****

# 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** | **September 18th, 2022** | **Nevena Young** | **Project One** |

## 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

Nevena Young

## Interpreting Client Needs

The client, Artemis Financial, is a consulting company that handles sensitive information. The client wants to modernize their operations, in doing so would also need current and effective software security.

The client would need an accurate and secure log in method. This method would require a username and password. After the user’s log in information is authenticated, they would be granted their level of access to the company’s website.

Their website uses APIs (Application Program Interface), therefore they would need protection from external threats and leaks. This is imperative due to them being international. When a company is international, they need to abide by the restrictions and laws of their geographic location. This is an urgent matter due to hackers and potential leaks.

## Areas of Security

Input Validation – Validating all input would keep the system secure and assist in preventing attacks

APIs – A RESTful API is needed to allow access from outside of the system. We would need to create the API in a way that would not allow unexpected access to the system.

Cryptography – Since we are using APIs, we would need to encrypt all data, ours and the customer’s, to keep sensitive information safe.

Client/Server – We must ensure that proper certificates are used to ensure the data is safe while being transferred via https requests.

Code Error – We would need to review the code, input functions, and any API access layer code.

Encapsulation – If a command input function needs access to the stored data in our system, we must make sure it is being used properly and not messing up any data.

## Manual Review

While going over Greeting.java, GreetingController.java, and Application.java I noticed the version were out of date… The current version is 4.0.0 and the system is dependent on an outdated version, 2.2.4 RELEASE.

This is something that needs to be addressed due the system being improperly secured. Having an outdated version operating on an updated version is bound to run into problems due to the outdated security within the outdated version. To fix this, we would simply need to update the system, therefore all versions of any program will be current.

Another security flaw I seen was on lines 27 - 30 in the GreetingController.java file. The method used to script these lines has a major security flaw! The data is taken from parser and transported to the template via plain text. This is a hacker’s dream! One could simply form their own string and upload it to the system’s template, in doing so making it possible for them to implement the system. To fix this we would need to incorporate a size limit for the string or use encryptions.

## Static Testing

Text

Description automatically generated

A picture containing text, indoor, open

Description automatically generated

## Mitigation Plan

1. We will ensure to update the dependency version to the current version.
2. We need to attach some type of defense to the system that would detect problematic string patterns, as well as finding a more secure way to transport data from the parser to the template.