****

# 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** | **04-21-23** | **Karama Crager** | **Checkup** |

## 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

Karama Crager

## Interpreting Client Needs

The client, Artemis Financial, are looking to modernize their operations. This modernization means a newer version of up-to-date and effective security software is required. The goal the client has prompt us with is to protect the organization from external threats. Considering this client deals with savings, retirement, investments, and insurance, there’s a lot to consider when talking about external threats. Threats could be spoofing, hacking, or leaking data, to name a few that could be considered external threats.

## Areas of Security

For the program, it’s necessary to consider input validation to be an important part of the vulnerability assessment. API’s are necessary for the program to have secure interactions with the API, and cryptography is needed for encryption. We’ll need encapsulation for secure data structures, while code error for vulnerability identification.

## Manual Review

Looking through the code everything looks fine. The only issue that could be a problem is one found in Eclipse. There’s the consideration of a specific version of compiler and JRE required for the program as far as issues, but everything seems to be well written with the checked coding.

## Static Testing

Doing static checking, it seems the issue is outdated spring and .jar files that are integrated into the project. The most important issue that was found would be with the bcprov-jdk15on-1.46.jar dependency. This deals with the cryptography, which is very important considering we need to have secured data structures. The Bouncy Castle Java library that’s currently being used (CVE-2013-1624) seems to have the issue of not properly timing side-channel attacks on a noncompliant MAC.

## Mitigation Plan

Overall, I’d say the best way to increase security and modernize this program would be to update the spring framework, while also updating the rest of the dependency files. For the coding, I’d do a double check and make sure everything’s up to date (ex. We found a different version of complier being used) while also checking to make sure that the code is all being used and is used properly.