****

# 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** | **11/12/2023** | **Ben Miller** |  |

## 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

Ben Miller

## Interpreting Client Needs

Artemis Financials needs to strongly consider secure communications to protect the privacy of their customers. Since they are not specifically limited to the US, any cryptography used must be checked to ensure it follows government regulations. Their program code needs to be reviewed for vulnerabilities to help them implement the most current and advanced security practices.

## Areas of Security

The areas of security are Input Validation, API’s, Cryptography, and Code Quality.

## Manual Review

CRUDController.java: Need input validation and need to implement PreparedStatement or other parameterization process to prevent SQL injection.

myDateTime.java: Create a copy of the date/time info and do processing on the copy of the data instead of the original (vulnerable to time-of-check, time-of-use inconsistency).

DocData.java: because the username and password are both hardcoded, every time the “Test” database is accessed through DocData.java the user has full access to the information. There should be a user access plan in place to determine which users have access to what information and capability. The access control should be done with a Java EE access control filter or similar secure implementation. Should have input validation for the read\_document() method parameters.

GreetingController.java: Need to implement input validation and need PreparedStatement or other parameterization process to prevent SQL injection.

customer.java: need input validation for deposit() method parameter. Account balance should be private. Should change account balance to float type- int type does not allow for anything except whole dollar amounts and can cause unintended issues with the program.

CRUD.java: need input validations for CRUD() method parameters.

## Static Testing

A screenshot of a computer

Description automatically generated

## Mitigation Plan

Aside from the improvements identified in the Manual Review, the following mitigation steps are needed to improve application security:

1. Upgrade Bouncy Castle JCE Provider Version to version greater than 1.55. The current outdated version is susceptible to high severity vulnerabilities.
2. Update Spring Boot version- the current version is unsupported, outdated, and vulnerable to denial-of-service attacks. Update to version 3.0.6+ or 2.7.11+.
3. Update cpe:/:apache:log4j version to version 2.12.4. The current version does not protect against attacker controlled LDAP and other JNDI related endpoints.
4. Update cpe:/:snakeyaml\_project:snakeyaml to version 2.0+. The current version is vulnerable to denial-of-service and remote code execution attacks.
5. Update cpe:/:fasterxml:jackson-databind to version 2.13.4.1+. The current version is vulnerable to denial-of-service attacks and other issues.
6. Update cpe:/:apache:tomcat to version 9.0.81+. The current version is vulnerable to a number of denial-of-service attacks and a few other issues.
7. Update cpe:/:pivotal\_software:spring\_framework to version 5.3.7+. The current version is vulnerable to remote code execution attacks and other issues.