****

# 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** | **3/18/2023** | **Anthony D’Angelo** |  |

## 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

Anthony D’Angelo

## Interpreting Client Needs

This client in particular values security very high. Artemis Financial’s mission statement is, “Security is everyone’s responsibility.” Given that this company is a financial company, and they have a RESTful API, I would assume that the company is making international transactions. Depending on the company’s location(s), governmental limits on secure communications may exist. Encryption legislation is one to be aware of. There are external threats that might be present now or soon. A couple are cyberattacking and application vulnerabilities. Modernization requirements that we must consider would be to keep all the software up to date, run dependency checks, and keep an eye on all the vulnerabilities that could be present within the software/application/API.

## Areas of Security

The following are areas of security we should consider:

* Cryptography: This will be good to have to reduce external threats.
* Input Validation: Users will be inputting data, so this is a must.
* Code Error: Most software applications should have this. We do want to make sure this is in place for Artemis.
* Code Quality: Dealing with personal data, we need to follow secure coding practices.
* Encapsulation: Assuming there will be large amounts of data, and we are using data structures, we want to ensure that they are all secure.

I did not add APIs or Client/Server because those seem to already be in place for the company. But a quick look-through couldn’t hurt.

## Manual Review

While reviewing the code base I noticed that there is not a lot of input validation. This will need to change to make the application more secure. Another thing I noticed is that all the class/functions are public. We could think about making changes to make those final or private where needed. We will also need to update the Maven version from 5.3.0 to 8.1.2.

## Static Testing

The following are screenshots of the dependency check:

Text

Description automatically generated

Graphical user interface, text, application, email, Teams

Description automatically generated

The dependency check shows that there are 38 in total, with 22 unique dependencies. The report also shows that there are 101 vulnerabilities found.

## Mitigation Plan

1. We would need to dig into the vulnerabilities and cherry pick which ones we need to fix depending on the application’s needs.
2. We would then need to update/edit/refactor the code base according to “Areas of Security” and “Static Test” section documented above.
3. We would need to evaluate the cost for our approach.
4. Finally, we would need to continuously check the code base and application for security vulnerabilities.