****

# 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** | **May 20, 2023** | **Thomas Bartlett** | **Initial Findings** |

## 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 Bartlett

## Interpreting Client Needs

Artemis Financial is a consulting firm that provides tailored financial plans in the form of savings, retirement, investments, and insurance for their clients. Because they collect, store, and use sensitive information for their clients and employees, the value of secure communications is very high. At this time, it is not known if Artemis Financial makes any international transactions, but it would be best to move forward with the notion that if they do not currently that they could possibly in the future. There are governmental policies that guide how companies need to approach secure communications that will need to be considered for this project. The Federal Trade Commission and other agencies that watch over financial companies like Artemis Financial are made to enforce the financial privacy provisions of the Gramm-Leach-Bliley Act that was put in place November 12, 1999. This act mandates that Artemis Financial will need to explain their policies regarding information-sharing and how they safeguard sensitive data. The largest threat to Artemis Financial is cyber-attacks. This may be in the form of hacking directly into their databases or could be phishing scams conducted on employees to gain access to sensitive information. As far as modernization requirements are concerned, Artemis Financial may want to consider having a team of IT professionals to perform routine security checks, update any programs when new versions of open sourced libraries are released and actively watch for breaches in security.

## Areas of Security

After reviewing the needs of Artemis Financial, there are several areas of security from the Vulnerability Assessment Process Flow Diagram that would need to be addressed. These areas are as follows:

* Input Validation: Any inputs need to be properly validated so as to not allow unwanted access to sensitive information.
* APIs: This area would control the information that may be accessed from outside sources.
* Cryptography: This adds that extra level of security to make sure when information is flowing and being accessed, that the information is protected.
* Code Quality: Poor code quality could lead to not only poor functionality but also potential security concerns.

## Manual Review

There are multiple concerns as of right now with looking at the current code. As of right now, there is no authentication system present to be able to verify user credentials when accessing information. There is also no signs of input validation within the code. I tried to review the cryptography within the code, but I was not able to find any present at this time. Although the code that is there is written properly and may meet quality standards, when looking at the code as a whole the quality is lacking. The previous concerns mentioned would definitely need to be addressed to bring up the quality of the code.

## Static Testing

|  |  |
| --- | --- |
| Dependency and Description | Vulnerabilities |
| bcprov-jdk15on-1.46.jar –  The Bouncy Castle Crypto package is a Java implementation of cryptographic algorithms. This jar contains JCE provider and lightweight API for the Bouncy Castle Cryptography APIs for JDK 1.5 to JDK 1.7 | CVE-2016-1000338, CVE-2016-1000342, CVE-2016-1000343, CVE-2016-1000344, CVE-2016-1000352, CVE-2016-1000341, CVE-2016-1000345, CVE-2017-13098, CVE-2020-15522, CVE-2020-0187 (OSSINDEX), CVE-2016-1000339, CVE-2020-26939 (OSSINDEX), CVE-2015-7940, CVE-2018-5382, CVE-2013-1624, CVE-2016-1000346, CVE-2015-6644 (OSSINDEX) |
| hibernate-validator-6.0.18.Final.jar –  Hibernate's Bean Validation (JSR-380) reference implementation. | CVE-2020-10693 |
| jackson-databind-2.10.2.jar-  General data-binding functionality for Jackson: works on core streaming API | CVE-2020-25649, CVE-2020-36518, CVE-2021-46877, CVE-2022-42003, CVE-2022-42004 |
| log4j-api-2.12.1.jar –  The Apache Log4j API | CVE-2020-9488 |
| logback-core-1.2.3.jar –  logback-core module | CVE-2021-42550 |
| snakeyaml-1.25.jar –  YAML 1.1 parser and emitter for Java | CVE-2022-1471, CVE-2017-18640, CVE-2022-25857, CVE-2022-38749, CVE-2022-38751, CVE-2022-38752, CVE-2022-41854, CVE-2022-38750 |
| spring-boot-2.2.4.RELEASE.jar –  Spring Boot  spring-boot-starter-web-2.2.4.RELEASE.jar –  Starter for building web, including RESTful, applications using Spring MVC. Uses Tomcat as the default embedded container | CVE-2022-27772 |
| spring-core-5.2.3.RELEASE.jar –  Spring Core  spring-webmvc-5.2.3.RELEASE.jar –  Spring Web MVC | CVE-2022-22965, CVE-2021-22118, CVE-2020-5421, CVE-2022-22950, CVE-2022-22971, CVE-2023-20861, CVE-2023-20863, CVE-2022-22968, CVE-2022-22970, CVE-2021-22060, CVE-2021-22096 |
| spring-web-5.2.3.RELEASE.jar –  Spring Web | CVE-2016-1000027, CVE-2022-22965, CVE-2021-22118, CVE-2020-5421, CVE-2022-22950, CVE-2022-22971, CVE-2023-20861, CVE-2023-20863, CVE-2022-22968, CVE-2022-22970, CVE-2021-22060, CVE-2021-22096 |
| tomcat-embed-core-9.0.30.jar –  Core Tomcat implementation | CVE-2020-1938, CVE-2020-11996, CVE-2020-13934, CVE-2020-13935, CVE-2020-17527, CVE-2021-25122, CVE-2021-41079, CVE-2022-29885, CVE-2022-42252, CVE-2020-9484, CVE-2021-25329, CVE-2021-30640, CVE-2022-34305, CVE-2021-24122, CVE-2021-33037, CVE-2019-17569, CVE-2020-1935, CVE-2020-13943, CVE-2023-28708, CVE-2021-43980 |
| tomcat-embed-websocket-9.0.30.jar –  Core Tomcat implementation | CVE-2020-1938, CVE-2020-8022, CVE-2020-11996, CVE-2020-13934, CVE-2020-13935, CVE-2020-17527, CVE-2021-25122, CVE-2021-41079, CVE-2022-29885, CVE-2022-42252, CVE-2020-9484, CVE-2021-25329, CVE-2021-30640, CVE-2022-34305, CVE-2021-24122, CVE-2021-33037, CVE-2019-17569, CVE-2020-1935, CVE-2020-13943, CVE-2023-28708, CVE-2021-43980 |

For the dependencies and known vulnerabilities the best solution would to be sure that the current versions of the open-source libraries are being used. Most of the vulnerabilities have been fixed in updated versions.

## Mitigation Plan

The biggest area that needs addressed is making sure when working with tools and open-sourced libraries that the most current versions/updates are being used. As shown in the static testing, not working with the most current version opens the program up to a myriad of vulnerabilities. The other area to focus on is addressing the areas of security as described above. Focus should be on making the program more secure through following best practices when it comes to input validation and authentication, cryptography of sensitive data, and overall code quality.

Reference:

*How To Comply with the Privacy of Consumer Financial Information Rule of the Gramm-Leach-Bliley Act*. (2002, July 2). Federal Trade Commission. https://www.ftc.gov/business-guidance/resources/how-comply-privacy-consumer-financial-information-rule-gramm-leach-bliley-act