****

# 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** | **7/10/2023** | **Marcus Smith** |  |

## 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

Marcus Smith

## Interpreting Client Needs

Artemis Financial wishes to modernize their operations and to update their RESTful web application API to the most current and effective software security. They wish to protect their organization from external threats. The company takes user information and in turn allows them to view their account and their funds, which would mean we need to secure personal information and protect it from outsider.

## Areas of Security

When it comes to the RESTful web application the areas of security that need to be taken into consideration are APIs, Encapsulation, Input Validation, and Client/Server.

When it comes to the APIs, being that the application will run as a web browser, securing what access the user has with the application and also preventing improper access on the site by other people being that it will be over the internet is crucial for the company.

Encapsulation is important for access to certain information will need to be guarded and well protected, such as the users account information and funds that they have in their personal accounts.

Input Validation works in tandem with encapsulation, as it will need to be used to allow those who need access to the more internal part of the system, the same security to the users who will be using the system. Making sure to protect and secure the users from both the admin standpoint and the customer standpoint to allow the application to be used with confidence.

Client/Server security will also be important, for the application will be running over the web, and having the right client and servers that will provide web support will be crucial since the application will be a web application.

## Manual Review

When looking over the code base I can see that encapsulation will need to be used in the greeting class as well as the customer class, to help hide their Id as well as their account number and account balance for only the customer to view and access. This leans into the importance of making sure that the application supports accurate input validation protocols. Making sure that the client/servers are secure is also important, for the application will need to be able to call the right screens and information to display and be called. Making sure that the application runs smoothly is as important. Because the client and servers are running over the web, the security of the APIs play a part in allowing for system to run smoothly. By protecting the application from improper access by outside parties, we will be able to run the application as intended.

## Static Testing

The list of vulnerabilities of this application are as follows:

Snakeyaml-1.25.jar – In which the constructor not restricting types which can be instantiated during deserialization. Upgrading to version 2.0 and beyond is recommended.

Spring-core-5.2.3.RELEASE.jar, Spring-web-5.2.3.RELEASE.jar, Spring-webmvc-5.2.3.RELEASE.jar – The spring framework suffers from potential remote code execution, and depending on the library and its implementation, authentication may be required.

Tomcat-embed-core-9.0.30.jar, tomcat-embed-websocket-9.0.30.jar – if tomcat is configured to ignore invalid HTTP headers via setting rejectIllegalHeader to false, it did not reject requests containing an invalid content-length header. This allows for smuggling attacks possible due to failure to reject requests with invalid headers.

## Mitigation Plan

After looking over the manual review and the static testing, by upgrading to the current versions will help mitigate some of these issues. Also making sure the application rejects invalid headers and making sure that the implementation of the spring framework is correct to defend from remote code execution.

Dependency Check

A screenshot of a computer

Description automatically generated

Resources:

Manico, J. (n.d.-b). *Iron-Clad java*. O’Reilly Online Learning. <https://learning.oreilly.com/library/view/iron-clad-java/9780071835886/?sso_link=yes&sso_link_from=SNHU>

*Secure Coding Guidelines for Java SE*. (n.d.). https://www.oracle.com/java/technologies/javase/seccodeguide.html