****

# 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** | **2023-05-16** | **Griffin Hood** |  |

## 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

Griffin Hood

## Interpreting Client Needs

Secure communications are of the utmost importance to Artemis Financial. They deal with lots of financial information which is often sensitive and client confidentiality must be retained. As of now, Artemis Financial can make international transactions so international standards must be considered as well. In the United States, there are no governmental restrictions on secure data communications. Some relevant external threats include unencrypted communications between clients and servers, SQL injection, and un-sanitized input. Modernization requirements that must be considered are validating the security of open-source software that may be used such as web frameworks. Additionally, keeping up to date with other software dependencies as well as infrastructure is also a must.

## Areas of Security

The areas of the VAPF diagram that apply to Artemis Financials’ project are APIs, Cryptography, and Client / Server. APIs are always an area of concern because they are an access point to an application from external sources. Proper authentication must be put in place. Cryptography is another big area of security. SSL is a great way to encrypt data on the move and can help to ensure that communications are not intercepted along the way. With the rise in popularity of REST APIs and Single-Page Apps, the client/server development model is being used more frequently. Ensuring that the API knows of all of the clients that will be accessing it is also important.

## Manual Review

* DocData.java line 27 – Database credentials are in the source code.
* DocData.java line 30 – Stack trace is printed to the console (I think) instead of in a secure log file.
* Customer.java – Account number is unencrypted.
* GreetingController.java line 17 – Input is not sanitized/escaped.
* MyDateTime.java – Class members are not hidden behind private scope

## Static Testing

* bcprov-jdk15on-1.46.jar
  + CVE-2013-1624
  + TLS implementation does not consider timing side-channel attacks
* spring-boot-2.2.4.RELEASE.jar
  + CVE-2022-27772
  + spring-boot versions prior to version v2.2.11.RELEASE was vulnerable to temporary directory hijacking
* logback-core-1.2.3.jar
  + CVE-2021-42550
  + In logback version 1.2.7 and prior versions, an attacker with the required privileges to edit configurations files could craft a malicious configuration
* log4j-api-2.12.1.jar
  + CVE-2020-9488
  + Improper validation of certificate with host mismatch in Apache Log4j SMTP appender
* snakeyaml-1.25.jar
  + CVE-2017-18640
  + The Alias feature in SnakeYAML before 1.26 allows entity expansion during a load operation.
* jackson-databind-2.10.2.jar
  + CVE-2020-256489
  + A flaw was found in FasterXML Jackson Databind, where it did not have entity expansion secured properly
* tomcat-embed-core-9.0.30.jar
  + CVE-2019-17569
  + The refactoring present in Apache Tomcat 9.0.28 to 9.0.30, 8.5.48 to 8.5.50 and 7.0.98 to 7.0.99 introduced a regression
* hibernate-validator-6.0.18.Final.jar
  + CVE-2020-10693
  + A bug in the message interpolation processor enables invalid EL expressions to be evaluated as if they were valid
* spring-web-5.2.3.RELEASE.jar
  + CVE-2016-1000027
  + Pivotal Spring Framework through 5.3.16 suffers from a potential remote code execution (RCE) issue if used for Java deserialization of untrusted data
* spring-beans-5.2.3.RELEASE.jar
  + CVE-2022-22965
  + A Spring MVC or Spring WebFlux application running on JDK 9+ may be vulnerable to remote code execution (RCE) via data binding
* spring-webmvc-5.2.3.RELEASE.jar
  + CVE-2021-22060
  + In Spring Framework versions 5.3.0 - 5.3.13, 5.2.0 - 5.2.18, and older unsupported versions, it is possible for a user to provide malicious input to cause the insertion of additional log entries
* spring-context-5.2.3.RELEASE.jar
  + CVE-2022-22968
  + In Spring Framework versions 5.3.0 - 5.3.18, 5.2.0 - 5.2.20, and older unsupported versions, the patterns for disallowedFields on a DataBinder are case sensitive which means a field is not effectively protected
* spring-expression-5.2.3.RELEASE.jar
  + CVE-2022-22950
  + Spring Framework versions 5.3.0 - 5.3.16 and older unsupported versions, it is possible for a user to provide a specially crafted SpEL expression that may cause a denial of service condition.

## Mitigation Plan

Most of these vulnerabilities can be mitigated by upgrading to a newer version of the package. Some vulnerabilities, such as the Alias feature in SnakeYAML, can be averted by not using that specific feature of the package. Another example is Jackson Databind. The vulnerability here has to do with the FasterXML feature and not the entire dependency itself. Avoiding FasterXML will be sufficient. Another interesting one is Apache Tomcat. There was a vulnerability that was introduced in newer versions as a result of refactoring. This vulnerability only applies if Tomcat is being used behind a reverse proxy.