****

# 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** | **NOV 12 2023** | **Seth Barr** |  |

## 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

Seth Barr

## Interpreting Client Needs

As a consulting company, Artemis Financial wants to implement more reliable software security due to having a web-based software application. Due to the nature of the company, security is a high priority for their consumer sensitive information. This information includes savings, retirement, investments, and insurance. This suggests security from outsider threats and ensures secure communication techniques between customers and coworkers. There is no interest in international transactions but that is a common threat to every company and should not be taken lightly. Due to the customer information at hand, there are specific standard minimums that security must achieve so the government deems it protected. This is to avoid external threats from outside the company trying to sell consumer data and identity theft from inside the company. The modernization requirements should also be considered, including the role of open-source libraries and evolving web application technologies.

## Areas of Security

Based on the Artemis Financial's web application, the following areas of security are relevant:

* **APIs:** The application should have authentication programs to ensure that only authorized users can access secure data.
* **Input Validation:** Proper input validation should always be implemented to prevent any vulnerabilities such as SQL injection.
* **Cryptography:** Sensitive data like the subject listed above, should be highly encrypted during transit and storage.
* **Client / Server:** The application should utilize secure communication to protect sensitive data between clients and servers.
* **Code Error / Code Quality:** As all code should stand, all programs should have a functioning system to stop any errors that accumulate to stop any threats from breaking in.
* **Encapsulation:** Due to the high number of clientele and subject matters, encapsulation is important to keep data secure and separated based off category and importance.

## Manual Review

The following vulnerabilities were identified:

* Cross-Site Scripting: There are potential attackers to inject malicious malware into the webpage code.

## Static Testing

By running the code through a Maven test, the following vulnerabilities were found:

* Using Apache software opens the ability to request and obtain information that should not be obtainable. In older versions of the spring configuration, a malicious user can artificially increase their privilege. This allows users to gain access where they previously were not allowed. This attack allows the ability to change shortcuts so functions do more than they should. There are some update releases that limit the damage that a someone can inflict on the system.

## Mitigation Plan

To mitigate any security vulnerabilities:

* Implement proper encoding in the file to prevent script injection. This can be achieved using encapsulation that handles encoding automatically.
* Modify the file to use prepared statements instead of inputting user input directly into SQL queries.
* Update as needed to prevent any failure of code.