****

# 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** | **July 11, 2023** | **Zachary Nicholas** | **Add the requested information** |

## 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

Zachary Nicholas

## Interpreting Client Needs

With Artemis Financial being a financial institution that makes financial plans for their customers, they are tasked with holding sensitive and Personal Identifiable Information (PII) with this being know security is one of their main concerns as well as any other government regulations that they are required to follow. Failure in doing this can leave a hole that others or any potential bad actors could use in order to either hurt Artemis Financial or could hurt one of their clients. Given what we know of different attack vectors things like the API being unsecure could cause damage, Another thing that could help to improve the overall security of a web based system is to use a type of two factor authentication whether it be in a authenticator app or just send over text would still add a layer of security to prevent any attempts to get into other accounts.

## Areas of Security

One of the main reasons to do a code review like this is to make sure that there are no areas in either the code we make or other code we are hired to look at that could present an open avenue for potential bad actors to use to either hurt the company that will use this code or the individuals that will have their sensitive information stored within it. The first area where we should look would be Input validation which would help to secure and validate any inputs made to the Api or any other related services, the second one I feel as important is our APIs since the web service we are looking at is running using the restful API which would need a way for secure communication, And the last one I feel is directly relevant to this would be code errors and how they can help to prevent multiple login attempts whether they be a brute force attack or just someone forgot their password or any other part of our code that could be looked over.

## Manual Review

Things that I found during this review are:

* Requests won’t be validated, which can lead to them being looked at by an unintended person.
* The data stored in our application can be found in other areas which with an unsecured system could result In someone getting access to all user’s names and passwords and potentially other PII relating to them

## Static Testing

During my review of the dependency-check I have identified the following:

* Bcprov-jdk15on-1.46jar which has 12 vulnerabilities. The vulnerability IDs I have found are “CVE-2016-1000352, CVE-2016-1000346, CVE-2016-1000345, CVE-2016-1000344, CVE-2016-1000343, CVE-2016-1000342, CVE-2016-1000341, CVE-2016-1000339, CVE-2016-1000338, CVE-2018-5382, CVE-2017-13098, and CVE-2013-1624.” It is suggested to update to the newest version available.
* Spring-boot-2.2.4.RELEASE.jar which has 2 vulnerabilities. The vulnerability codes I found are “CVE-2023-20883 and CVE-2022-27772.” The suggested action relating to this one is to update to the newest version available
* Hibernate-validator-6.0.18.Final.jar has 1 vulnerability labeled as “CVE-2020-10693”. To fix this it is suggested to update to the newest version available.
* Jackson-databind-2.10.2.jar which has 6 vulnerabilities them being labeled “CVE-2023-35116, CVE-2021-46877, CVE-2022-42004, CVE-2022-42003, CVE-2020-36518, CVE-2020-25649.” To fix these it is suggested to update to the newest version available.
* Log4j-api-2.12.1.jar which has 5 vulnerabilities those being “CVE-2021-44832, CVE-2021-45105, CVE-202145046, CVE-2021-44228, CVE-2020-9488.” To fix these it is suggested to update to the newest version available.
* Snakeyami-1.25.jar which has 8 vulnerabilities those being “CVE-2022-1471, CVE-2022-41854, CVE-2022-38752, CVE-2022-38751, CVE-2022-38750, CVE-2022-38749, CVE-2022-25857, CVE-2017-18640.” To fix these it is suggested to update to the newest version available.
* Tomcat-embed-core-9.0.30.jar which has 20 vulnerabilities those being “CVE-2023-28708, CVE-2022-42252, CVE-2021-43980, CVE-2022-34305, CVE-2022-29885, CVE-2021-41079, CVE-2021-33037, CVE-2021-30640, CVE-2021-25329, CVE-2021-25122, CVE-2021-24122, CVE-2020-17527, CVE-2020-13943, CVE-2020-13935, CVE-2020-13934, CVE-2020-8022, CVE-2020-11996, CVE-2020-9484, CVE-2020-1938, CVE-2020-1935.” To fix these it is suggested to update to the newest version available.

## Mitigation Plan

Things that I would suggest would be to install all the newest versions of all dependencies from the list that I have provided, I also recommend that the system finds a way to patch the hole that is currently in it that allows others to see unencrypted data transferred to and from the API and the system itself, The final thing I would suggest is to implement a way for two-factor authentication of all of the services in order for the data sent to users and back from them reach the intended person or user and no one else that may be looking at any of the data in the program.