****

# Artemis Financial Vulnerability Assessment Report

Table of Contents

[Document Revision History 3](#_Toc32574607)

[Client 3](#_Toc32574608)

[Instructions 3](#_Toc32574609)

[Developer 4](#_Toc32574610)

[1. Interpreting Client Needs 4](#_Toc32574611)

[2. Areas of Security 4](#_Toc32574612)

[3. Manual Review 4](#_Toc32574613)

[4. Static Testing 4](#_Toc32574614)

[5. Mitigation Plan 4](#_Toc32574615)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **[Date]** | **[Your name]** |  |

## Client



## Instructions

Submit this completed vulnerability assessment report. Replace the bracketed text with the relevant information. In the report, identify your findings of security vulnerabilities and provide recommendations for the next steps to remedy the issues you have found.

* Respond to the five steps outlined below and include your findings.
* Respond using your own words. You may also choose to include images or supporting materials. If you include them, make certain to insert them in all the relevant locations in the document.
* Refer to the Project One Guidelines and Rubric for more detailed instructions about each section of the template.

## Developer

Wesley Blackwell

## Interpreting Client Needs

Secure communications are valuable to Artemis because they handle sensitive, classified client information. According to the description, the business will not be bound to the U.S. The government has passed multiple rules and regulations on software security for financial institutions. One example is the Payment Card Industry Security Standards Council. This entity has established cybersecurity controls and business practices that any company accepting credit card payments must implement. One government restriction act passed is the Gramm-Leach-Bailey Act 2022. This act requires the financial institution to explain their information-sharing practices with their clients. Cyber attacks are an immediate threat, and in the future. One form of this is SQL injection. This type of cyber attack can lead to DOS (denial of service), hijacking of servers, or many other goals of attackers. Modern requirements to consider are to have a maintenance team perform routine security checks, provide updates for applications, monitor login fails, and most importantly watch for breaching attempts actively.

## Areas of Security

Input validation and authentication – Users must be validated and their access authenticated not only upon login but when making selections inside the application.

API – The client is running a RESTful API.

Cryptography – Data transferring and static data needs to be encrypted to protect the business and the customers.

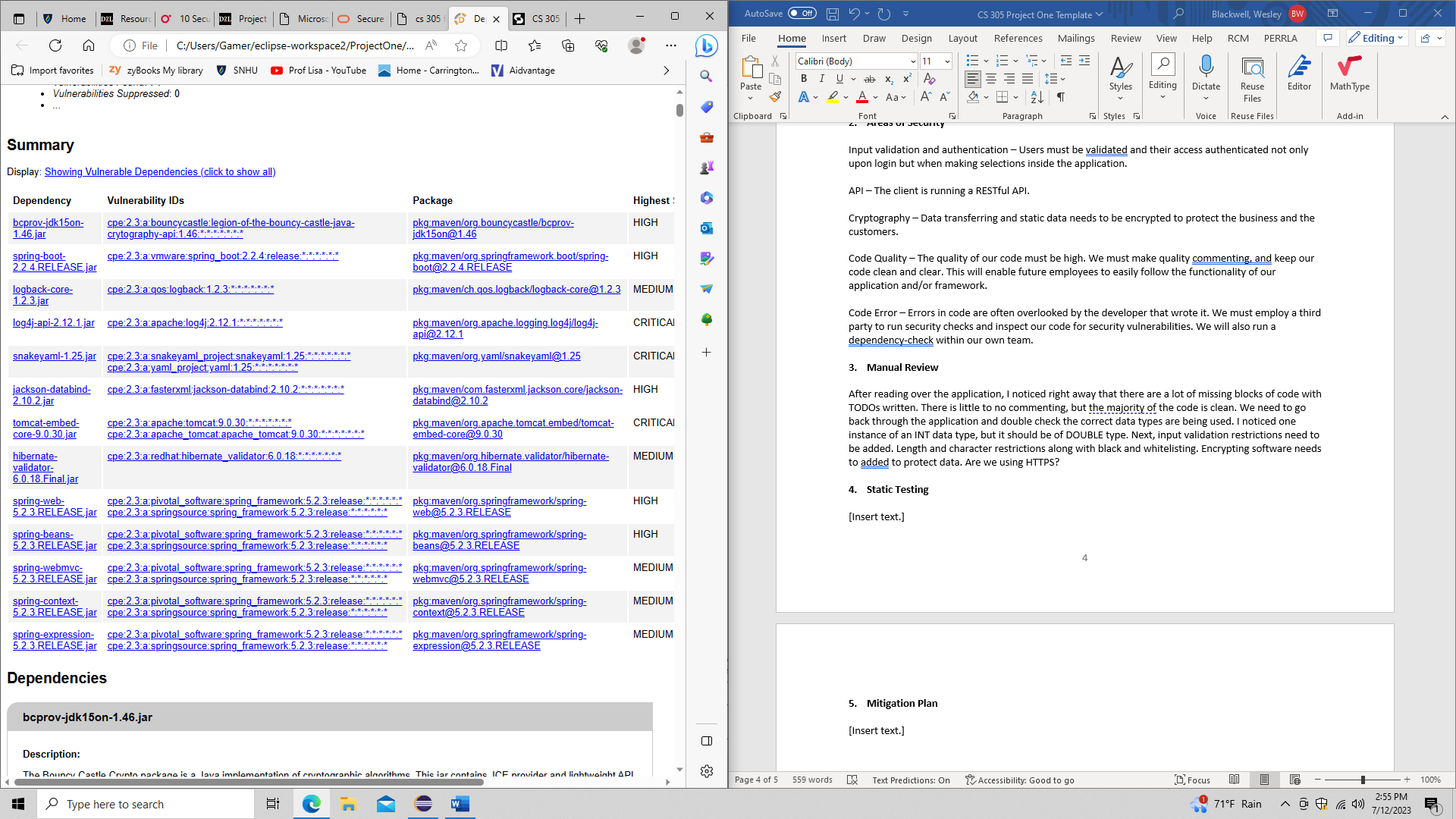
Code Quality – The quality of our code must be high. We must make quality commenting, and keep our code clean and clear. This will enable future employees to easily follow the functionality of our application and/or framework.

Code Error – Errors in code are often overlooked by the developer that wrote it. We must employ a third party to run security checks and inspect our code for security vulnerabilities. We will also run a dependency-check within our own team.

## Manual Review

After reading over the application, I noticed right away that there are a lot of missing blocks of code with TODOs written. There is little to no commenting, but the majority of the code is clean. We need to go back through the application and double check the correct data types are being used. I noticed one instance of an INT data type, but it should be of DOUBLE type. Next, input validation restrictions need to be added. Length and character restrictions along with black and whitelisting. Encrypting software needs to added to protect data. Are we using HTTPS?

## Static Testing



bcprov-jdk15on-1.46.jar

spring-boot-2.2.4.RELEASE.jar

logback-core-1.2.3.jar

log4j-api-2.12.1.jar

snakeyaml-1.25.jar

jackson-databind-2.10.2.jar

tomcat-embed-core-9.0.30.jar

hibernate-validator-6.0.18.Final.jar

spring-web-5.2.3.RELEASE.jar

spring-beans-5.2.3.RELEASE.jar

spring-webmvc-5.2.3.RELEASE.jar

spring-context-5.2.3.RELEASE.jar

spring-expression-5.2.3.RELEASE.jar

Each of these has an updated version that has addressed these vulnerabilities.

## Mitigation Plan

In order to mitigate vulnerabilities, we need to get the software updated to the current versions first. Next, we need add most secure input validation and authentication with blacklisting, whitelisting, character restrictions, and length limitations. Any inputs from users need to be validated through every action. Encryption of data during transfer and in static state is needed to secure the customers data. We should incorporate HTTPS to utilize its security capabilities. Transport Layer Security will be important for communications inside of our network.