****

# 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** | **03/23/24** | **Garret Blake** | **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

Garret Blake

## Interpreting Client Needs

Artemis Financial is a company that creates financial plans for people. These plans include a variety of different services for any need a client may have. Since there are a lot of private transactions and data that will constantly be transferred from client to server, or the opposite, there needs to be high level security and communications within the system. Secure communication is critical as protects this valuable information as it is transferred from system to system. It is assumed that Artemis Financial has international transactions, as there is nothing saying that they are only United States based. Governmental restrictions that could occur when using secure communications could include regulations on surveillance or what is allowed to be encrypted and not reported. External threats that could occur include cyberattacks and network intrusion. The usage of open-source libraries and keeping up to date with new web application technologies is great for the overall functionality of a software but can bring new vulnerabilities with it. The system may be upgraded when not fully ready and receive a vulnerability somewhere in it, for example.

## Areas of Security

The areas of the security listed in the Vulnerability Assessment Process Flow Diagram that apply to the software application include:

* Input Validation- Due to the financial security needed for a financial company, Artemis Financial will need to validate the information of each client. Doing this ensures that the person with access to an account is that person.
* APIs- Using APIs in the software would ensure that data being accessed is allowed for the receiving user.
* Cryptography- Using cryptography would encrypt the code so that neither Artemis Financials nor it’s users’ data and information is leaked. Since Artemis Financial is also assumed to be international, this would be highly recommended.
* Code Quality- Having good quality code will maximize the security of the software application. This is where restrictions, accesses, regulations, and other things can be set. Ensuring that the quality of the code meets industry standards helps strengthen the foundation of the application.
* Code Error- Having code error implementation into the code allows you to diagnose the problem that the API or system is having and fix it more quickly than before. This is crucial, as a financial corporation that serves people every second of the day cannot afford many errors.

## Manual Review

First, I noticed that there is no authentication scheme or input validation. This is necessary to ensure protection with authentication of users and their information. Next, I realized that there was no error handling, which could lead to lack of noticing errors in the system. Then, I seen that the request parameters are not valid and need to be moved to the header or body, which needs to be validated to securely send data. Lastly, the service does not use HTTPS, which needs to be fixed to run smoothly and properly on the internet, as well as for communications.

## Static Testing

When running static testing, there were many vulnerabilities found in the code:

* Bouncy Castle:
  + There were many CVE issues within the Bouncy Castle integrations. To resolve this, the system recommended updating Bouncy Castle. Examples include:
    - CVE-2016-1000352
    - CVE-2016-1000341
    - CVE-2016-1000344
* FasterXML Jackson Databind
  + Another dependency that had several vulnerabilities was FasterXML Jackson Databind. To resolve this, updating the integration was recommended. Examples include:
    - CVE-2021-46877
    - CVE-2020-36518
    - CVE-2020-25649
* Apache Log 4j API
  + The Apache Log 4j API had vulnerabilities within it. Upgrading was recommended. Examples include:
    - CVE-2021-44832
    - CVE-2020-9488
    - CVE-2022-41854
* SnakeYAML
  + SnakeYAML had eight vulnerabilities within it. Upgrading was recommended. Examples include:
    - CVE-2022-1471
    - CVE-2022-41854
    - CVE-2022-25857
* Apache Tomcat
  + Lastly, Apache Tomcat had twenty vulnerabilities. Upgrading was recommended again. Examples include:
    - CVE-2023-46589
    - CVE-2020-13935
    - CVE-2021-25122

## Mitigation Plan

Firstly, all dependencies need to be upgraded. There were a lot of vulnerabilities found when doing the static testing across many of the dependencies. Updating should resolve this, and if not then replacing it may be necessary. Next, authentication implementations must be done using input validation techniques. There is a lack of security in the current state regarding them. Then, HTTPS protocols need to be created to run smoothly and make secure communications. Next, ensure that all APIs get the attention that they require, as these will be crucial in safe data transfers. Lastly, after these issues have been fixed, repeat the cycle again and see if vulnerabilities in the dependencies and APIs are improved after updating and if input validations and HTTPS protocols are valid after changes.