). CWE is a community-developed taxonomy of common software security weaknesses. It serves as a common language, a measuring stick for software security tools, and as a baseline for weakness identification, mitigation, and prevention efforts.

To rate the severity of vulnerabilities, KLEAP Technologies Pvt. Ltd. uses the industry standard Common Vulnerability Scoring System (CVSS) to calculate severity for each identified security vulnerability. CVSS provides a way to capture the principal characteristics of a vulnerability, and produce a numerical score reflecting its severity, as well as a textual representation of that score.

To help prioritize vulnerabilities and assist vulnerability management processes, KLEAP Technologies Pvt. Ltd. translates the numerical CVSS rating to a qualitative representation (such as low, medium, high and critical):

CVSS Score v3.1	
Severity	**Score**
Critical	9.0 - 10.0
High	7.0 - 8.9
Medium	4.0 - 6.9
Low	0.1 - 3.9
Informational	0.0

Findings Summary

Findings are sorted by their severity and grouped by the asset and CWE classification. Each asset section will contain a summary. Table 1 in the executive summary contains the total number of identified security vulnerabilities per asset per risk indication.

Findings Overview

During the engagement, 5 unique vulnerabilities were found across mainly 3 different vulnerability categories. The most common vulnerability types identified were:

- Injection
- Broken Access Control
- Security Misconfiguration

Exploring the findings further by their actual vulnerability type as defined by CWE, Table 3 shows the number of individual findings and its distribution of severity.

Vulnerabilities	Critical	High	Medium	Low	Info
SQL Injection	0	1	0	0	0
Cross-Site Scripting	0	1	0	0	0
Directory Traversal	0	0	1	0	0
Server Version Disclosure	0	0	0	1	0
	0	2	1	1	0

Table 3: severity distribution across vulnerability types

Findings Overview as per OWASP Standards

Vulnerabilities	Results	Findings
A01:2021-Broken Access Control	Fail	1
A02:2021-Cryptographic Failures	Pass	0
A03:2021-Injection	Fail	2
A04:2021-Insecure Design	Pass	0
A05:2021-Security Misconfiguration	Fail	1
A06:2021-Vulnerable and Outdated Components	Pass	0
A07:2021-Identification and Authentication Failures	Pass	0
A08:2021-Software and Data Integrity Failures	Pass	0
A09:2021-Security Logging and Monitoring Failures	Pass	0
A10:2021-Server-Side Request Forgery	Pass	0

Technical Findings Details

01: SQL Injection

Vulnerability Severity	CWE ID
High	89
OWASP Category	CVSS Score
A03:2021 – Injection	8.6

Vulnerability Description

SQL injection is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database. This can allow an attacker to view data that they are not normally able to retrieve.

Vulnerable URL

http://testphp.vulnweb.com/login.php

Impact

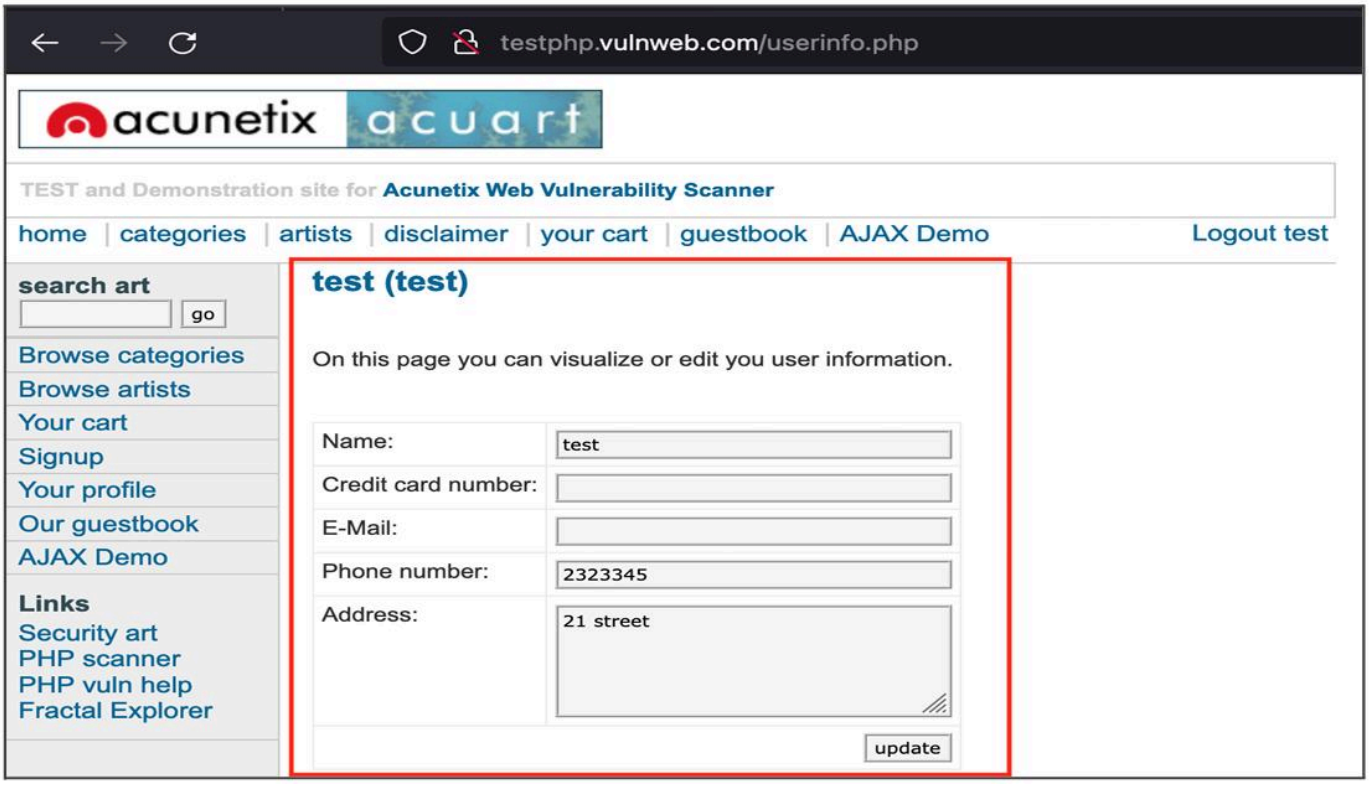
The impact of SQL Injection includes unauthorized data access, data manipulation, data theft and unauthorized application access.

Steps to Reporduce

1. Insert the malicious payload in the username and password input fields.

Steps to Reporduce

2. Successfully got the administrative access to the web application.



Remediation

- Use parameterized queries or prepared statements.
- Validate and sanitize user inputs.
- Escape user inputs if needed.
- Implement a Web Application Firewall (WAF).

References

https://owasp.org/www-community/attacks/SQL_Injection

02: Cross-Site Scripting

Vulnerability Severity	CWE ID
High	80
CVE ID	CVSS Score
A03:2021 – Injection	8.6

Vulnerability Description

Reflected XSS attacks, also referred to as non-persistent attacks, occur when a malicious script is reflected from a web application to the victim's browser. These attacks are triggered when a user interacts with a link that sends a request to a vulnerable website, allowing the execution of malicious scripts.

Vulnerable URL

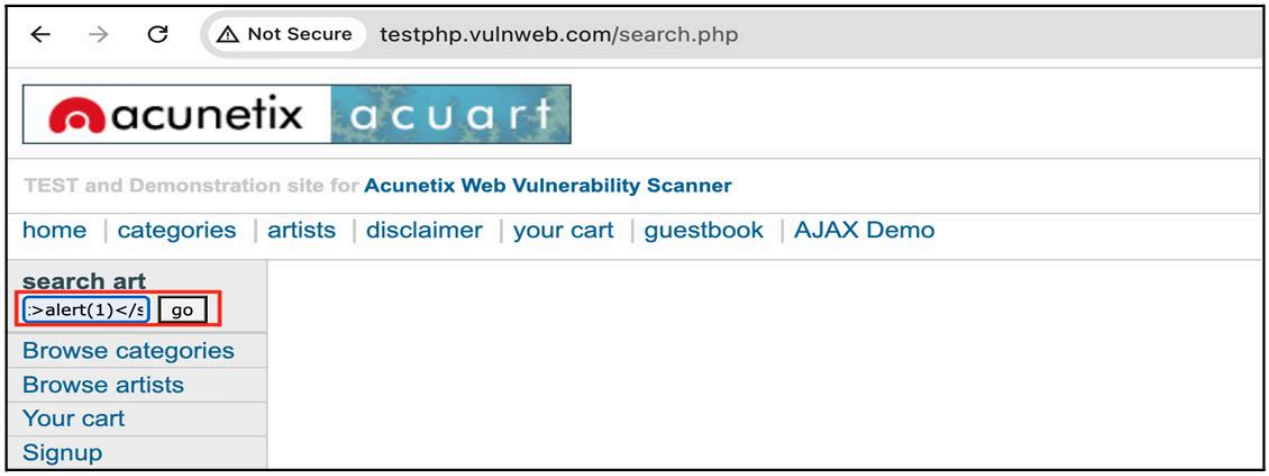
http://testphp.vulnweb.com/search.php

Impact

The impact of an exploited XSS vulnerability in a web application can vary significantly. It can result in various consequences, including the hijacking of a user's session, potential exposure of the user's cookies, and when combined with social engineering tactics, it can lead to the disclosure of sensitive data.

Step to Reproduce

1. Enter the malicious XSS payload i.e <script>alert(1)</script> in the search field.



Step to Reproduce

2. Observe the successful execution of the payload.

Remediation

- Validate and sanitize user inputs to block malicious scripts.
- Encode output data to prevent it from being treated as code when rendered on web pages.

References

https://owasp.org/www-community/attacks/xss/

03: Directory Traversal

Vulnerability Severity	CWE ID
Medium	35
CVE ID	CVSS Score
A01:2021 – Broken Access Control	5.3

Vulnerability Description

Directory traversal (also known as file path traversal) is a web security vulnerability that allows an attacker to read arbitrary files on the server that is running an application.

Vulnerable URL

http://testphp.vulnweb.com/admin

Impact

The impact of directory traversal can be significant as it enables attackers to view the sensitive files, including application code and data, credentials for back-end systems, and sensitive operating system files.

Step to Reproduce (Evidences)

1. Observe the POC which exposes the admin directory.

Remediation

- Input validation and sanitization.
- Strong access controls.
- Whitelisting allowed file paths.

References

https://www.acunetix.com/websitesecurity/directory-traversal/

04: Server Version Disclosure

Vulnerability Severity	CWE ID
Low	200
CVE ID	CVSS Score
A05:2021 – Security Misconfiguration	3.7

Vulnerability Description

The Server header discloses the server version details that handled the request.
The server version details are as follows:
Server: nginx/1.19.0

Vulnerable URL

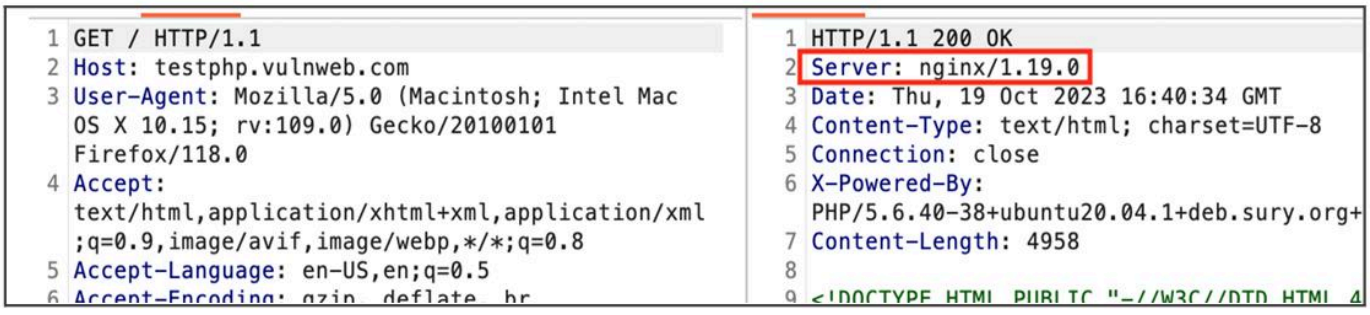
http://testphp.vulnweb.com/

Impact

Server version disclosure exposes vulnerabilities, facilitating targeted attacks and increasing the risk of data breaches and system compromise.

Step to Reproduce (Evidences)

1. Intercept the request and observe the disclosed server version in the response.



```
GET / HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/118.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
```

```
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Thu, 19 Oct 2023 16:40:34 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+
Content-Length: 4958

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4
```

Remediation

- Disable Server Header in the Response

References

https://www.thesmartscanner.com/vulnerability-list/server-version-d isclosure

KLEAP

CYBERSECURITY

https://kleapcybersecurity.com/

info@kleapcybersecurity.com

4111, Briargrove Circle, Raleigh,
North Carolina, 27607, USA