Jay's Bank
Application Penetration Testing Report

Business Confidential

Date: June 1st, 2024
Project: DC-001
Version 1.0

Table of Contents

Confidentiality Statement

This document is the exclusive property of FortifyTech and CyberShield. This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or in part, in any form, requires consent of both FortifyTech and CyberShield.

FortifyTech 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. CyberShield prioritized the assessment to identify the weakest security controls an attacker would exploit. CyberShield 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
FortifyTech		
John Smith	Global Information Security Manager	Email: jsmith@democorp.com
CyberShield		
Marcelinus A	Lead Penetration Tester	Email: marcelalvin11@gmail.com

Assessment Overview

From May 28th, 2024 to June 1st, 2024, FortifyTech engaged Jay's Bank Application to evaluate the security posture of its infrastructure compared to current industry best practices that included an internal network penetration test.

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.
- Exploit - Perform attack based on vulnerabilities found at discovery process.
- Reporting – Document all found vulnerabilities and exploits, failed attempts, and company strengths and weaknesses.

Assessment Components

Internal Penetration Test

An internal penetration test emulates the role of an attacker from inside the network. An engineer will scan the network to identify potential host vulnerabilities and perform common and advanced internal network attacks, such as: finding website's ports and endpoints, finding services that being used inside the website, OS discovery, et cetera. Then exploit weakness that already been found such as SQL Injection (SQLi) is an attack where attackers inject malicious SQL code into input fields or URLs. This allows them to access, modify, or delete database data. Common techniques include Union-based, Error-based, and Blind SQLi. Prevention involves using parameterized queries, input validation, and least privilege principles for database accounts. And Cross-Site Scripting (XSS) is an attack where malicious scripts are injected into trusted websites. These scripts run in the user's browser and can steal data or hijack sessions. Types include Stored, Reflected, and DOM-based XSS. Prevention includes using Content Security Policy (CSP), output encoding, and input validation.

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	• 167.172.75.216 • All application functions • User account mechanism and authentication • Web interface and API • Database interaction and data handling processes

Scope Exclusions

Jay's Bank forbid some attacks during testing, here is the detail:

- It is not allowed to carry out attacks that can damage the data or application infrastructure.
- It is not allowed to exploit vulnerabilities that can provide access to the server (e.g., RCE, privilege escalation).
- Avoid DoS/DDoS attacks that can disrupt the availability of application services.

Client Allowances

Jay's Bank did not provide CyberShield any forms of allowances.

Executive Summary

CyberShield evaluated Jay's Bank's internal security posture through penetration testing from May 28nd, 2024 to June 1st, 2024. 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 five (5) business days.

Testing Summary

First pentester use Gobuster to search for endpoint that used by the application. The result is there are many endpoints such as, /register, /login, /dashboard, /profile. By testing the application, pentester found major vulnerability that can be exploited, such as sql injection and xss attack.

Vulnerability Summary & Report Card

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

Internal Penetration Test Findings

0	2	0	0	0
Critical	High	Moderate	Low	Informational

Finding	Severity	Recommendation
Vulnerable to XSS (Cross-Site Scripting)	High	Implement proper input validation and output encoding to prevent the injection of malicious scripts. Utilize Content Security Policy (CSP) headers to restrict the sources of executable scripts on web pages. Regularly sanitize user input and educate developers on secure coding practices to prevent XSS vulnerabilities in the future.
Vulnerable to SQLi (SQL Injection)	High	Apply vendor patching promptly to address any known vulnerabilities that could be exploited for SQL Injection attacks. Additionally, avoid using Group Policy Preferences (GPP) cpasswords as they are susceptible to exposure. Implement strict input validation and parameterized queries to prevent SQL Injection attacks. Regularly review and update

		database security configurations to ensure robust protection against SQL Injection threats.

External Penetration Test Findings

Vulnerable to XSS (Cross-Site Scripting)

• **Description:**	Cross-Site Scripting (XSS) is another type of attack where malicious scripts are injected into web pages viewed by users. These scripts run within the user's browser, allowing attackers to steal session cookies, redirect users to malicious sites, or deface the website. While CSRF attacks exploit user trust in a website, XSS attacks exploit the trust users have for the content served by that website.
• **Impact:**	High
• **System:**	167.172.75.216/dashboard
• **References:**	• OWASP_2021_A05

Vulnerable to SQLi (SQL Injection)

• **Description:**	SQL Injection (SQLi) is a type of attack where malicious SQL code is injected into input fields or URLs of a web application. This allows attackers to manipulate the backend database, potentially accessing, modifying, or deleting sensitive data. Unlike Cross-Site Request Forgery (CSRF) attacks, which exploit the trust a website has for a user, SQL Injection attacks directly target the vulnerability in the application's database handling mechanism.
• **Impact:**	HIgh
• **System:**	167.172.75.216/change_password
• **References:**	• CWE-89

Exploitation Proof of Concept

Vulnerabilities to SQLi (SQL Injection)

- First pentester create new user with its own password like below.

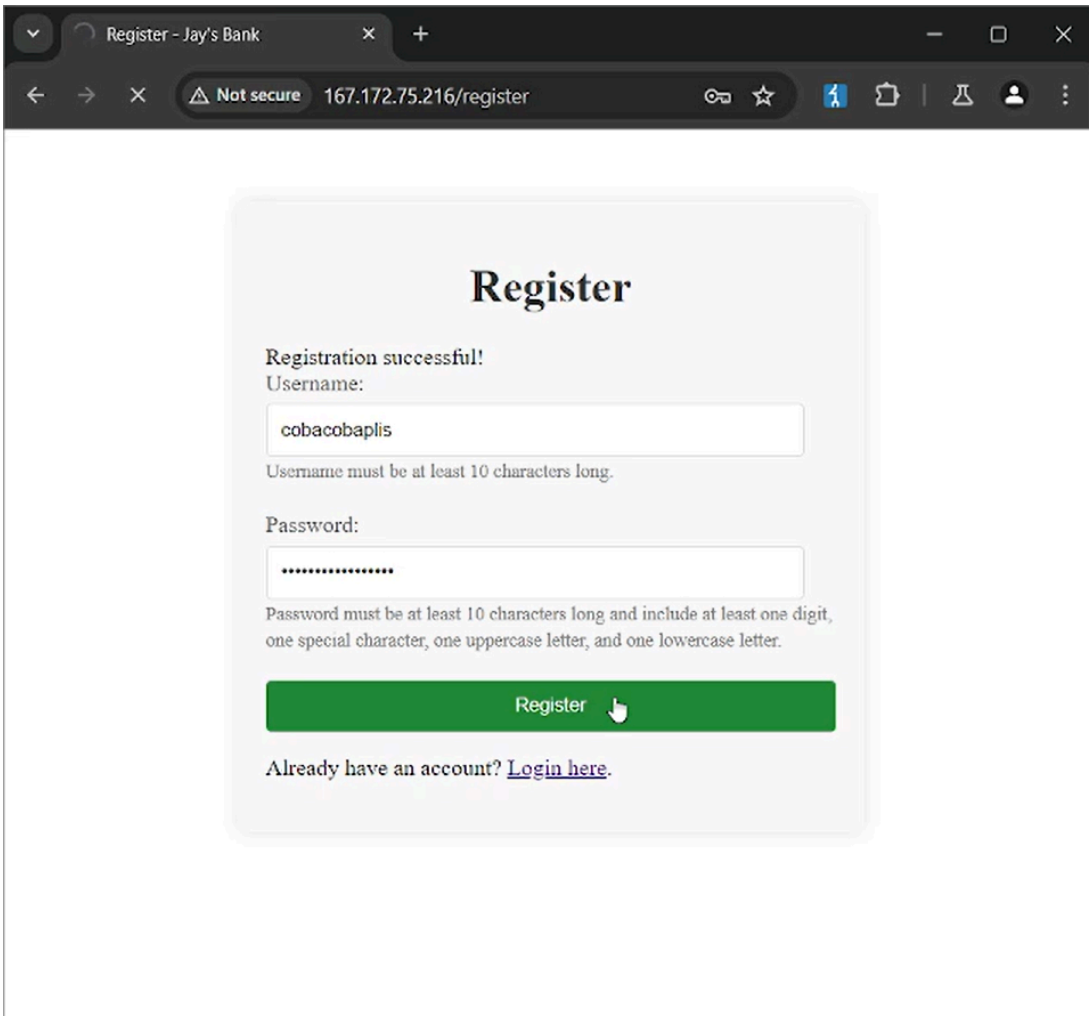

Figure 1: Make a new user from /register endpoint

- Next do login process with newly created username and password.

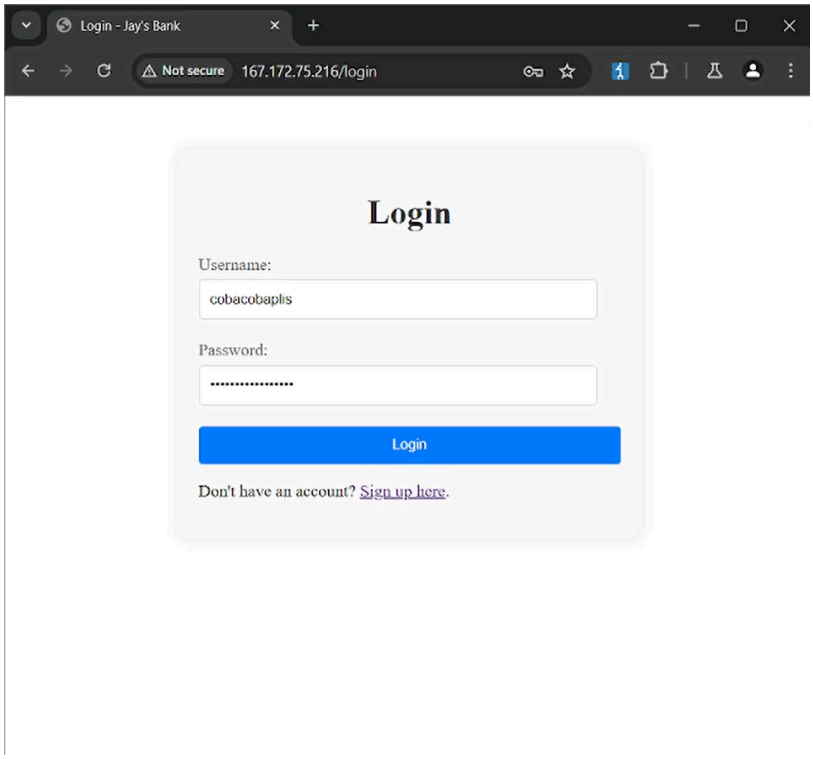

Figure 2: Make login attempt using username and password that just being created

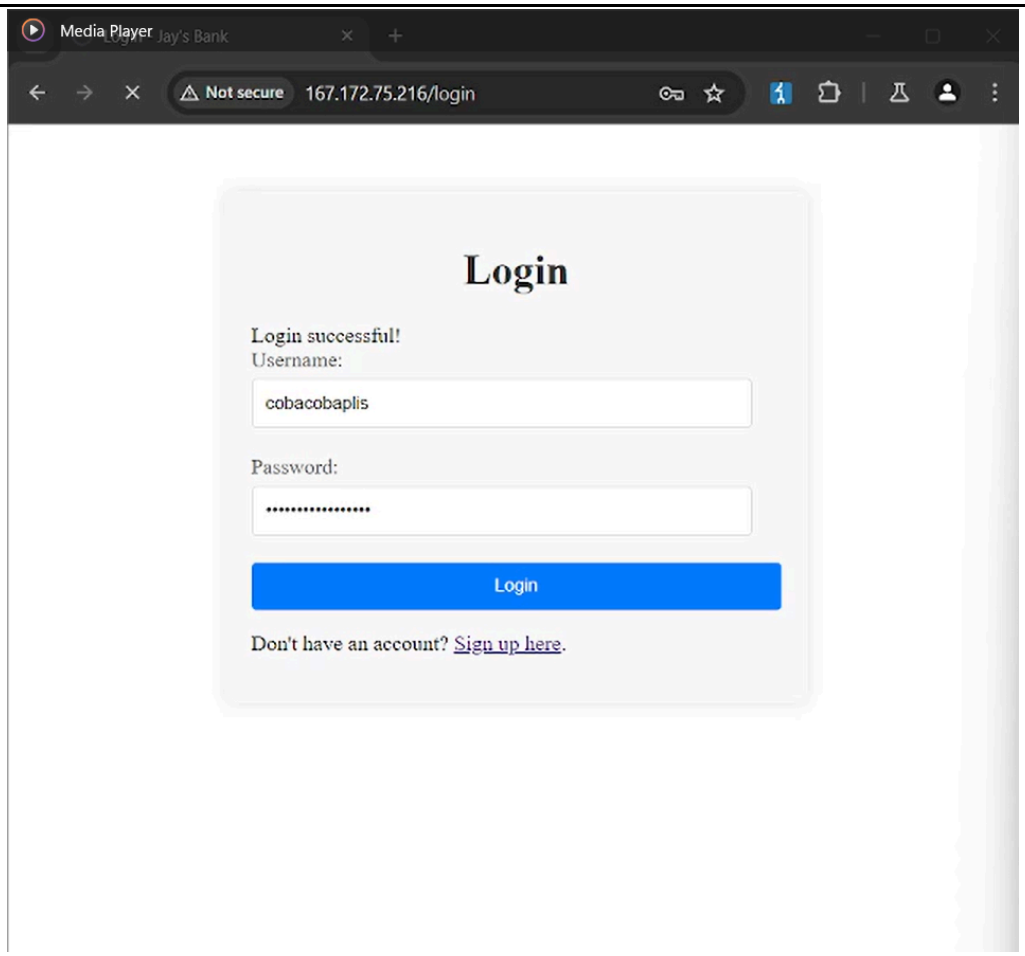

Figure 3: Login successful proof

- Next go to the endpoint /profile to fill necessary information to gain full access to the application's features.

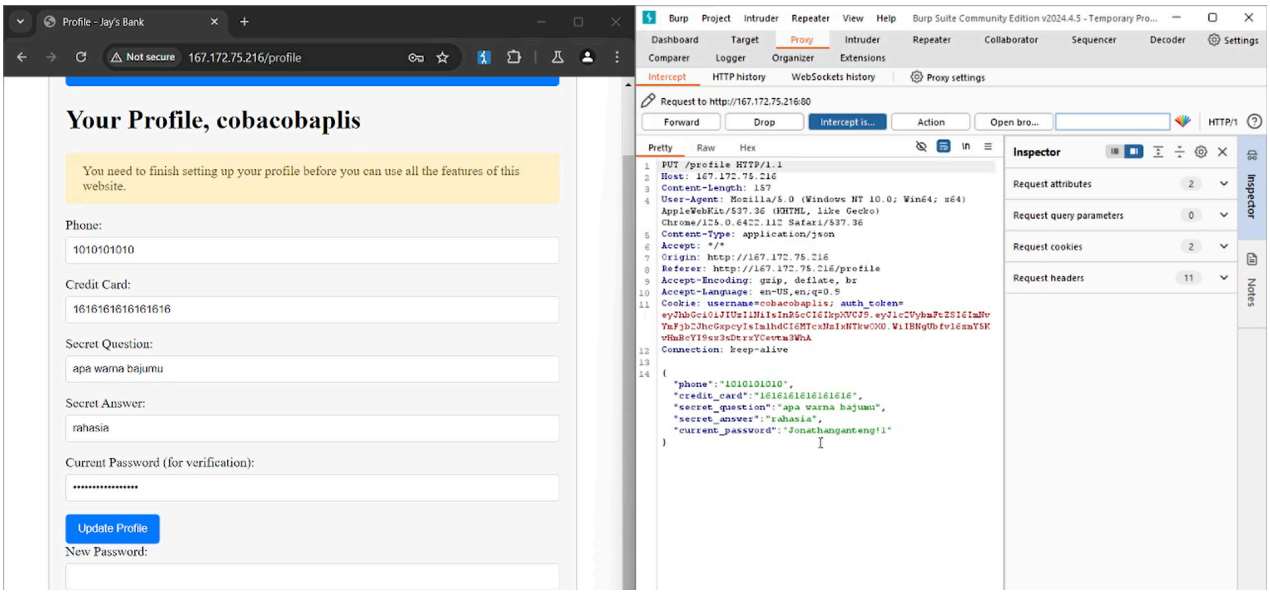

Figure 4: Filling profile column and its payload json form accessed from burpsuite

- Now to gain admin privilege, pentester will make changes to the admin password by using the change password feature from the application. Fill the new password column whatever you like and fill the answer to the secret question set before in profile. After that do intercept with burp suite. Change the username section from the payload that is being sent in to admin.

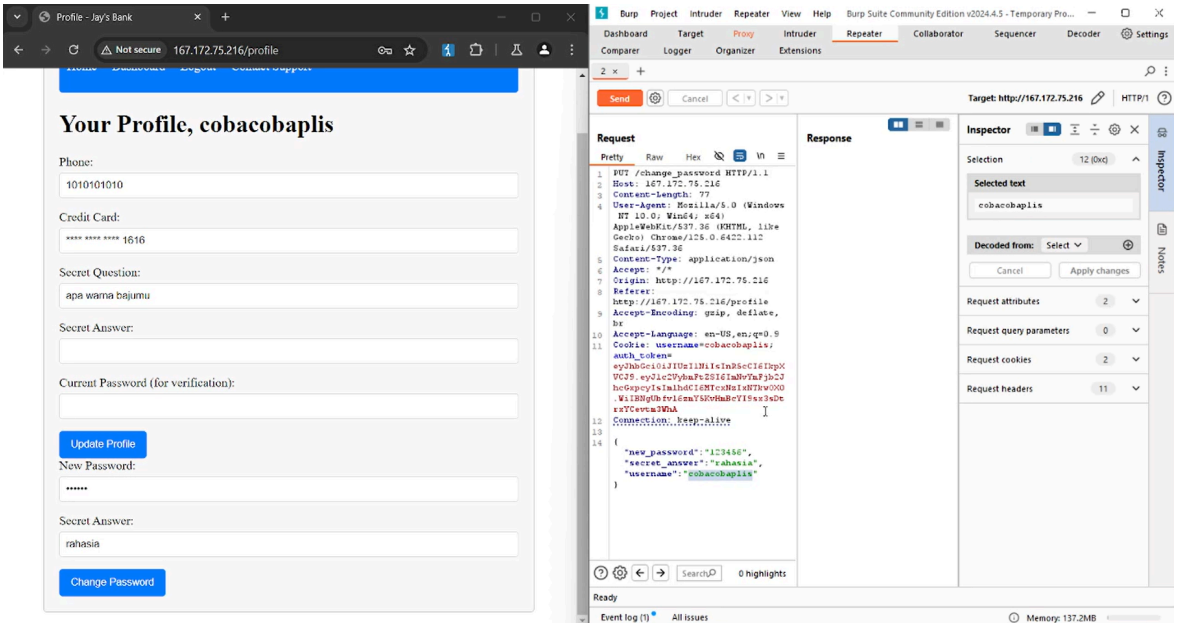

Figure 4: Before changing username to admin using repeater in burp suite

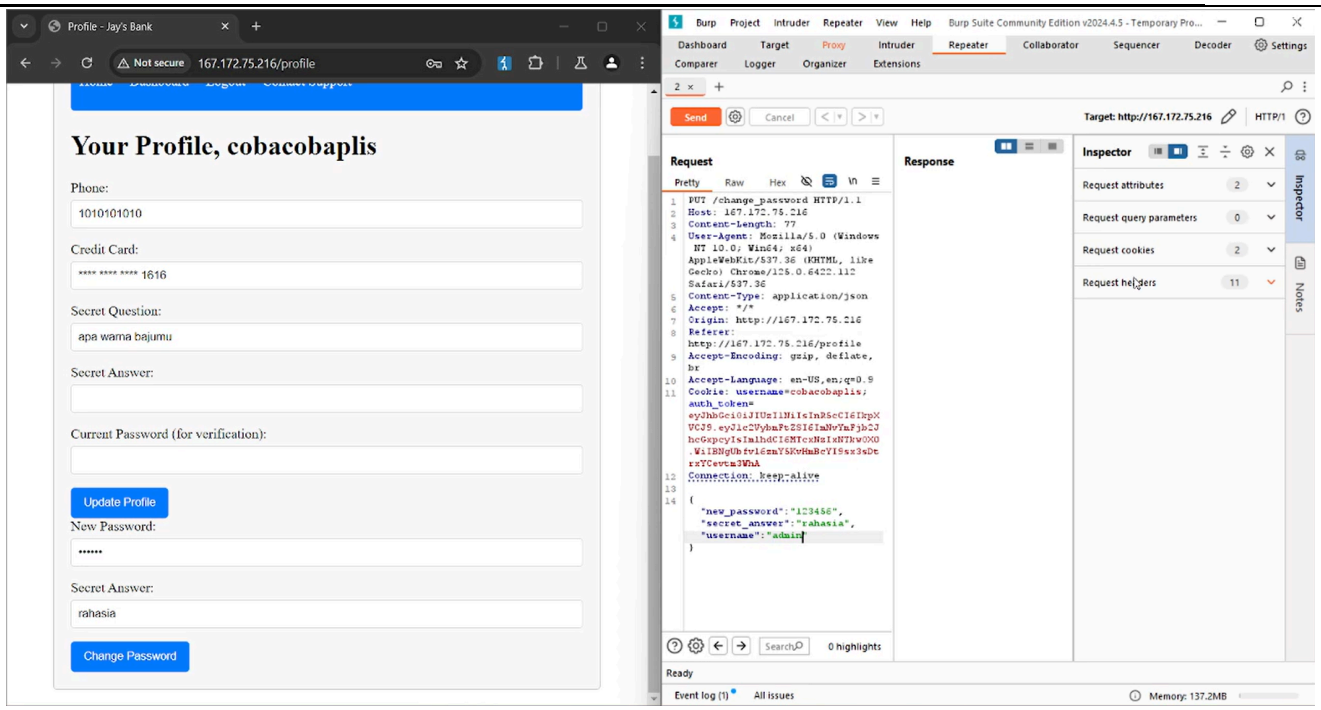

Figure 5: After changing username to admin using repeater in burp suite

- *Now try login to admin with updated password*

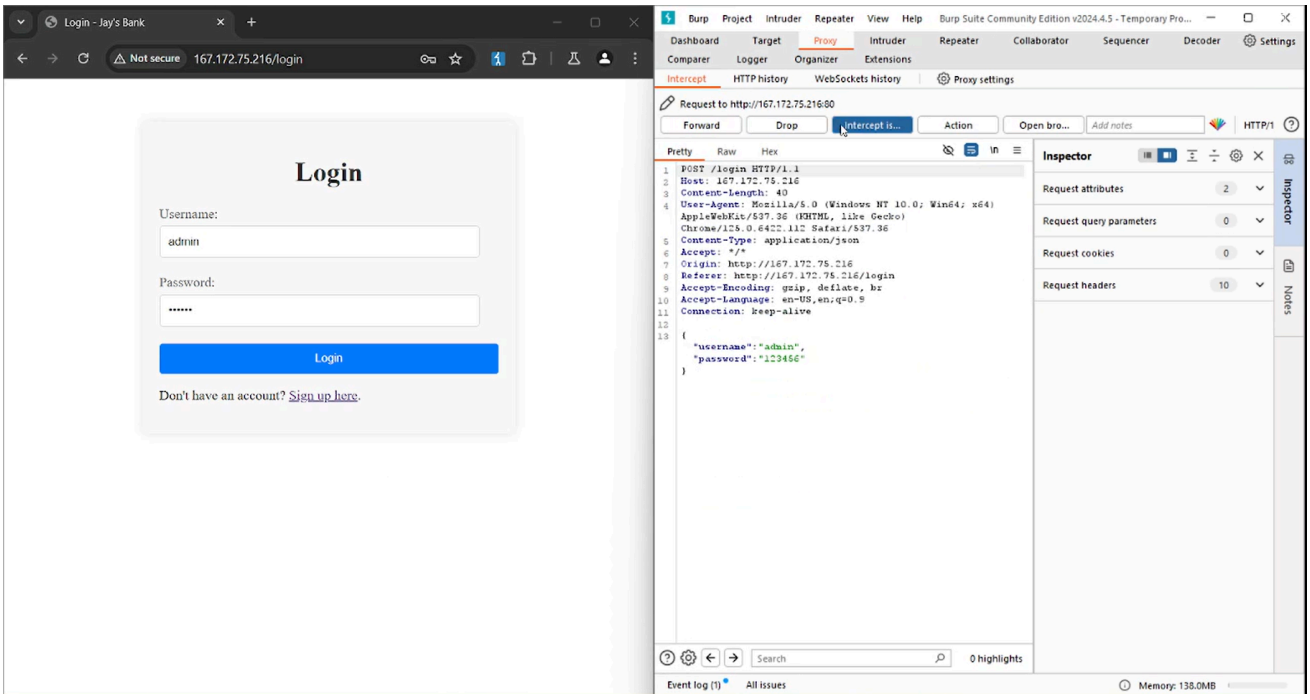

Figure 6: Login to admin using updated password

- *Here's the result. I can log into the admin dashboard and access its data in the profile page.*

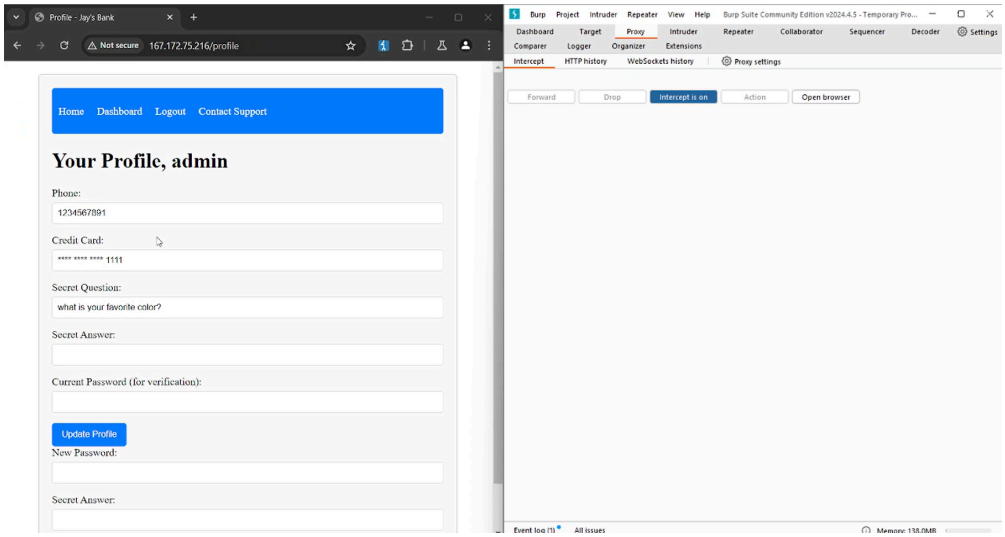

Figure 7: Admin dashboard

Vulnerabilities to XSS (Cross-Site Scripting)

- Create new username that consist of html tag and javascript alert function.

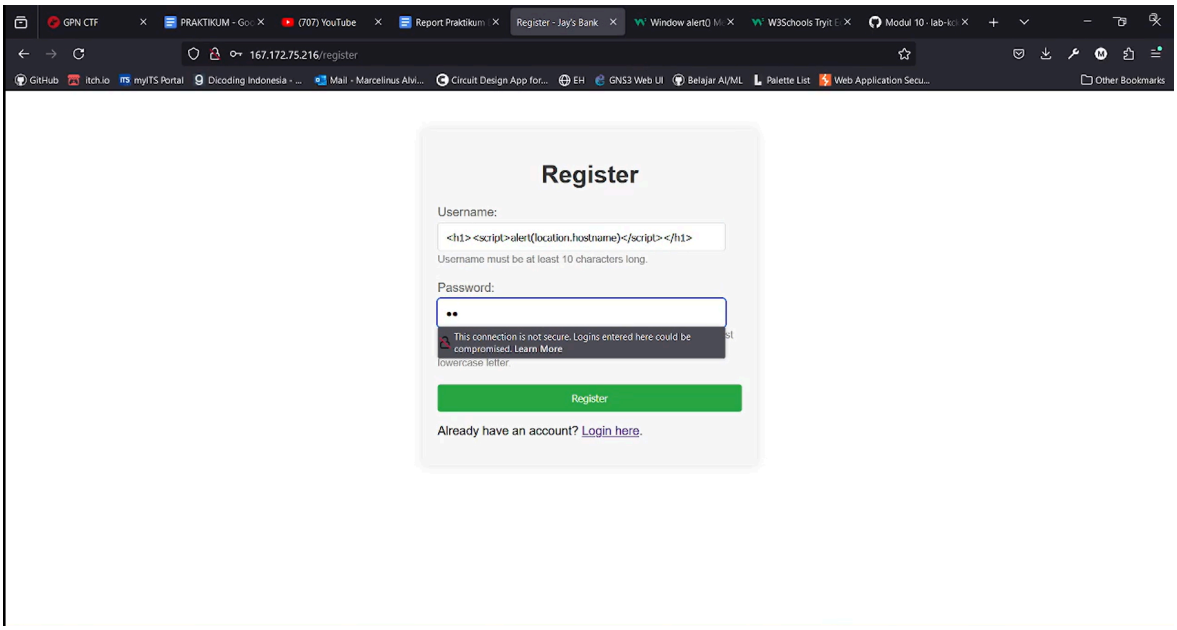

Figure 8: Creating new user for xss

- Login into newly created account at /login

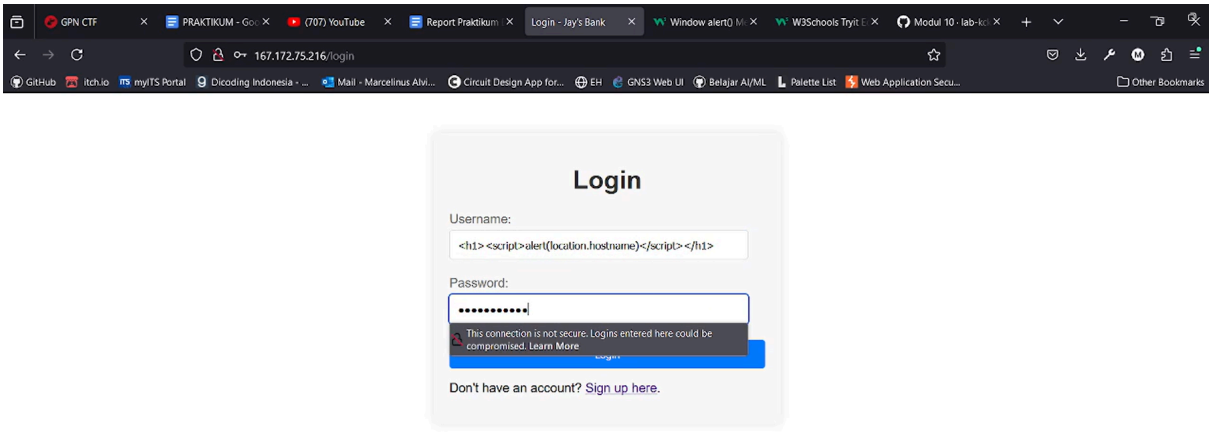

Figure 9: Login to user dashboard

- Fill the profile section with information needed

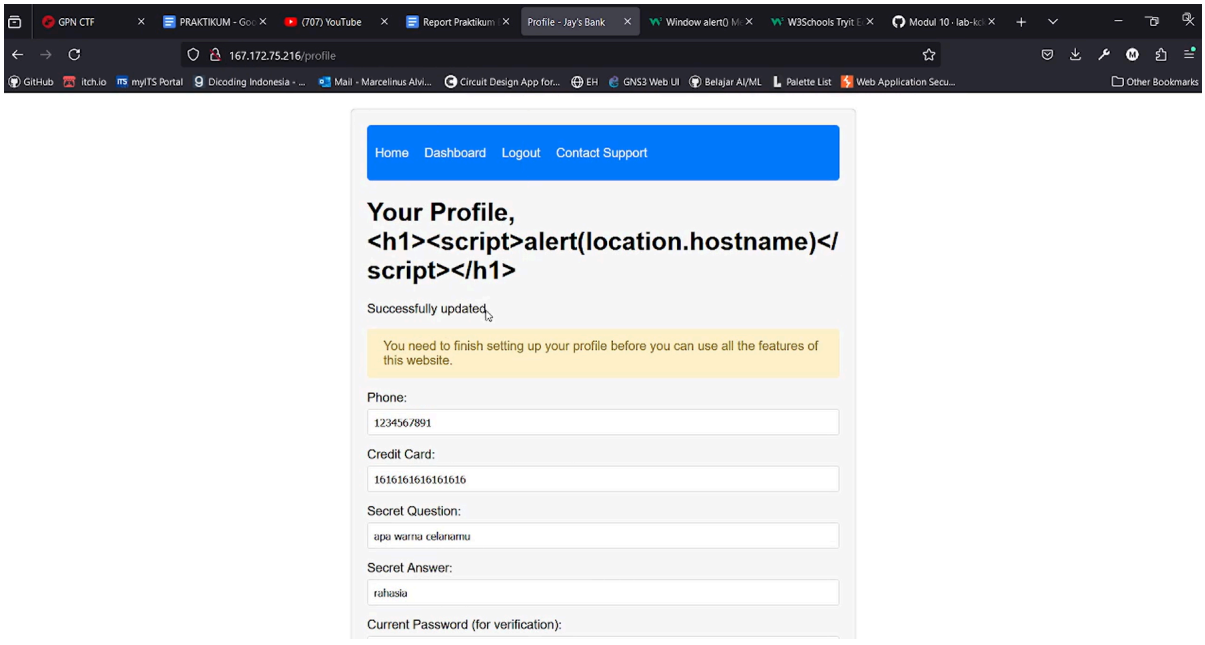

Figure 10: Fill necessary information to access application's features

- Go back into /dashboard endpoint, there the script will be executed and will show alert that contain host location

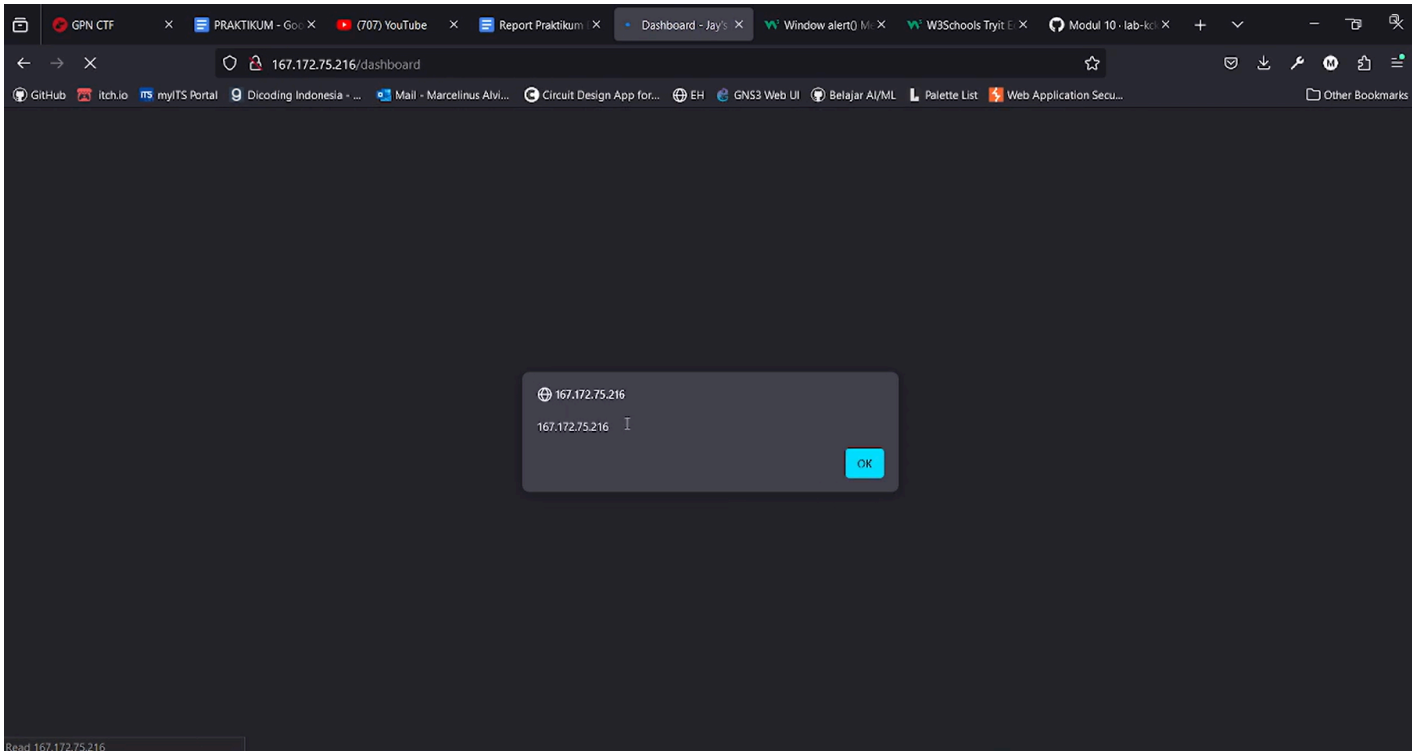

Figure 11: Script are executed

Additional Proof

Pentester includes step by step process video within this report. It included xss attack and sql injection.

Last Page