Jay's Bank
Security Assessment Findings Report

Business Confidential

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

Table of Contents

Table of Contents 2
Confidentiality Statement 4
Disclaimer 4
Contact Information 4
Assessment Overview 5
Assessment Components 5
  Internal Penetration Test 5
Finding Severity Ratings 6
Risk Factors 6
  Likelihood 6
  Impact 6
Scope 7
  Scope Exclusions 7
  Client Allowances 7
Executive Summary 8
  Scoping and Time Limitations 8
  Testing Summary 8
  Tester Notes and Recommendations 9
  Key Strengths and Weaknesses 10
Vulnerability Summary & Report Card 11
  Internal Penetration Test Findings 11
Technical Findings 13
  Internal Penetration Test Findings 13
    Finding IPT-001: Get the database access using SQL Injection (High) 13
    Finding IPT-002: Can change password after getting the database (Broken Access Control) (Moderate) 14

Confidentiality Statement

This document is the exclusive property of Jay's Bank and SafeGuard Solutions. 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 internal or third-party assessors to ensure the continued success of the controls.

Contact Information

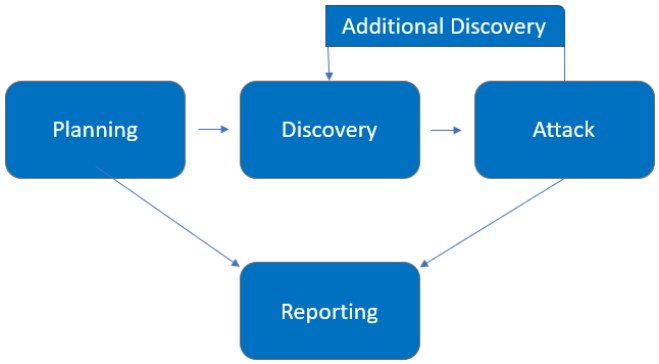
Name	Title	Contact Information
Jay's Bank		
John Smith	Global Information Security Manager	Email: jsmith@jaysbank.com
SafeGuard Solutions		
Gavriel Pramuda Kurniaadi	Lead Penetration Tester	Email: gavriel.k.adi12@gmail.com

Assessment Overview

From May 28th, 2024 to June 1st, 2024, Jay's Bank engaged SafeGuard to evaluate the security posture of its infrastructure compared to current industry best practices that included an internal network penetration test. All testing performed is based on the NIST *SP 800-115 Technical Guide to Information Security Testing and Assessment, OWASP Testing Guide (v4), and customized testing frameworks*.

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 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: LLMNR/NBT-NS poisoning and other man- in-the-middle attacks, token impersonation, kerberoasting, pass-the-hash, golden ticket, and more. The engineer will seek to gain access to hosts through lateral movement, compromise domain user and admin accounts, and exfiltrate sensitive data.

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

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

- Denial of Service (DoS)
- Phishing/Social Engineering

All other attacks not specified above were permitted by Jay's Bank.

Client Allowances

Demo Corp provided TCMS the following allowances:

- Internal access to network via dropbox and port allowances

Executive Summary

SafeGuard evaluated Jay's Bank internal security posture through penetration testing from May 28th, 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 four (4) business days.

Testing Summary

The SafeGuard team has completed evaluating the internal network security of Jay's Bank. From an internal perspective, the SafeGuard team conducted vulnerability scanning against all IPs provided by Jay's Bank to assess the overall patching health of the network. The team performed scanning by conducting reconnaissance, vulnerability testing, and searching for public exploits or CVEs.

From the scanning conducted, the SafeGuard team discovered a database access after doing SQLi using sqlmap (IPT-001). The SafeGuard team also found a Broken Access Control after getting the database (IPT-002).

The remainder of the findings were high, moderate, low, or informational. For further information on findings, please review the Technical Findings section.

Tester Notes and Recommendations

The SafeGuard team has completed an internal network security assessment for Jay's Bank, focusing on evaluating the network's patching health through reconnaissance, vulnerability scanning, and testing for public exploits or CVEs.

During the testing, SafeGuard team has successfully performed SQL injection using sqlmap, and gained unauthorized access to the database. After accessing the database, the team discovered broken access control mechanisms that allowed further unauthorized access and manipulation.

Based on that, we recommend that Jay's Bank should implement input validation and parameterized queries to prevent SQL injection, use ORM (Object-Relational Mapping) frameworks that inherently protect against SQL injection, and conduct thorough code reviews and implement automated testing to identify and mitigate SQLi vulnerabilities to prevent database access via sql injection. We also recommend reviewing and update access control policies to ensure proper implementation and enforcement and conduct a comprehensive audit of access controls across all systems and databases to prevent broken access control.

For long-term actions, we recommend to doing regularly update and patch database management systems, educate developers on secure coding practices and the risks of SQL injection, integrate a Web Application Firewall (WAF) to detect and block SQL injection attempts, implement role-based access control (RBAC) to ensure users have the minimum necessary permissions, regularly review and update access controls to adapt to changes in the organization's structure and technology, and perform regular security training for employees to understand the importance of proper access control to prevent from penetration testing.

Key Strengths and Weaknesses

The following identifies the key strengths identified during the assessment:

1. The database is vulnerable to SQL Injection
2. Account control and authorization is easily breached

The following identifies the key weaknesses identified during the assessment:

1. Use framework for the website application to protect it more safely, especially ORM frameworks
2. Use user input sanitation
3. Use another way for the account authorization

Vulnerability Summary & Report Card

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

Internal Penetration Test Findings

0	1	1	0	0
Critical	High	Moderate	Low	Informational

Finding	Severity	Recommendation
Internal Penetration Test		
IPT-001: Get the database access using SQL Injection	High	Implement input validation and parameterized queries
IPT-002: Can change password after getting the database (Broken Access Control)	Moderate	Using email as an authorization for changing password

Technical Findings

Internal Penetration Test Findings

Finding IPT-001: Get the database access using SQL Injection

Description:	Doing SQL Injection using sqlmap tool. The result is all user account database information.
Risk:	Likelihood: High – SQL Injection is one of the many popular penetration method Impact: Very High – All user account data can be leaked and may even include admin account data
System:	All
Tools Used:	Burp Suite, Sqlmap
References:	

Evidence

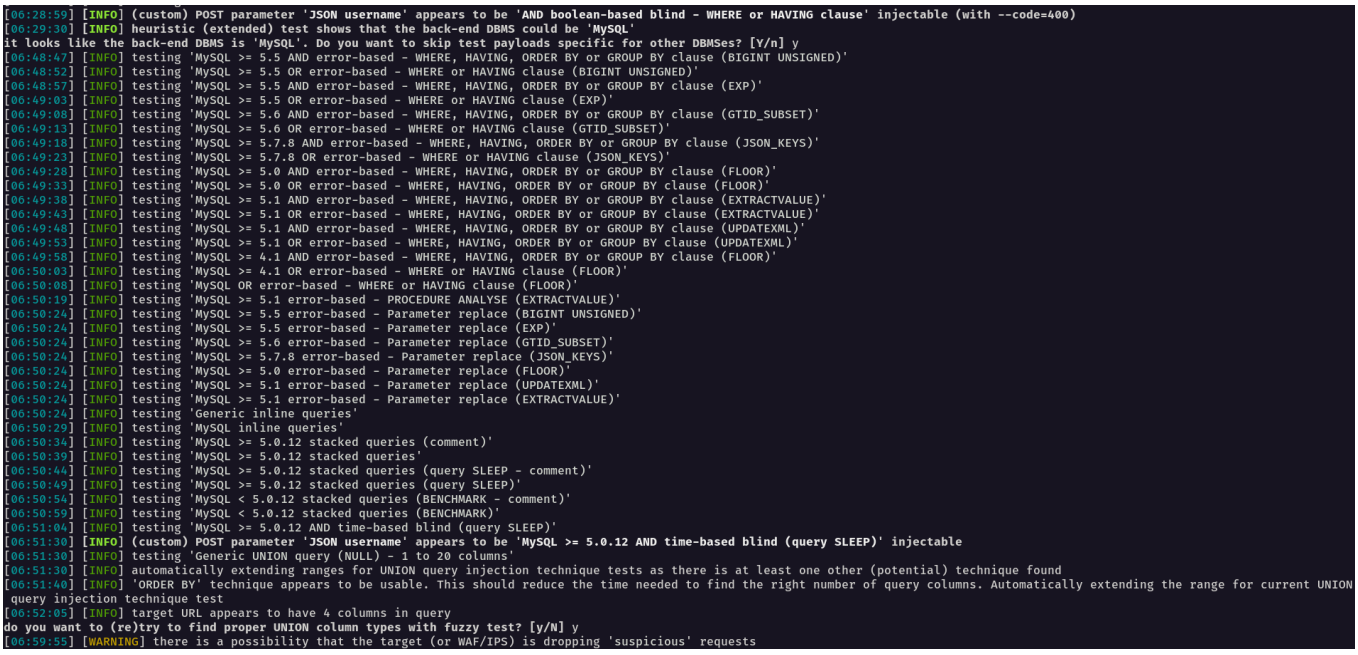


Figure 1: Script after run the sqlmap command

Remediation

Using ORM (Object-Relational Mapping) frameworks that inherently protect against SQL injection, and conduct thorough code reviews and implement automated testing to identify and mitigate SQLi vulnerability to prevent database access via sql injection.

Finding IPT-002: Can change password after getting the database (Broken Access Control)

Description:	After getting the database, the attacker might be using the information that they get from it, especially the user account. When changing password, the authorization is using Secret Answer which is easily get from the database that got breached before.
Risk:	Likelihood: Very High – After getting the database, the chance is very high for attacker to try to access the account and changing it password Impact: Very High – Users can lose their account and even have admin access
System:	All
Tools Used:	Burp Suite
References:	

Evidence

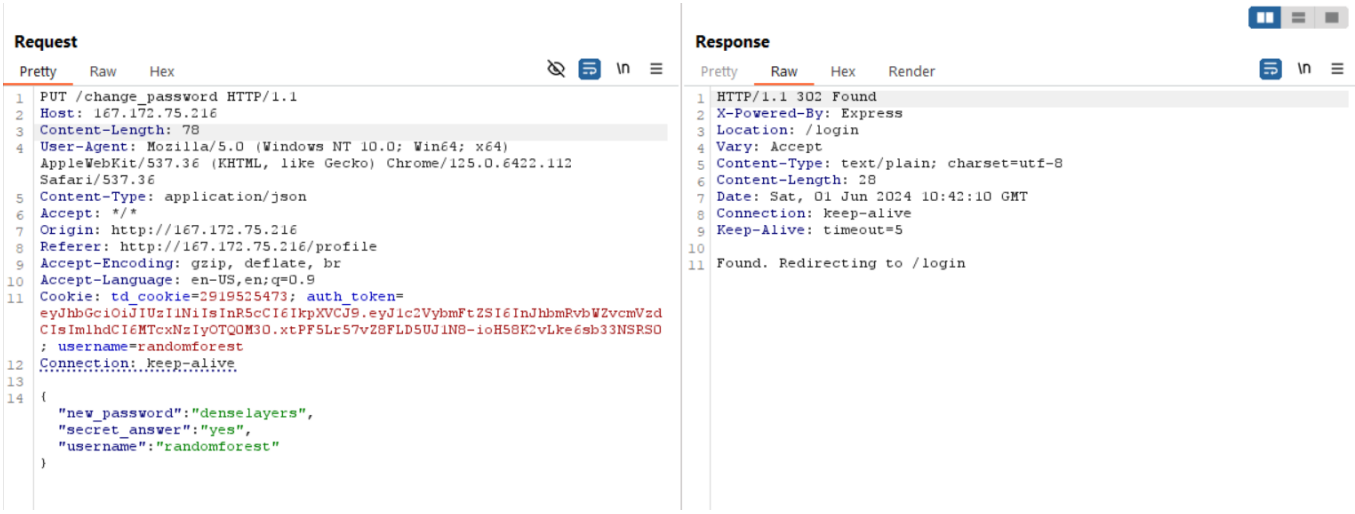


Figure 2: Change the password of first account

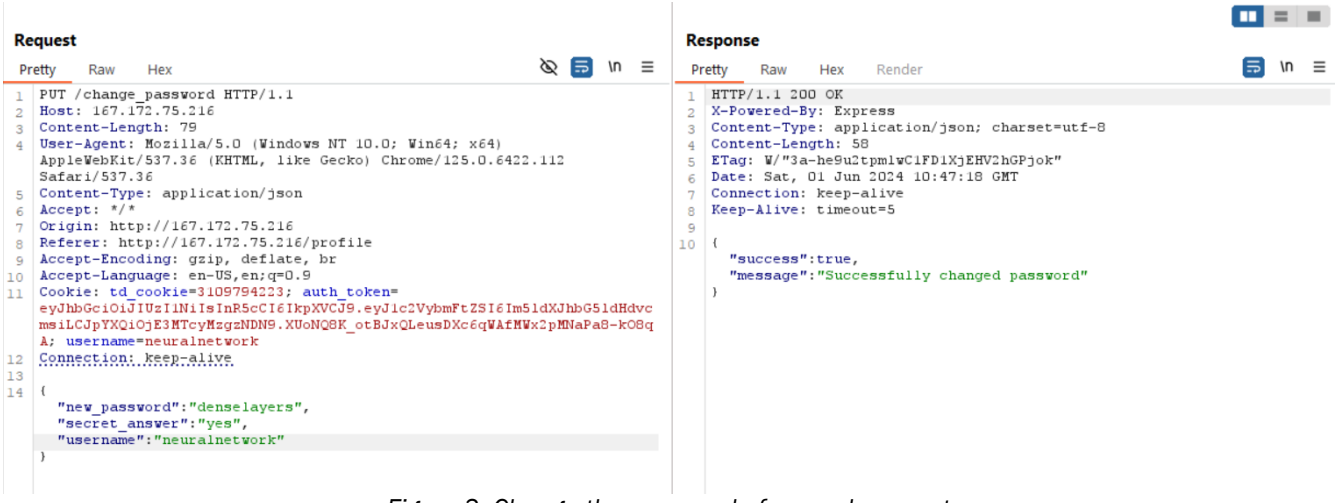


Figure 3: Change the password of second account

Request

Pretty Raw Hex

```
POST /login HTTP/1.1
Host: 167.172.75.216
Content-Length: 56
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.6422.112
Safari/537.36
Content-Type: application/json
Accept: */*
Origin: http://167.172.75.216
Referer: http://167.172.75.216/login
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: td_cookie=2919525473
Connection: keep-alive

{
  "username":"neuralnetwork",
  "password":"s3cretMessage?"
}
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 401 Unauthorized
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 58
ETag: W/"3a-fkOCQAqVT8Ghd+OoLFw36R215DQ"
Date: Sat, 01 Jun 2024 10:47:50 GMT
Connection: keep-alive
Keep-Alive: timeout=5

{
  "success":false,
  "message":"Invalid username or password"
}
```

Figure 4: Login using old password of second account

Request

Pretty Raw Hex

```
POST /login HTTP/1.1
Host: 167.172.75.216
Content-Length: 53
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.6422.112
Safari/537.36
Content-Type: application/json
Accept: */*
Origin: http://167.172.75.216
Referer: http://167.172.75.216/login
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: td_cookie=2919525473
Connection: keep-alive

{
  "username":"neuralnetwork",
  "password":"denselayers"
}
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
X-Powered-By: Express
Set-Cookie: auth_token=
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6Im51dXJhbG51dHdvc
msiLCJpYXQiOjE3MTcyMzg4OTV9.5bDZmAj6UZnYb7bHbsTOEDPXWZkmxLdLCmquc235pE
Y; Path=/; HttpOnly
Set-Cookie: username=neuralnetwork; Path=/; HttpOnly
Content-Type: application/json; charset=utf-8
Content-Length: 46
ETag: W/"2e-C9NpmX7OzdNmNDHp1WOc4SLXeMQ"
Date: Sat, 01 Jun 2024 10:48:15 GMT
Connection: keep-alive
Keep-Alive: timeout=5

{
  "success":true,
  "message":"Login successful!"
}
```

Figure 5: Login using new password of second account

Remediation

Using email as an authorization for changing password

Last Page