PRAKTIKAN DIMSUM
Security Assessment Findings Report

Business Confidential

Date: March 2nd, 2024
Version 1.0

Confidentiality Statement

This document is the exclusive property of Praktikum ETHACK made by Dimas Andhika D 074. This document contains proprietary and confidential information.

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. TCMS prioritized the assessment to identify the weakest security controls an attacker would exploit. TCMS recommends 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
Praktikan		
Dimsum Andhika	Praktikan Ethack A8	Email: dimasandhk@gmail.com

Assessment Overview

From November 1st, 2024 to November 3rd, 2024, Praktikan ETHACK IT was assessed to find and test for 3 vulnerabilites in a website such as Broken Access Control, SQL Injection, and XSFinding Severity Ratings

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.

Risk Factors

Risk is measured by two factors: Likelihood and Impact:

Likelihood

Likelihood measures the potential of a vulnerability being exploited. Ratings are given based on the difficulty of the attack, the available tools, attacker skill level, and client environment.

Impact

Impact measures the potential vulnerability's effect on operations, including confidentiality, integrity, and availability of client systems and/or data, reputational harm, and financial loss.

Scope

Assessment	Details
Internal Penetration Test	http://40.76.248.146/

Scope Exclusions

Per client request, TCMS did not perform any of the following attacks during testing:

- Denial of Service (DoS)
- Phishing/Social Engineering
- Brute-force
- Takeover server
- Privilege escalation
- Script and/or tools otomation

All other attacks not specified above were permitted by Asisten.

Executive Summary

The following sections provide a high-level overview of vulnerabilities discovered, successful and unsuccessful attempts, and strengths and weaknesses.

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. Internal network penetration testing was permitted for three (3) business days.

Testing Summary

Testing results of the Pentest I've done, application indicate critical vulnerabilities, specifically SQL Injection (SQLi), Cross-Site Scripting (XSS), and Broken Access Control. These issues highlight common security flaws that can lead to unauthorized access, data theft, or manipulation within the system.

The SQL Injection vulnerability was discovered in the login functionality, where user inputs were not properly sanitized, allowing for direct database manipulation. Cross-Site Scripting (XSS) was identified in the search functionality, permitting malicious scripts to be injected and potentially executed in a user's browser. This can lead to session hijacking or data leakage. Additionally, Broken Access Control was noted, allowing unauthorized access to restricted pages, potentially exposing sensitive information.

To address these issues, we recommend implementing parameterized queries to prevent SQL injection, encoding user inputs to mitigate XSS, and enforcing Role-Based Access Control (RBAC) to restrict unauthorized access. Further training in secure coding practices and regular penetration testing are also advised to enhance the overall security posture of Demo Corp's web application.

Key Weaknesses

The following identifies the key weaknesses observed during the assessment:

1. **SQL Injection (SQLi)** vulnerabilities were found in critical input fields, such as the login form.
2. **Cross-Site Scripting (XSS)** vulnerabilities were present in user input fields, allowing malicious script injection.
3. **Broken Access Control** was identified, enabling unauthorized access to restricted pages.
4. **Improper Input Validation** allowed unfiltered input, leading to potential data leaks and vulnerabilities.
5. **Insufficient Security Headers** were detected, increasing the risk of certain attacks like clickjacking. These findings highlight critical areas for security improvement within the Demo Corp web application to mitigate risks and strengthen defenses against potential threats.

Vulnerability Summary & Report Card

The following tables illustrate the vulnerabilities found by impact and recommended remediations:

Internal Penetration Test Findings

3	0	0	0	0
Critical	High	Moderate	Low	Informational

Finding	Severity	Recommendation
Internal Penetration Test		
VULN-001: SQL Injection in Login Form	Critical	Implement parameterized queries and input validation to prevent SQL injection.
VULN-002: Cross-Site Scripting (XSS) in Search and Comment Functionality (Improper Input Validation)	Critical	Encode and sanitize user inputs, and implement a Content Security Policy (CSP).
VULN-003: Broken Access Control	Critical	Implement Role-Based Access Control (RBAC) and validate user permissions on each request.

Technical Findings

Internal Penetration Test Findings

Finding VULN-001: SQL Injection in Login Form (Critical)

Description:	The SQL Injection vulnerability in the login form allows an attacker to inject malicious SQL code into the application's database queries by manipulating user inputs, specifically in the username or password fields. This vulnerability occurs because the application does not properly validate and sanitize user inputs before executing SQL commands. By using specially crafted inputs, an attacker can bypass authentication mechanisms, retrieve sensitive information from the database, or even modify the database contents.
Risk:	**Likelihood:** High – SQL Injection is a common vulnerability, especially in web applications that lack proper input validation, and can be easily exploited by attackers with basic knowledge of SQL. **Impact:** Very High – Successful exploitation of SQL Injection in a login form could allow attackers to gain unauthorized access to user accounts, including administrative accounts. Additionally, attackers could potentially access, modify, or delete sensitive data within the database, leading to data breaches and significant reputational damage.

Evidence

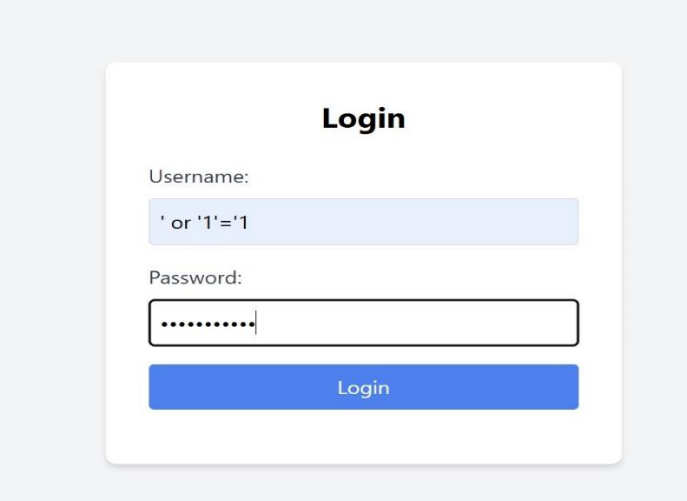

Figure 1: Testing Login Form with basic sqli Payload

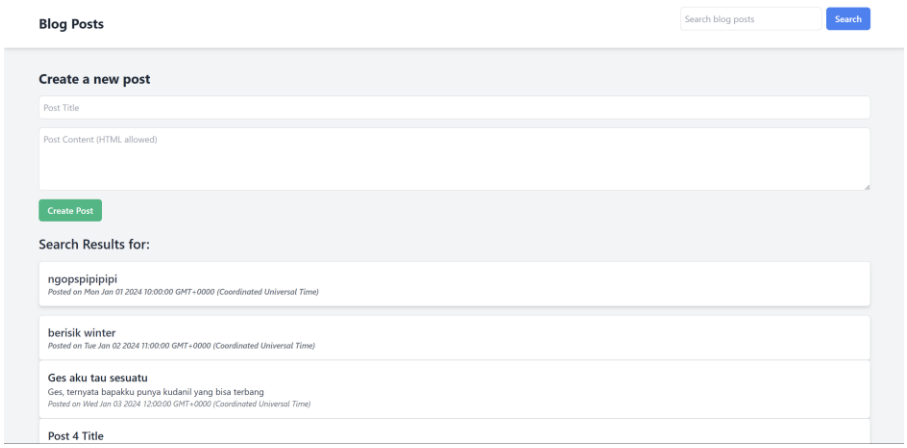

Figure 2: Successfully logged in

Remediation

To remediate the SQL Injection vulnerability in the login form, Demo Corp should use parameterized queries to prevent executable code injection. Enforce input validation, use an ORM framework for secure input handling, and limit database account privileges to reduce impact. A Web Application Firewall (WAF) can help detect and block SQL injection attempts. Regular security testing and developer training on secure coding practices will further protect against this vulnerability.

Finding VULN-002: Cross-Site Scripting (XSS) in Search and Comment Functionality (Improper Input Validation) (Critical)

Description:	The Cross-Site Scripting (XSS) vulnerability in the search and comment functionality allows attackers to inject malicious scripts into the application by exploiting improper input validation. When users input data into these fields, the application fails to sanitize it properly, enabling attackers to embed JavaScript or other executable code. Once injected, this code can execute in the context of other users' browsers, potentially stealing session tokens, redirecting users to malicious sites, or performing actions on behalf of the user.
Risk:	**Likelihood:** High – XSS is a common web application vulnerability and can be easily exploited, especially in fields that allow user input without proper sanitization. **Impact:** High – Successful exploitation of XSS could lead to sensitive information theft, session hijacking, unauthorized actions on behalf of the user, and a severe loss of trust if users fall victim to malicious redirects or phishing attempts.

Evidence

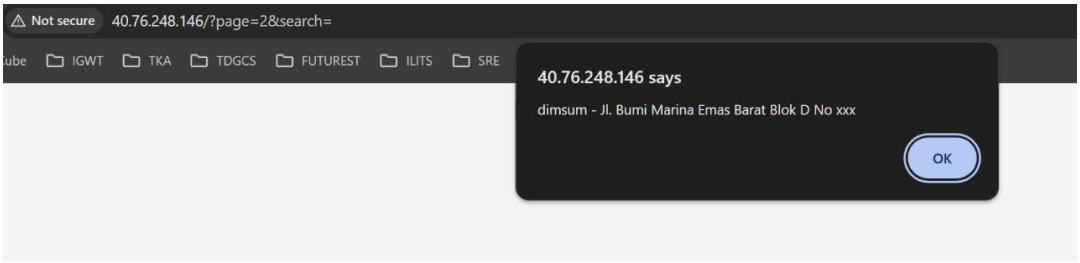

Figure 3: XSS can Append script

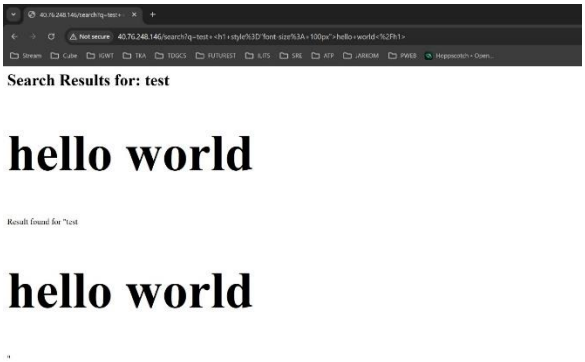

Figure 4: XSS in Search Functionality

Remediation

To mitigate the XSS vulnerability, Demo Corp should ensure all user input is properly sanitized and encoded before it is displayed back to users. Implement input validation to reject or sanitize any characters that could be used in XSS attacks, such as <, >, and &. Additionally, use output encoding to prevent injected scripts from executing in the browser. Employ Content Security Policy (CSP) headers to restrict the sources of executable scripts on the site, adding an extra layer of defense. Regular security testing, such as XSS-focused penetration testing, should be conducted to identify and address any potential XSS issues in the future.

Finding IPT-003: Broken Access Control (Critical)

Description:	The Broken Access Control vulnerability allows unauthorized users to access resources or perform actions that should be restricted. This issue arises when the application fails to enforce proper access controls, enabling attackers to bypass restrictions by directly accessing URLs, manipulating requests, or exploiting insufficiently protected endpoints. As a result, attackers can potentially access sensitive data, modify user accounts, or perform administrative actions without authorization.
Risk:	**Likelihood:** High – Broken Access Control is a common and highly exploitable vulnerability, especially when access controls are not consistently enforced across all parts of the application. **Impact:** Very High – Exploiting this vulnerability could allow attackers to gain unauthorized access to sensitive information, modify data, or perform critical actions, which can lead to data breaches, loss of user trust, and regulatory non-compliance.

Evidence

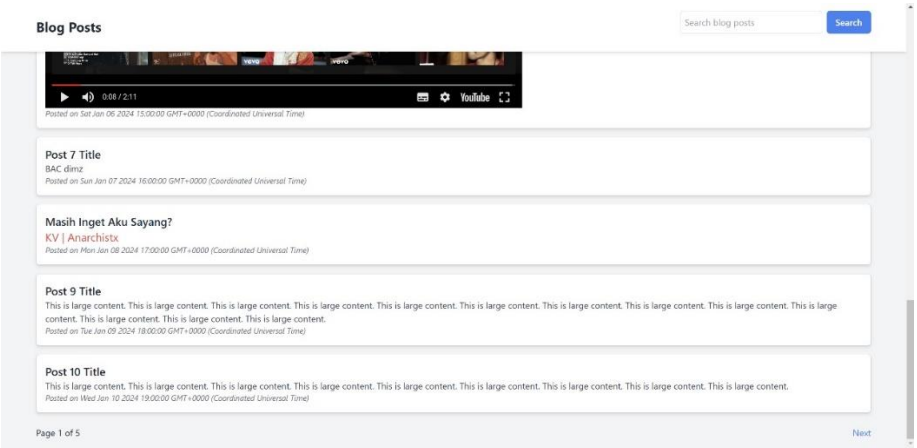

Figure 5: Page 1 where there are no edit button but can be still edited shown in post 7

Remediation

To mitigate the Broken Access Control vulnerability, Demo Corp should implement robust access control mechanisms throughout the application. Ensure that all resources are properly restricted based on user roles and permissions, and avoid client-side access control checks, as they can be bypassed. Use server-side validation to enforce permissions consistently, and limit direct object references or predictable URLs that expose sensitive resources. Additionally, regularly conduct access control testing to verify that users are only able to access resources appropriate to their roles, and review access policies to align with the principle of least privilege.

Last Page