

# CS 305 Project One

**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** | **1-17-21** | **Keith Ellison** |  |

## Client



## Instructions

Deliver this completed vulnerability assessment report, identifying your findings of security vulnerabilities and articulating recommendations for next steps to remedy the issues you have found.

Respond to the five steps outlined below and include your findings. Replace the bracketed text on all pages with your own words. If you choose to include images or supporting materials, be sure to insert them throughout.

## Developer

Keith Ellison

## 1. Interpreting Client Needs

Determine your client’s needs and potential threats and attacks associated with their application and software security requirements. Consider the following regarding how companies protect against external threats based on the scenario information:

* If connections aren’t secure and/or encrypted, then malicious actors can infiltrate client’s accounts or possibly even sensitive information related to Artemis itself.
* Yes, their business model includes providing it’s services to global clientele.
* Some governments impose export restrictions on information
* Hackers looking to target the clients of a group as large as Artemis Financial comes to mind as the most unique threat; needing to be able to verify logins and tasks like that are common to web-connected applications.
* Open source libraries help a modern program to be more modular so that it can adapt to further tasks over time. Also to make all or most of Artemis’s services available through their modernized application.

## 2. Areas of Security

Referring to the Vulnerability Assessment Process Flow Diagram, identify which areas of security are applicable to Artemis Financial’s software application. Justify your reasoning for why each area is relevant to the software application.

Cryptography – Such important communications between Artemis and it’s clients will need to be encrypted.

A modern API – A strong API will unify the diverse systems of a new, more powerful financial app that can do more than manage large amounts of money.

Client/Server resources – An area that would obviously need strengthening in a new generation of a program like this; managing communications over such sensitive information like this, this area is the most attractive to “bad actors” meaning to cause harm, so it needs to be fortified.

Overall “secure” coding – There are certain practices that can be implemented program wide concerning how the program is written; the code itself can be modular and “faults” in how things need to work can be contained, potentially exposing any further vulnerabilities.

## 3. Manual Review

Continue working through the Vulnerability Assessment Process Flow Diagram. Identify all vulnerabilities in the code base by manually inspecting the code.

Data access/ Access to data – DocData.java exposes the location of sensitive user information

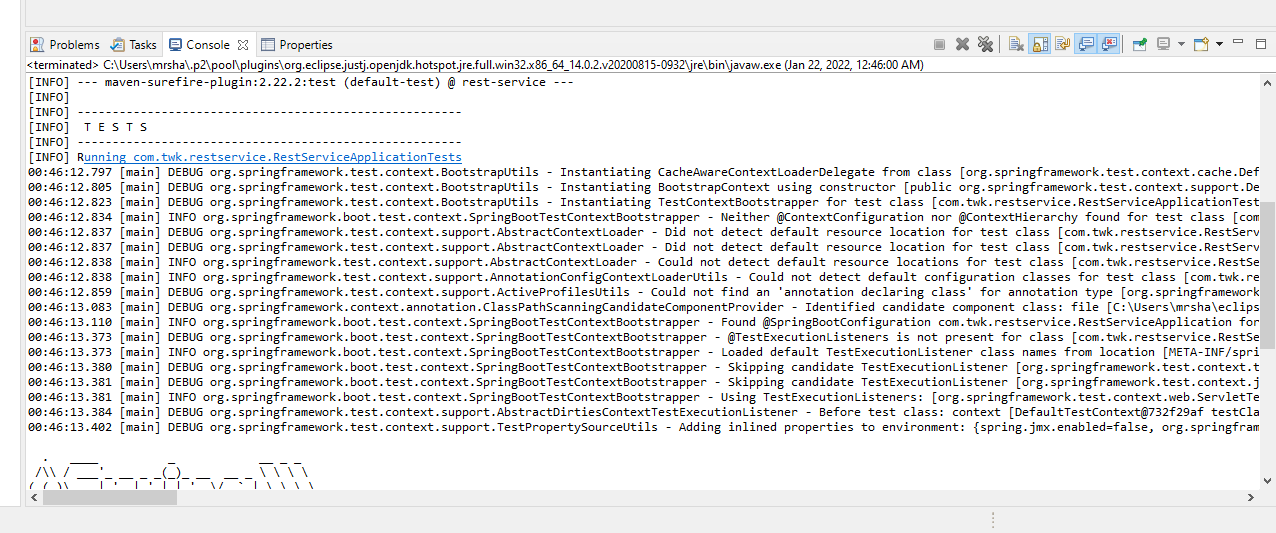
Direct object references – The above can apply here by it’s references to sensitive, protected information. It includes CrudController.java where there is a method that could expose the DocData file of a company without a user having proper clearance to see it.

## 4. Static Testing

Run a dependency check on Artemis Financial’s software application to identify all security vulnerabilities in the code. Record the output from dependency check report. Include the following:

1. The names or vulnerability codes of the known vulnerabilities
2. A brief description and recommended solutions provided by the dependency check report
3. Attribution (if any) that documents how this vulnerability has been identified or documented previously





## 5. Mitigation Plan

After interpreting your results from the manual review and static testing, identify the steps to remedy the identified security vulnerabilities for Artemis Financial’s software application.

Better username and password system - Fortify the password and username system by requiring numbers and letters in a user’s name and pass.

Encryption - Of course these communications should be carried out on secure connections.

Encapsulation – Modular design in the coding itself can mitigate the damage done to the program overall if the program is built to be dependent on methods; that way, malicious activity can fail, or be caught and stopped right there.

Review your code – It’s simply a good habit, to review your code; as coders progress, we can learn to refine past work, and this program will presumably be updated creating a need for refinement.

Update the server – CVE-~~2020-9~~488 in the current state of the server

Certificate validation - Freshly validating these will go a long way towards catching intrusion attempts