****

# 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** | **03/24/2024** | **Roxanna Wing** |  |

## 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

Roxanna Wing

## Interpreting Client Needs

Artemis Financial considers securing communications to be a top priority. The company strives to maintain confidentiality during the transmission of sensitive data. Part of this secure communication includes international transactions. During these international transactions, it is vital that Artemis Financial complies with any data protection regulations as they are defined in the different jurisdictions. Depending on what regions Artemis Financial conducts business will determine what restrictions may be imposed by the local governments. In order to avoid liabilities Artemis Financial needs to ensure they are in compliance with any government restrictions. Artemis Financial may be vulnerable to a number of external threats such as data breaches, injection attacks, and phishing attacks. If Artemis Financial wants to modernize, they should make sure their libraries stay up to date so they are able to implement any fixes for any bugs that may appear.

## Areas of Security

Input Validation – Input validation is vital anytime an external source can input information. In order to avoid any injection attacks, Artemis Financial needs to be sure that there is sufficient input validation throughout its program.

API’s – Since this program will be used by parties both inside and outside of the company, a secure API will be necessary. It is vital that the program is able to perform proper authentication for each user log-on. Artemis Financial will also need to ensure they are not accidentally exposing any sensitive data through improper error handling or poor session management.

Cryptography – Since Artemis Financial will be dealing with Global customers and their sensitive data, it is vital that cryptography is involved. All information regarding the customers needs to be secured and securely encrypted.

Code Quality – The code should be of high quality. It should follow all updated guidelines and should be run on a recently updated version. Secure coding practices should be followed.

## Manual Review

While reviewing DocData.java shows some vulnerabilities. The exception handling with ‘SQLException’ only prints the stack trace and may leak vulnerable information. DocData.java also does not have any input validation with means it could be vulnerable to SQL injections. GreetingController.java also lacks input validation. In CRUDController.java the value ‘business\_name’ is received, but there is not input validation.

## Static Testing

Below is a screenshot of the static testing:

A screenshot of a computer

Description automatically generated

bcprov-jdk15on-1.46.jar – can allow extra elements to be injected into the signature sequence, need to update to a newer version as the 1.46 version is outdated

hibernate-validator-6.0.18.Final.jar – a bug that can allow a message in the processer to validate invalid EL expressions

jackson-databind-2.10.2.jar – allows vulnerability of data integrity and denial of service – newer versions offer a patch

log4j-api-2.12.1.jar – invalid verification with host, connections may be intercepted by man-in-the-middle attacks – patched in a newer version.

logback-core-1.2.3.jar – may allow an attacker to mount a denial of service attack

snakeyaml-1.25.jar – could allow remote code execution, upgrade to 2.0 and newer

spring-boot-2.2.4.RELEASE.jar – cloud foundry could be susceptible to security bypasses, update to newer version.

spring-boot-starter-web-2.2.4.RELEASE.jar - cloud foundry could be susceptible to security bypasses, update to newer version.

spring-core-5.2.3.RELEASE.jar – vulnerable to remote code execution- update to patched version

spring-web-5.2.3.RELEASE.jar - vulnerable to remote code execution- update to patched version

spring-webmvc-5.2.3.RELEASE.jar - vulnerable to remote code execution- update to patched version

tomcat-embed-core-9.0.30.jar – various exploits available to an attacker – update to patched version

tomcat-embed-websocket-9.0.30.jar - various exploits available to an attacker – update to patched version

## Mitigation Plan

Most of the dependency errors can be solved by updating the versions to the newer/ patched version. The code needs to be updated and for input validation and secure coding practices should be followed.