****

# 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** | **01/25/2024** | **Thomas Martin** |  |

## 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

Thomas Martin

## Interpreting Client Needs

Artemis Financials, a consulting firm specializing in personalized financial planning, has developed a highly effective web-based software application with up-to-date software security measures. To further enhance its security posture, Artemis Financials has engaged Global Rain to conduct a comprehensive vulnerability assessment aimed at identifying and addressing potential security vulnerabilities in their software application. Artemis Financials is dedicated to prioritizing the importance of secure communications. A clear understanding is essential whether the company engages in international transactions and must safeguard its proprietary information. Compliance with any relevant data protection laws or regulations is paramount. The vulnerability assessment comprises identifying external threats with the web-based software application as well as considering modernization. Regularly accessing and updating open-source dependencies and staying current with web technologies ensures better performance, security, and user experience. Artemis Financials’ mission spans a range of critical dimensions, including safeguarding data confidentiality, fostering user trust, ensuring regulatory compliance, mitigating threats, maintaining data integrity, implementing robust authentication mechanisms, ensuring business continuity, and fortifying defenses against potential reputational damage.Top of Form

## Areas of Security

In the process of conducting the vulnerability assessment for Artemis Financial’s web-based software application, four areas were identified that would apply to Artemis Financial’s area of security.

Input Validation: Proper input validation prevents injection attacks while ensuring industry standard input sanitation protocols lower security vulnerabilities.

API: Application Programming Interfaces play a crucial role in modern web applications, enabling communication and data exchange between different components, services, or systems. Secure communication prevents eavesdropping and data tampering.

Cryptography: Encryption is essential for safeguarding customer’s financial information. Artemis Financial must protect client information focusing on encryption in transit and at rest.

Client / Server: Artemis Financial must ensure secure communication between clients and servers. Client/Server architecture in the security assessment of Artemis Financials can ensure a broad scope of the application's security, covering both the user interface and the backend processes.

## Manual Review

To conduct a manel review of the vulnerabilities in the Project One Code Base the Vulnerability Assessment Process Flow Diagram was used to identify vulnerabilities. In reviewing the file

GreetingController.java, it contains a /GET command, ensure proper input validation for the "name" string. The pom.xlm file does not directly handle data protection but includes dependencies, such as Bouncy Castle. The review of Project One Code Base lacked the addition of an API resulting in loss of data integrity. No cryptography was reviewed, files did not contain any type of data encryption. Code Quality was good while examining file DocData.java contusing no error handling.

## Static Testing

Dependency:

hibernate-validator-6.0.18.Final.jar, flaw in input validation https://ossindex.sonatype.org/component/pkg:maven/org.

jackson-databind-2.10.2.jar, upgrade to versions 2.13.2.1 and 2.12.6.1

pkg:maven/com.fasterxml.jackson.core/jackson-databind@2.10.2

logback-core-1.2.3.jar, upgrade to version 1.2.13

pkg:maven/ch.qos.logback/logback-core@1.2.3

snakeyaml-1.25.jar, constructor class does not restrict input validation

pkg:maven/org.yaml/snakeyaml@1.25

spring-boot-2.2.4.RELEASE.jar, upgrade to newest verison

pkg:maven/org.springframework.boot/spring-boot@2.2.4.RELEASE

spring-web-5.2.3.RELEASE.jar, , remote code execution issue

pkg:maven/org.springframework/spring-web@5.2.3.RELEASE

spring-webmvc-5.2.3.RELEASE.jar, malicious input issue

pkg:maven/org.springframework/spring-webmvc@5.2.3.RELEASE

## Mitigation Plan

Global Rain’s mitigation plan is based on the results from the manual review and static testing report. A comprehensive mitigation plan for Artemis Financial's software application would begin by addressing the lack of proper input validation for the "name" parameter. Preventing malicious data entry for the username and password issue identified by vulnerability hibernate-validator-6.0.18.Final.jar. The next issue would address and prevent the exploitation of vulnerability identified in vulnerability jackson-databind-2.10.2.jar by upgrading the Apache Server to the newer version of 2.13.2. This update will fix some of the vulnerabilities in the previous version of the Apache Server. Next issue would address the code review and error handling. Integrate secure industry standard coding practices and error handling. The next issue is to address the issues of certificate validation, the code should be properly sanitized to allow for proper validation and verification of all of the digital certificates for the application and the web server, helping to prevent the exploitation of vulnerability spring-webmvc-5.2.3.RELEASE.jar and snakeyaml-1.25.jar. Address and configure the Spring Boot application to use HTTPS for secure communication preventing the potential exposure of sensitive information in the code from the lack of explicit enforcement of HTTPS. The migration plan should include detailed steps, responsible parties, timelines, and verification mechanisms for each identified vulnerability.