****

# 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** | **13 Nov 2022** | **Justin M. Dougherty** | **INITIAL DRAFT** |

## 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

Justin Dougherty

## Interpreting Client Needs

Artemis Financial, provides financial planning services for their trusted customers. This highly sensitive information must remain secure for the integrity of the company as well as the safety and security of their customers. All data pertaining to this critical information must be secure. Artemis Financial, in addition to appeasing their customers, must also follow government regulations with all financial transactions and communications. Adherence to the government regulations will require data retention policies and security requirements to be current and consistent with the demands of the regulations set forth. If not appropriately secured RESTful APIS like Artemis Financial will be highly susceptible to data interception. One of the largest data interception threats to Artemis Financial will be through phishing attempts. These attempts will likely be in effort to gain customers log in information. Two-factor authentication and implementing open-source libraries in their software can be simple solutions to surface needs. The significant number of individuals testing and maintaining the software can ensure that any bugs or coding errors can be resolved expeditiously.

## Areas of Security

Input Validation- As we discussed with phishing threats. Input validation is significant, RESTful API will accept user input that must be validated. This is also a valid deterrent to attacks via injection methods.

APIs- The RESTful web application or API, is at risk of attack via injection, broken authentication and other prominent data interception attempts.

Code Error- If errors are too transparent attackers could have insight to the systems operations and expose vulnerabilities

Code Quality- The code should be tested for bugs and errors and immediately fixed. The data is highly sensitive and the coding methods should reflect its sensitivity.

Cryptography- Financial data, especially that of which is being translated and subject to federal inspection should be encrypted. All customer data should be encrypted to protect the customer.

## Manual Review

Identified vulnerabilities by way of manual code inspection include:

There are a significant amount of dependencies that should be updated to the most current version regularly.

The code has no authentication methods, in the Customer.java class

There is nothing in place to validate request parameters

The @RequestParam reads HTML data from the user and appears to be open to injection attacks

Few security measures completely implemented

## 

## Static Testing

Graphical user interface, application, Teams

Description automatically generatedGraphical user interface, application, Teams

Description automatically generatedPlease see below for static testing results:

## 

Findings and potential solutions of this test include:

bcprov-jdk15on-1.46.jar – CVE-2016-1000338 – Findings show that on this version there isn’t complete validation encoding of signature on verification. Injection threats and data mismanagement are possible if dependency isn’t updated to the newest release.

hibernate-validator-6.0.18.Final.jar - CVE-2020-10693 –Input validation is insecure, leaving the validation methods subject to attackers if not updated to the newest version.

jackson-databind-2.10.2.jar – CVE-2020-25649 – An XML vulnerability that can be resolved by updating to the newest release.

log4j-api-2.12.1.jar – CVE-2020-9488 – Upgrading to the newest version can prevent possible log message leaks.

logback-core-1.2.3.jar – CVE-2021-42550 and logback-core-1.3. snakeyaml-1.25.jar – CVE-2017-18640 –can be easily resolved by updating to to snakeyami-1.31.jar.

## Mitigation Plan

Updating the most current dependencies on a regular basis would be the most practical form of mitigation. After doing so running another static test could identify any additional security issues. Adding input validation and authorization control methods to the GreetingController.java.class and Customer.java classes respectfully are two early implementations of many of the required security methods that need to be added.