Security Assessment Findings Report

Jay's Bank

Date: June 1th, 2024

Table of Contents

Confidentiality Statement

This document is the exclusive property of Jay's Bank and SafeGuard. This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or in part, in any form, requires consent of both Jay's Bank and SafeGuard.

Jay's Bank 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. SafeGuard prioritized the assessment to identify the weakest security controls an attacker would exploit. SafeGuard recommends conducting similar assessments on an annual basis by external or third-party assessors to ensure the continued success of the controls.

Contact Information

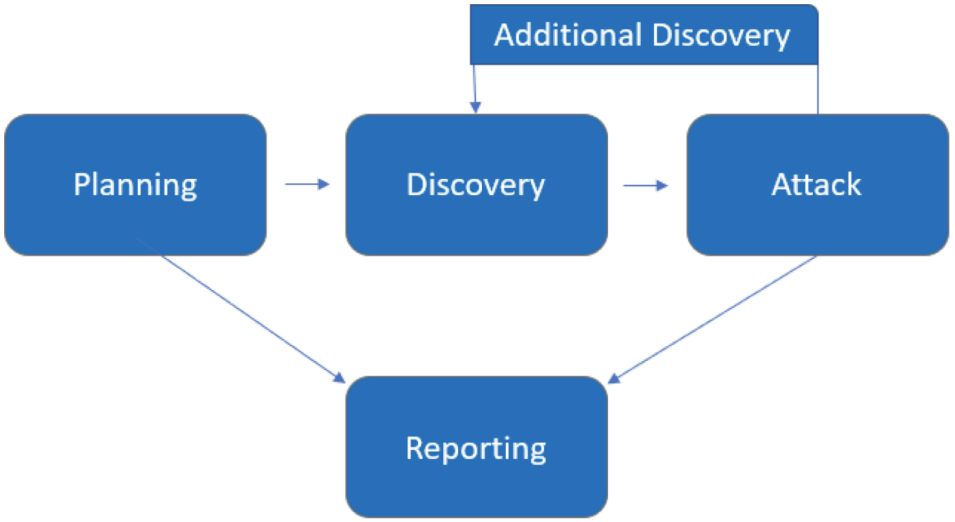
Name	Title	Contact Information
Jay's Bank		
Jay	Security Manager	Email: jay@jaybank.com
SafeGuard		
Mutiara Nurhaliza	Security Expert	Email: mutiaranurhaliza24@gmail.com

Assessment Overview

From May 28nd, 2024 to June 1th, 2024, Jay's Bank engaged SafeGuard Solutions 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.
- 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

Internal Penetration Test

An external 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 are being used inside the website, OS discovery, et cetera.

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

Scope Exclusions

1. Destructive Testing:
    - No attacks that damage, delete, or alter data.
    - No attacks that harm network infrastructure or hardware.
2. Server Exploitation:
    - No Remote Code Execution (RCE).
    - No Privilege Escalation attempts.
3. Denial of Service (DoS/DDoS):
    - No DoS attacks.
    - No DDoS attacks.

Client Allowances

1. Non-Destructive Methods:
    - Use non-destructive testing methods that do not harm systems or data.
    - Techniques must identify vulnerabilities without negative impact on operations or data integrity.
2. Verification and Validation:
    - All found vulnerabilities must be carefully verified and validated before reporting.
    - Confirm that vulnerabilities are real and exploitable without violating set restrictions.
3. Detailed Documentation:
    - Maintain detailed records of all testing steps.
    - Include techniques used, findings, and steps for verification and validation.
4. Testing Within Approved Scope:
    - Conduct tests on web, mobile applications, and network infrastructure within the approved scope.
    - Follow Jay's Bank security policies and procedures during testing.

Executive Summary

SafeGuard Solutions evaluated Jay's Bank Internal security posture through penetration testing from May 28nd, 2024 to June 1th, 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 Destructive Testing, Server Exploitation, and Denial of Service (DoS/DDoS)

Time limitations were in place for testing. External network penetration testing was permitted for five (5) business days.

Testing Summary

One notable vulnerability was identified in the form of a **Broken Access Control** flaw CVE-2018-16476. This flaw allowed unauthorized users to manipulate other users' passwords simply by substituting usernames. Through controlled testing using two accounts, SafeGuard successfully demonstrated the exploitability of this vulnerability. By leveraging tools like Burp Suite, the team verified the ease with which improper access controls could be bypassed, highlighting the urgent need for strengthened access control measures.

Furthermore, the assessment revealed an alarming **Cross-Site Scripting (XSS)** vulnerability OWASP_2021_A05 during the registration process. Injecting a malicious script disguised as a username resulted in a pop-up appearing upon subsequent login attempts. This vulnerability, validated through manual testing, underscores the susceptibility of the system to malicious script injection, posing a serious risk to the integrity and security of user data.

These findings emphasize the critical importance of promptly addressing these vulnerabilities to Jay's Bank's security posture. By implementing robust access control mechanisms and enhancing input validation procedures, Jay's Bank can effectively mitigate the risk of unauthorized access and potential data breaches.

Tester Notes and Recommendations

The findings from the penetration testing conducted on Jay's Bank's network reveal characteristics typical of an organization undergoing its initial security assessment. Among the discovered vulnerabilities, several relate to default configurations within the network infrastructure, such as Broken Access Control and XSS Injection.

A prominent weakness identified is the presence of a Broken Access Control flaw, as evidenced by the ability to modify other users' passwords by substituting usernames. This flaw poses a significant security risk and requires immediate attention. Broken Access Control vulnerabilities often result from misconfigured or inadequate access control mechanisms, allowing unauthorized users to perform actions beyond their intended permissions.

Additionally, the assessment uncovered vulnerabilities stemming from Cross-Site Scripting (XSS), presenting a potential avenue for malicious script injection and unauthorized access. XSS Injection vulnerabilities arise when user input is not properly validated, allowing attackers to inject malicious scripts into web applications. Such vulnerabilities can lead to the theft of sensitive information, session hijacking, or the execution of arbitrary code within the context of the affected application.

Moving forward, Jay's Bank should prioritize implementing the recommended measures to enhance its security posture and mitigate potential risks effectively. By addressing these vulnerabilities promptly and proactively, Jay's Bank can strengthen its security defenses and safeguard against potential cyber thread

Key Strengths and Weaknesses

In assessing Jay's Bank's security posture, several key strengths and weaknesses were identified, providing insights into areas of resilience and vulnerability within the network infrastructure.

Key Weaknesses:

1. Broken Access Control: The discovery of a Broken Access Control flaw, allowing password modification by substituting usernames, poses a significant security risk. This vulnerability underscores the importance of implementing granular access controls to prevent unauthorized modifications to user accounts.
2. XSS Injection Vulnerability: The presence of an XSS Injection vulnerability introduces the risk of malicious script injection and unauthorized access. Proper input validation mechanisms should be implemented to mitigate this risk effectively.

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
CVE-2018-16476:Broken Access Control	High	Manage the setting, management, and handling of privileges, particularly by explicitly managing trust zones within the software. The strategy of Separation of Privilege should be employed to compartmentalize the system, creating "safe" areas where trust boundaries are clearly defined. Sensitive data should never be allowed to leave these safe areas, and caution should always be exercised when interfacing with compartments outside of the safe area. Architects and designers must ensure that appropriate compartmentalization is integrated into the system design, reinforcing privilege separation functionality. This entails relying on the principle of least privilege to determine when privileges should be used and when they should be dropped, thus enhancing the security posture of the software system.

OWASP A05:2021 – Security Misconfiguration	High	establishing a repeatable hardening process for swift and consistent deployment of secure environments across development, QA, and production stages. This process should be automated to streamline setup efforts and ensure identical configurations. Additionally, utilizing a minimal platform devoid of unnecessary features, components, documentation, and samples is essential. Regularly review and update configurations in line with security advisories, updates, and patches as part of a robust patch management process, including cloud storage permissions like S3 bucket permissions. Implementing a segmented application architecture ensures effective and secure separation between components or tenants, leveraging techniques such as segmentation, containerization, or cloud security groups (ACLs). Send security directives to clients, such as Security Headers, to reinforce security measures and best practices. Finally, employ an automated process to verify the efficacy of configurations and settings across all environments, ensuring consistent adherence to security standards and minimizing potential vulnerabilities.

Technical Findings

External Penetration Test Findings

Vulnerable to BAC (Broken Access Control)

• **Description:**	A Broken Access Control vulnerability allows an attacker to craft user input which can cause Active Job to deserialize it using GlobalId and give them access to information that they should not have.
• **Impact:**	HIgh
• **System:**	167.172.75.216/change_password
• **Tools**	Burpsuite
• **References:**	• CVE-2018-16476

Evidence

- First Account has username ethack1234 and password Ethack1234!

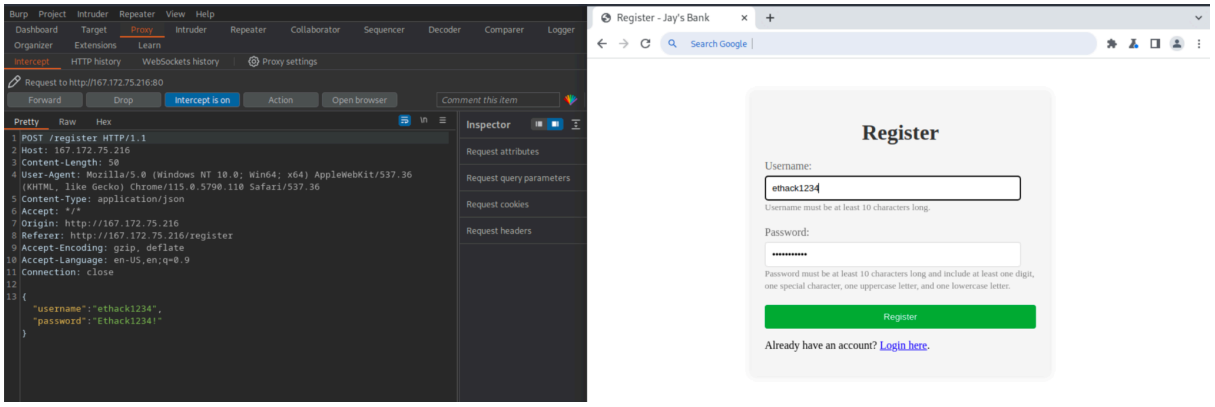


Figure 1: Make a new user from /register endpoint

- Second Account has username ethack5678 and password Ethack5678!

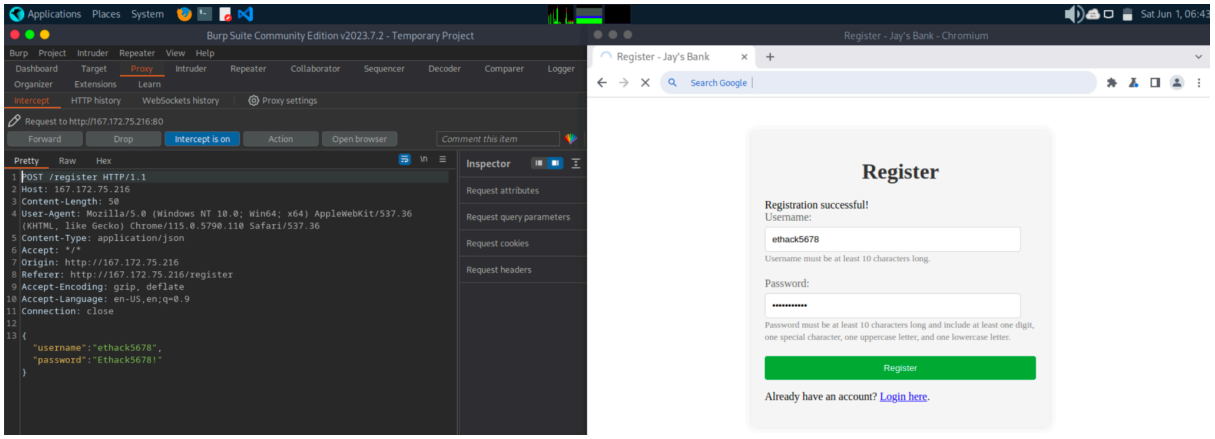


Figure 2: Make a new user from /register endpoint

- Change the first account username to the username of the second username, and change the password to the new one

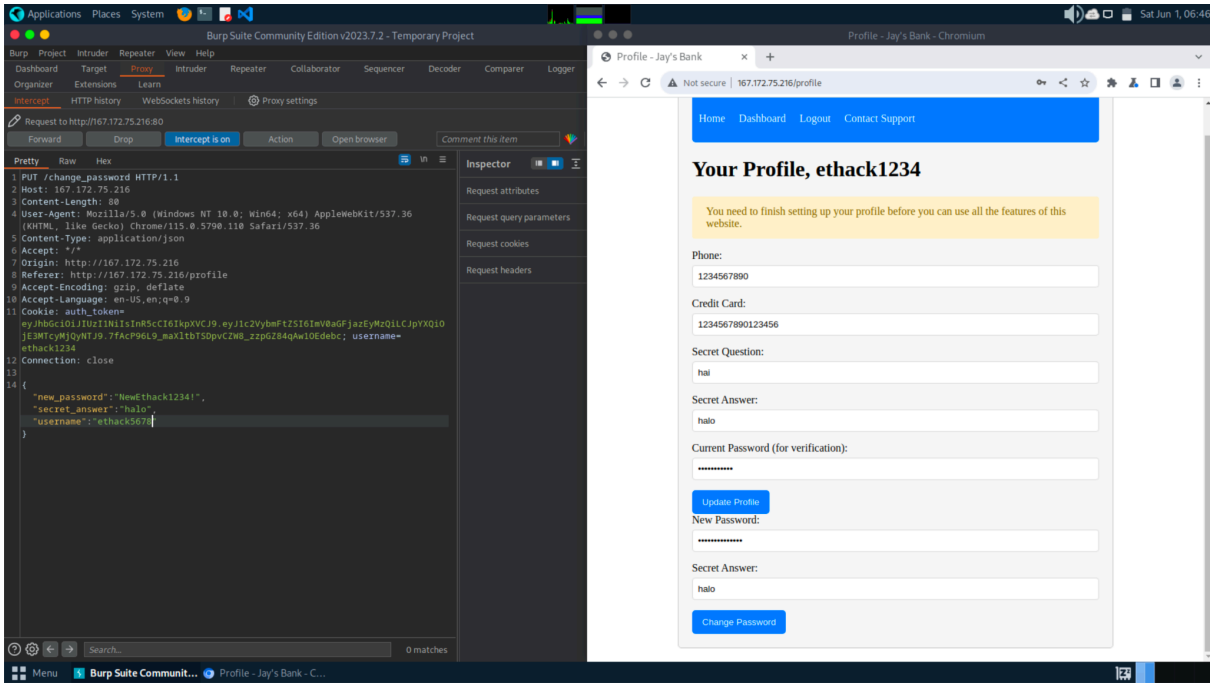


Figure 3: Change username and password

- Second username, can't login with the password that been registered

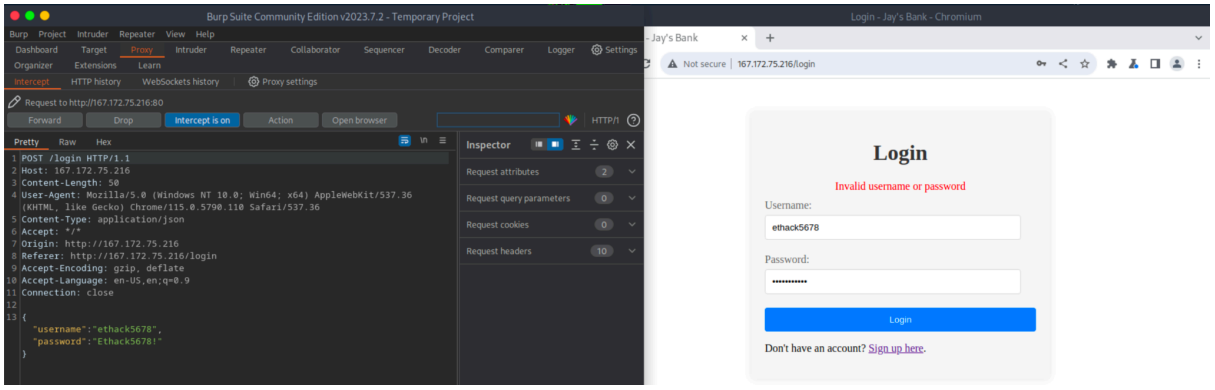


Figure 4: Can't login with the original password

- Second username only can login with the password that has been changed in first username

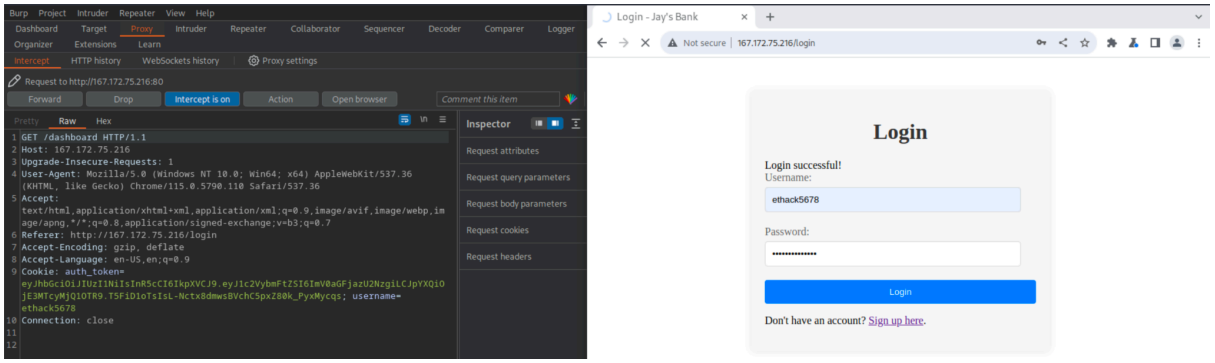


Figure 5: Login Successful with the password that has been changed

Remediation

Access controls within the application, ensuring that user privileges are accurately enforced to prevent unauthorized access. Additionally, thorough validation and sanitization of user input, particularly within the Active Job framework, should be enforced to thwart malicious attempts to manipulate data. Strengthening authentication mechanisms and regularly updating dependencies, including Active Job and GlobalId, are imperative to mitigate known vulnerabilities. Continuous monitoring, auditing of access, developer education on secure coding practices, and timely application of security patches, such as those addressing CVE-2018-16476, further bolster the system's resilience against exploitation and unauthorized access attempts.

Vulnerabilities to XSS (Cross-Site Scripting)

• **Description:**	Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it.
• **Impact:**	High
• **System:**	167.172.75.216/dashboard
• **Tool**	Manual
• **References:**	• OWASP_2021_A05

Evidence

- Create new username that consist of html tag and javascript alert function. In here we use <h1><script>alert(99)</script></h1>

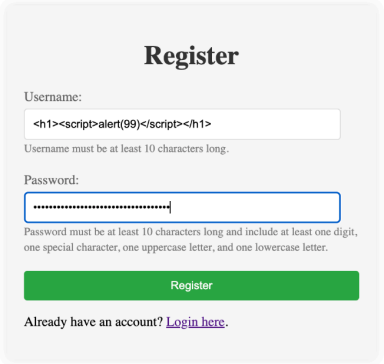
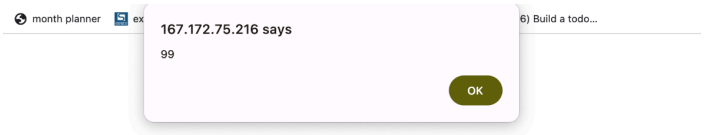


Figure 6: Register with <h1><script>alert(99)</script></h1> as username

- After logging the script will be executed and will show pop up alert that contain host location



Remediation

implement comprehensive measures such as input validation and encoding to sanitize user input, enforce a Content Security Policy (CSP) to restrict unauthorized script execution, and employ output encoding techniques to neutralize potential script injections. Additionally, sanitize user input rigorously, educate developers on secure coding practices, and integrate Web Application Firewalls (WAFs) for real-time monitoring and filtering of incoming traffic. Utilize automated security scanning tools to detect and resolve XSS vulnerabilities early, and ensure prompt application of security updates to mitigate emerging threats. By adopting these remediation strategies collectively, the risk of XSS attacks can be effectively mitigated, bolstering the overall security resilience of the system.