****

# 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** | **March 15th** | **Samuel Bailey** |  |

## 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

Samuel Bailey

## 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:

* What is the value of secure communications to the company?

The company Artemis values security not only for themselves but for their clients that give them financial numbers of their lives.

* Are there any international transactions that the company produces?

After reviewing the code given I did not find any international transactions.

* Are there governmental restrictions about secure communications to consider?

I couldn’t tell about any ‘government’ restrictions based on the code provided.

* What external threats might be present now and in the immediate future?

Almost every line of code is vulnerable, there is ZERO input checking as well as the database can be exploited in such a way that all usernames, passwords, amount of money in account is open.

* What are the “modernization” requirements that must be considered, such as the role of open source libraries and evolving web application technologies?

Using open source libraries is never fully secure since the code is open for hackers to not only add to but also to review and easily hack. In such a program like this I would highly discredit using any and all open source code.

## 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.

Every single section of the diagram is open to be hacked. There is zero input validation. There is some open source API’s in the program which can be easily modified. For Cryptography there is hardly 5 lines of code and this can be manipulated by using the database which has zero verification as well. Following that the code is also prone to some bugs/errors and does not have any error handling except for in the database but one error exception will not save the rest of the program. Next the code is miserably commented failing best practices. However I will say the data structures to my knowledge are safe, they can be accessed far to easily but they data itself is somewhat secure.

## 3. Manual Review

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

In CRUD.java the setters and getters are public and can be easily accessed for anyone who simply runs the code. In CRUDController.java the program is using open source spring work which can be viewed manipulated and hacked due to it’s open source. In customer.java is the first time we see a private variable which is the first somewhat secure feature of this program.

## 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

According to the static testing the code aligns with things that were mentioned in the last manual review. Starting from the reliability of the open source code to validation checks throughout the entire program. Some of the highest ranked vulnerabilities were the imported packages with the open source code.

## 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.

Aside from the thought of starting from scratch, we can start by implementing our own API’s and packages to avoid using open source code. Next we should move to input validation and API validation. Then from there we can secure our database usage by using some other languages to more securely validate code.