****

# 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** | **1/22/24** | **Dylan Ackron** |  |

## 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

Dylan Ackron

## Interpreting Client Needs

Secure communications are very important for this business because personal data could be sent between the two parties. Making sure sensitive data is guarded by the company is essential for Artemis Financial, having data breaches doesn't promote good security for the company. When it comes to international transactions, the scenario does explicitly say that there are international transactions being dealt with, but being a financial company its very likely that they do. If they in fact do deal with international transactions, then that adds for additional security measures. Depending on where Artemis Financial is located, the laws on secure communications can differ. In the United States its hard to say that there's any singular federal law regulating it, however there may be state laws that do. The threats that are prevelant to Artemis Financial are pretty straightforward, being data breaches, different types of cyberattacks, and exploitation of vulnerabilities. Modernization requirements that must be considered such as open sourced libraries could possibly create new vulnerabilities, but could also decrease development time. If maintained properly it should show no issues. For evolving web application technologies, this could help improve security and function of the application.

## Areas of Security

* Input Validation: Input validation would provide security and protection for Artemis Financial’s customers, input validation for users is important for protecting information.
* API’s: Using an API is necessary to choose which data can be accessed by the user.
* Cryptography: cryptography is crucial in this situation, with cryptography it would be helping to make sure that user’s information isnt compromised in transit.
* Code Error: Handling code error will allow for things to be fixed to prevent future vulnerabilities.
* Code Quality: This is a must, code quality will ensure that the best coding techniques and strategies will be used to ensure the security of Artemis Financial.

## Manual Review

* I struggled to find any secure input validation in the code
* No validation of requests
* No cryptography
* Application does utilize https when sharing data

## Static Testing

[Insert text.]

* bcprov-jdk15on-1.46.jar - Multiple vulnerabilities, suggestion is to update to 1.6 or later
* spring-boot-2.2.4.RELEASE.jar - multiple security vulnerabilities, update per vendors recommendation
* logback-core-1.2.3.jar - update past 1.4.11 to fix denial of service vulnerabilities
* log4j-api-2.12.1.jar - Vulnerable to multiple attacks on different versions, update per vendor recommendations.
* snakeyaml-1.25.jar - vulnerable to multiple types of attacks on different versions, update per vendor recommendations.
* jackson-databind-2.10.2.jar - Update to current version
* tomcat-embed-core-9.0.30.jar - multiple vulnerabilities, update to 10.0.6 and beyond.
* hibernate-validator-6.0.18.Final.jar - flaw was found in 6.1.2 final, update past this version

## Mitigation Plan

The steps that are needing to be taken to fix vulnerabilities is to update current dependencies as per vendor recommendation to ensure we are using the most secure versions. Next after the manual review I would suggest implementation of more input validation, as well as the implementation of HTTPs to secure communications throughout the system. The use of cryptography would also greatly boost security in the system, as well as further validating requests made by the application/user.