****

# 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** | **07/24/2022** | **Zavalla Huggins** |  |

## 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

Zavalla Huggins

## 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?
* Are there any international transactions that the company produces?
* Are there governmental restrictions about secure communications to consider?
* What external threats might be present now and in the immediate future?
* What are the “modernization” requirements that must be considered, such as the role of open source libraries and evolving web application technologies?

[Include your findings here.]

When dealing with any type of financial information, privacy should be the top priority. However, the gravity of retirement information is much greater than what is typical. This money usually represents the clients hard-earned wages achieved through numerous years of commitment.

Artemis Financial does handle international transactions. This could include international wire transfers, out-of-country ATM withdrawals/deposits, or purchases through ecommerce originating from foreign countries.

The Electronic Communications Privacy Act will allow for the possibility of some personal financial information from a client to be shared with the US government. In some cases a warrant is required, however, depending the severity of the crime a warrant may not be needed to access any form of communication. Some financial transactions can catch the attention of the US government. These things could include frequent, out-of-country, wire transfers, deposits/withdrawals exceeding $10,000, or transfers to unknown contacts/entities.

One external threat is scammers. These people will pose to be a figure of high authority, and try to convince the victim to withdraw a large sum of money to avoid incarceration (which was never a possibility in the first place). Phishing scams are also used to get personal/login information from a victim. With this information, the scammer can easily drain the life-savings of a victim by either simply asking for their information, or having the victim click a malicious link, which executes code that gathers all of the users information.

In the future, as technology advances, so will the complexity of these breaches of information. To combat this, security measures must be heightened. New methods will need to be developed to combat these insecurities.

This is where modernization comes into play. Open source libraries can be a helpful source of information when designing new safety procedures. Methods used for a specific application can be adjusted and applied to the goal of Artemis Financial. On the other hand, this same information sharing is done with malicious codes. Scammers usually sell their code to anyone who can afford the price tag, and their customer can do with it as they please.

## 2. Areas of Security

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

Four main areas of security Artemis Financials would focus on is cryptography, client/server, secure coding and API. Secure coding ensures that the code is clean and efficient, meaning that the desired results are given with the least amount of code possible. This includes having secure data structures/objects instantiated appropriately, having each object be correctly associated with its attributes and methods.

Client and server is crucial to the foundation of the application since it communicates the data stored by the application to the user and vice versa. This would include things such as posting transactions the user inputs, then user then reading their transaction log.

Cryptography plays a substantial role in encrypting files to ensure that any outside entities can view the user’s sensitive information.

## 3. Manual Review

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

The code effectively uses the singleton pattern since it has objects that can have multiple instances, where there are objects that should only have one instance that modifies these other objects. However, this application is susceptible to SQL injection since the query strings are not protected.

## 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

Graphical user interface, text, application, email

Description automatically generated

1. A brief description and recommended solutions provided by the dependency check report

* Depending on the version of Java driver, the CSFLE could fail to perform valid host name verification. This leaves the program susceptible to interception of traffic between the Java driver and the KMS service. This is also a medium threat level.
* The first error detected is a bug within the message interpolation
* processor that will allow invalid EL expressions to be accepted as if they are valid, which allows attackers to infiltrate controls that function off error detection in input. This vulnerability has medium severity.
* The Alias feature in SnakeYAML 1.18 allows entity expansion during load operation. There is a high threat level.
* Spring-boot versions prior to version v2.2.11.RELEASE were vulnerable to directory highjacking. This is a high threat level.
* The program could be susceptible to remote code execution via data binding (code injection), a critical threat.

## 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.

* Performing risk-based validation can help identify vulnerabilities in data validation. By searching for input that is seen to be invalid but is still accepted, can help us know what to protect against. Also, logs should be checked again to see if the appropriate data is being logged, and errors are not slipping through the cracks.
* Because of the previous reason, therefore false positives should be inspected. Just because an input is accepted, does not always mean it is fitting into the desired criteria. The code could be missing some pivotal logic that completely changes the acceptance criteria.
* Code injection, being one of the most critical threats, should also be one of the most protected against. Such things such as special symbols for coding languages (termination characters, command delimiters, and comment marks) should be scanned for, since they could indicate code being injected into the program.
* Content control must be practiced regularly to prevent directory hijacking. Folders should not be able to be created by outside, unprivileged users. This might include blocking folders to users.