****

# 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** | **09/11/23** | **Darren Nason** | **Vulnerability Assessment** |

## 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

Darren Nason

## Interpreting Client Needs

The nature of Artemis Financial application and what their company is involved with such as sensitive information relating to customers savings, retirement, brokerage, and insurance accounts. This is of concern on a security part focused on data authorization and focus on not allowing any data breaches, leaks, or unauthorized access. The value of secure communications for Artemis are highly valuable as it pertains to data protection as they store customers sensitive information. This also can be of legal threat if Artemis doesn’t comply with regulatory requirements as secure communications could be mandated and these can be strict when dealing with international transactions as there are international data protection laws. For governmental restrictions, this depends on the location as data communication regulations to follow in relations to data encryption and residency requirements. The immediate threats concerned about are as follows: cyberattacks such as phishing, ransomware, or DDoS, data theft by gaining unauthorized access, malware that infects the systems with malicious software, lastly insider threats by employees who don’t take their jobs seriously and neglect job responsibility. Now for emerging threats the biggest of concern are zero-day vulnerabilities or advanced persistent threats known as APTs. When it comes to working with open-source libraries the need to make sure they are up to date, free from known vulnerabilities, and is essential to have proper monitoring and version control as well. Also, when implementing evolving web app technologies this calls for implementing the latest security headers, encryption protocols, and authentication mechanisms.

## Areas of Security

* Authentication and Authorization
* This will ensure that the right user has access to their data and not allow for data breaches.
* Cryptography
* This is extremely important considering the confidential data that Artemis is handling integrity, and availability is crucial for the customer that their data is not lost or stolen such.
* Network Security
* Having secure network infrastructure and communicating channels helps in preventing DDoS attacks that could disrupt services.
* Secure Logic and Code
* Essential security measure to prevent against attacks such as SQL injection and XSS which is crucial when trying to prevent exploitation.
* APIs
* Need proper security headers, input validation, rate limiting as these 3 key characteristics play a huge role in securing our APIs in our source code.

## Manual Review

* Input Validation and Sanitization
* SQL injection
* Cross-site-scripting (XSS)
* Authentication and Authorization
* Unauthorized Access
* Account hijacking
* Database Security
* Credential exposure
* Data leaks
* HTTPS Encryption
* Data interception
* Man – in – the – middle attack
* Error Handling
* Brute force attacks
* Rate Limiting and Throttling
* Denial of service attacks
* Data Validation Inside of Data Classes
* Incorrect behavior due to badly stored data

## Static Testing

A screenshot of a computer

Description automatically generated

Vulnerabilities Found below:

* Spring-boot-starter-web-2.2.4.release.jar:
* Critical severity
* 3 CVEs
* Spring-boot-2.2.4.release.jar:
* Critical severity
* 3 CVEs
* Snakeyml-1.25.jar:
* Critical severity
* 8 CVEs
* Spring-webmvc-5.2.3.release.jar:
* Critical severity
* 11 CVEs
* Spring-core-5.2.3.release.jar:
* Critical severity
* 11 CVEs
* Spring-web-5.2.3.release.jar:
* Critical severity
* 12 CVEs
* Tomcat-embed-websocket-9.0.30.jar:
* Critical severity
* 22 CVEs
* Jackson-databind-2.10.2.jar:
* High severity
* 6 CVEs
* Bcprov-jdk14on-1.46.jar:
* High severity
* 18 CVEs
* Logback-core-1.2.3.jar:
* Medium severity
* 1 CVE
* Hibernate-validator-6.0.18.final.jar:
* Medium severity
* 1 CVE
* Log4j-api-2.12.1.jar:
* Low severity
* 1 CVE

## Mitigation Plan

* Data Encryption
* Configure server to support HTTPS and obtain a valid SSL/TLS certificate.
* Error Handling
* Implement a centralized error handling mechanism to handle consistent error responses such as 404 errors, etc.
* Rate Limiting and Throttling
* Use libraries or frameworks that provide rate limiting features.
* Data Validation and Data Classes
* Implement data validation checks within the classes themselves to enforce data integrity.
* Database Security
* Implement secure credential storage and appropriate access controls to the database.
* Authentication and Authorization
* Use a robust authentication framework such as spring security for handling authentication and authorization
* Input validation and sanitization
* Use strict input validation along with parameterized queries for database queries.