****

# CS 305 Project One

**Artemis Financial Vulnerability Assessment Report**

**Michael R. Jones II**

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** | **1 June 2021** | **Michael R Jones II** |  |

## Client



## Instructions

Deliver this completed vulnerability assessment report, identifying your findings of security vulnerabilities and articulating recommendations for next steps to remedy the issues you have found.

Respond to the five steps outlined below and include your findings. Replace the bracketed text on all pages with your own words. If you choose to include images or supporting materials, be sure to insert them throughout.

## Developer

Michael R Jones II

## 1. Interpreting Client Needs

Determine your client’s needs and potential threats and attacks associated with their application and software security requirements. Consider the following regarding how companies protect against external threats based on the scenario information:

* What is the value of secure communications to the company?
* Are there any international transactions that the company produces?
* Are there governmental restrictions about secure communications to consider?
* What external threats might be present now and in the immediate future?
* What are the “modernization” requirements that must be considered, such as the role of open source libraries and evolving web application technologies?

Security overall and mainly secure communications is very important to Global Rain. Being a financial institution, if the wrong people gained access to that information it can be detrimental to the customers. It doesn’t explicitly say that they will be working with only clients in the US or international clients. When looking at governmental restrictions, contacting the FTC will be the way to go. Open Source libraries can be a phenomenal tool but also very sketchy when thinking about security. If a developer found some open source code to use to complete their specific task and they use it, then it could be easily crackable for attackers.

## 2. Areas of Security

Referring to the Vulnerability Assessment Process Flow Diagram, identify which areas of security are applicable to Artemis Financial’s software application. Justify your reasoning for why each area is relevant to the software application.

I would think that all of the areas will be relevant points of security for the application. However, there are a couple that I think would be the most important. These processes are APIs and Client/Server. Given that the application being developed, the client/server relationship is extremely important. How the application interacts with servers can be easily penetrable if the server is infected and security is not a priority within the application’s development. Also, APIs are very important. APIs are usually managed by a 3rd party outside of your organization. For example, if there were some data to pull stock/financial data to the application. This could be easily penetrable by attackers because those who developed the API may not have taken security as a priority.

## 3. Manual Review

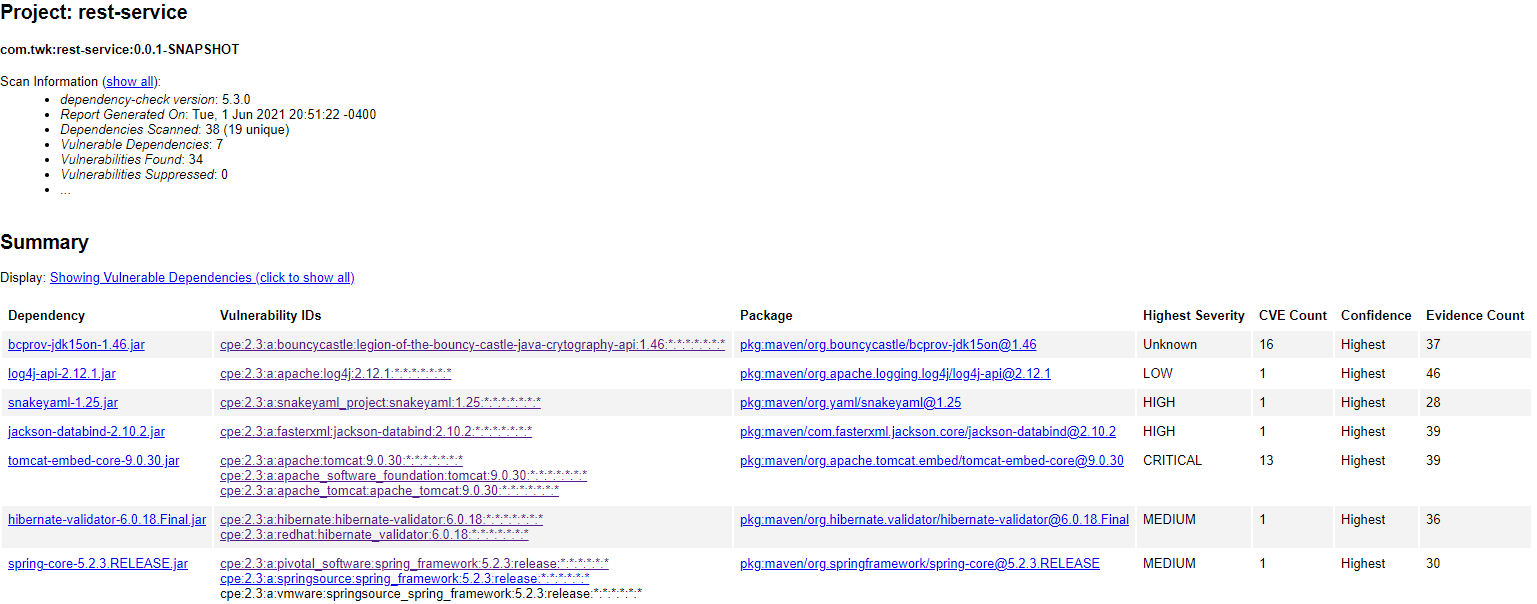
Continue working through the Vulnerability Assessment Process Flow Diagram. Identify all vulnerabilities in the code base by manually inspecting the code.

The main issue that I found is that in the DocData.java file, there is a local variable without distinction. Perhaps this could be a method of entry if a user is meant to input the variable.

## 4. Static Testing

Run a dependency check on Artemis Financial’s software application to identify all security vulnerabilities in the code. Record the output from dependency check report. Include the following:

1. The names or vulnerability codes of the known vulnerabilities
2. A brief description and recommended solutions provided by the dependency check report
3. Attribution (if any) that documents how this vulnerability has been identified or documented previously



* bcprov-jdk15on-1.46.jar - Bouncy Castle Java Crypto APIs using external controlled inputs
* log4j-api-2.12.1.jar – Improper certificate validation.
* snakeyaml-1.25.jar – allows entities to expand during load operations
* jackson-databind-2.10.2.jar- another error where entity expansion was not secured properly
* tomcat-embed-core-9.0.30.jar - a fix for a CVE was incomplete. Related to using Apache Tomacat
* hibernate-validator-6.0.18.Final.jar – a flaw within hibernation validator
* spring-core-5.2.3.RELEASE.jar – an opening against protections against RFD attacks

## 5. Mitigation Plan

After interpreting your results from the manual review and static testing, identify the steps to remedy the identified security vulnerabilities for Artemis Financial’s software application.

To fix these major issues I would first make sure that certificates are validated. Being a web-based application, web certificates can be a back breaker. Also, fixing the API issue for external inputs can be done with not allowing external inputs. Also, fixing the entity expansion issues will be another top priority. I did not find any ways to specifically fix the entity expansion.