****

# 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** | **23 March 2024** | **Justin Angle** | **Initial** |

## 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

Justin Angle

## Interpreting Client Needs

Artemis Financial is looking to modernize their operations that they use. They handle all kinds of financial plans to include savings, retirements, investments, and insurance. Secure communication is most likely extremely valued as with anything dealing with money, this is someone’s livelihood and should be handled with utmost security. This company most likely has international transactions as there was nothing stating otherwise and most banks have their toes dipped in the water internationally. Depending on where Artemis Financial is doing business will depend on what laws they have to follow, in U.S. we have certain protections on our personal information that needs to be followed like the privacy act of 1974. Some external threats that might be out there are groups looking for PII like SSNs and full names to be able to steal someone’s identity. These groups may also want to steal some money from people by hacking the bank. Some modernization requirements that Artemis Financial should consider is using end-to-end encryption, using 2FA, and encrypting all information stored.

## Areas of Security

* APIs – Having an API would be necessary because it would be running both internally and externally allowing for data that is accessed to be acceptable.
* Input Validation – The company would need to have input validation to validate the owner of the information providing protection for users.
* Code error – There would need to be error handling in the company’s website. This way there would not be any security vulnerabilities if an error happened when accessing the website.

## Manual Review

Reviewing the code and applying the vulnerability assessment, I looked at the pom.xml file and saw that the program is not using the latest version of Spring Framework, which could be a security vulnerability. I also took a look at the greeting controller and saw there was no input validation, which should be added. It also didn’t look like there was any error handling in the code which could be cause for concern. For the API, I noticed that there was a possible security breach that could expose the user input as it was not written in the POST method.

## Static Testing

* bcprov-jdk15on-1.46.jar
* The TLS implementation in the Bouncy Castle Java library before 1.48 and C# library before 1.8 does not properly consider timing side-channel attacks on a noncompliant MAC check operation during the processing of malformed CBC padding, which allows remote attackers to conduct distinguishing attacks and plaintext-recovery attacks via statistical analysis of timing data for crafted packets, a related issue to CVE-2013-0169.
* Recommendation is to update to the newest software version.
* spring-boot-2.2.4.RELEASE.jar
* spring-boot versions prior to version v2.2.11.RELEASE was vulnerable to temporary directory hijacking. This vulnerability impacted the org.springframework.boot.web.server.AbstractConfigurableWebServerFactory.createTempDir method. NOTE: This vulnerability only affects products and/or versions that are no longer supported by the maintainer
* Recommendation is to update to the newest software version.
* logback-core-1.2.3.jar
* In logback version 1.2.7 and prior versions, an attacker with the required privileges to edit configurations files could craft a malicious configuration allowing to execute arbitrary code loaded from LDAP servers.
* Recommendation is to update to the newest software version.
* jackson-databind-2.10.2.jar
* A flaw was found in FasterXML Jackson Databind, where it did not have entity expansion secured properly. This flaw allows vulnerability to XML external entity (XXE) attacks. The highest threat from this vulnerability is data integrity.
* Recommendation is to update to the newest software version.
* tomcat-embed-core-9.0.30.jar
* 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. The result of the regression was that invalid Transfer-Encoding headers were incorrectly processed leading to a possibility of HTTP Request Smuggling if Tomcat was located behind a reverse proxy that incorrectly handled the invalid Transfer-Encoding header in a particular manner. Such a reverse proxy is considered unlikely.
* Recommendation is to update to the newest software version.
* hibernate-validator-6.0.18.Final.jar
* A flaw was found in Hibernate Validator version 6.1.2.Final. A bug in the message interpolation processor enables invalid EL expressions to be evaluated as if they were valid. This flaw allows attackers to bypass input sanitation (escaping, stripping) controls that developers may have put in place when handling user-controlled data in error messages.
* Recommendation is to update to the newest software version.
* spring-web-5.2.3.RELEASE.jar
* Pivotal Spring Framework through 5.3.16 suffers from a potential remote code execution (RCE) issue if used for Java deserialization of untrusted data. Depending on how the library is implemented within a product, this issue may or not occur, and authentication may be required. NOTE: the vendor's position is that untrusted data is not an intended use case. The product's behavior will not be changed because some users rely on deserialization of trusted data.
* Recommendation is to not use untrusted data.

## Mitigation Plan

The best remedy for the security flaws and vulnerabilities found would be to go through all applications the program uses and update to the latest version to fix most if not all vulnerabilities found. Artemis Financial should also add error handling in the code and add input validation. Artemis Financial should also ensure they do not use untrusted data.