Jay’s Bank Application Security Assessment Findings Report

Business Confidential

Date : June 1th,
2024 Project:
Version 1.0

Table of Contents

Confidentiality Statement

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

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

Contact Information

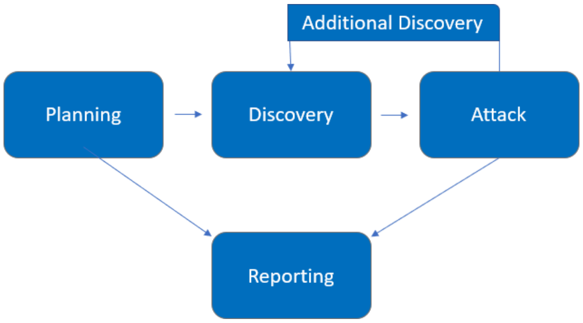
Name	Title	Contact Information
SafeGuard Solutions		
Azzahra Sekar Rahmadina	Penetration Tester	Email: zaraxxxxx@gmail.com

Assessment Overview

From May 28th, 2024 to June 1th, 2024, SafeGuard Solutions engages SafeGuard Solution to evaluate its infrastructure security posture against current industry best practices, including external penetration testing

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

External Penetration Test

An external penetration test emulates the role of an attacker attempting to gain access to an External network without External resources or inside knowledge. A TCMS engineer attempts to gather sensitive information through open-source intelligence (OSINT), including employee information, historical breached passwords, and more that can be leveraged against external systems to gain External network access. The engineer also performs scanning and enumeration to identify potential vulnerabilities in hopes of exploitation.

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
External Penetration Test	- 167.172.75.216 - All application functions - User account and authentication mechanisms - Web interface and API - Database interactions and data handling processes

Scope Exclusions

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

Client Allowances

Jay’s Bank Application doesn't provide any allowances to assist with testing.

Executive Summary

SafeGuard Solution evaluated Jay's Bank Application's External security posture through penetration testing from May 28th, 2024 to June 1th, 2024. By leveraging a series of attacks, SafeGuard Solution identified several moderate-level vulnerabilities. It is advised to create an action plan and patch after addressing the highest priority issues.

Scoping and Time Limitations

Scoping during the engagement did not permit things that violate ethics. Time limitations were in place for testing. Penetration testing was permitted for four (4) days.

Testing Summary

Using Nmap and SQL Injection to gather information about services, applications, devices, and other elements that we will test for security vulnerabilities.

Vulnerability Summary & Report Card

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

External Penetration Test Findings

0	1	4	1	0
Critical	High	Moderate	Low	Informational

Finding	Severity	Recommendation
External Penetration Test		
IPT-001: HTTP service on port 80 (Node.js with Express middleware)	High	Implement HTTPS to encrypt traffic. Ensure all dependencies are up to date and secure. Perform regular security audits and use security-focused middleware.
IPT-002: No SQL Injection Vulnerabilities Detected	Informational	The tests performed did not find any SQL injection vulnerabilities in the URI parameter #1*

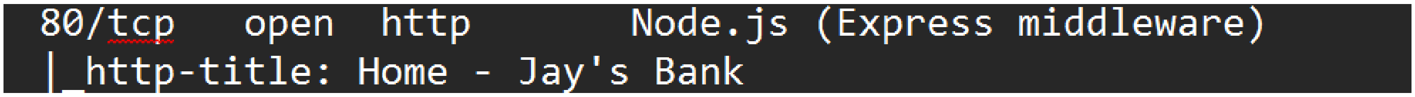
Technical Findings

HTTP service on port 80 (Node.js with Express middleware)

Description	

Risk	Likelihood: High - Web services are frequently targeted for various attacks (e.g., SQL injection, XSS, CSRF) due to their exposure to the internet. Impact: Successful exploitation can lead to data breaches, defacement, unauthorized access, and potentially full server compromise depending on the application and server configuration.
System	http://167.172.75.216
Tools Used	Nmap
References	-

Evidence



```
 80/tcp   open  http       Node.js (Express middleware)
 |_http-title: Home - Jay's Bank
```

Remediation

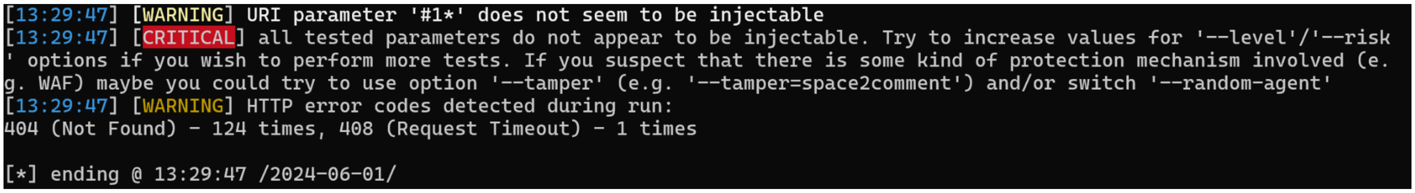
Implement HTTPS to encrypt traffic. Ensure all dependencies are up to date and secure. Perform regular security audits and use security-focused middleware.

Open SSH Version

Description	The tests performed did not find any SQL injection vulnerabilities in the URI parameter #1*
Risk	Likelihood: Given the extensive testing and the negative results for SQL injection vulnerabilities, the likelihood of this specific parameter being vulnerable to SQL injection is low. Impact: If there were an SQL injection vulnerability, the impact could be high, potentially allowing attackers to execute arbitrary SQL

	commands, leading to data breaches, unauthorized data access, and possibly full system compromise.
System	http://167.172.75.216
Tools Used	SQL Injection
References	-

Evidence



```
[13:29:47] [WARNING] URI parameter '#1*' does not seem to be injectable
[13:29:47] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk
' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involved (e.
g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'
[13:29:47] [WARNING] HTTP error codes detected during run:
404 (Not Found) - 124 times, 408 (Request Timeout) - 1 times

[*] ending @ 13:29:47 /2024-06-01/
```

\

Last Page