****

# 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** | **09/12/2023** | **Eric Breznen** | **Added assessment report** |

## 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

Eric Breznen

## Interpreting Client Needs

The primary target of attackers is Artemis’ clients’ financial information. A smart attacker will attempt to quietly sneak into the system so that they can steal information over a longer period of time. A man in the middle attack could also be used to catch the information in transit so secure communications are a must whether domestic or international. SQL Injection attacks could be used to alter the nature of transactions and where money would be sent to. With investments a part of the company’s repertoire and the concept of international stock purchases, each nation will have their own guidelines and expectations as far as how money is transferred that must be considered while the company interacts with their stock exchanges. Utilizing open-sourced libraries can be a helpful resource for aspects of the software, and many eyes and hands on the library can be both good and bad. The good side is that many people have had the opportunity to catch security vulnerabilities, however, vulnerabilities can also be introduced by malicious programmers that could be inadvertently added to the company’s software meaning that all libraries must be scrutinized, and vulnerabilities caught and patched.

## Areas of Security

The most important areas of security that must be considered are input validation, each attempt to input information into a request must be valid to prevent attack strings, or other harmful information. Cryptography is highly important to keep Artemis’ client’s information secure from anybody listening in. Using a RESTful API, the interaction with this interface must be secure. The Client/Server communications must be dependable to ensure that actions that are being taken are exactly as the client requests. In the event of invalid input, the program must be able to handle the error without opening the system to attack. If an attacker is able to get in, good encapsulation practice will help prevent the attack from spreading, or at least delay it to mitigate damage until the breach can be patched.

## Manual Review

Manual review revealed that there was little in the way of input validation, however this is an incomplete program and there is room to add it. The application uses a Spring framework which gives potential for client/server communications, however risks service going down in the event the Tomcat server does. The application as laid out also supports strong encapsulation practices by giving each object it’s own class.

## Static Testing

* bcprov-jdk15on-1.46.jar
  + The Bouncy Castle JCE Provider version 1.55 and earlier does not fully validate ASN. 1 encoding which makes it possible to inject elements into the signature and still have it validate allowing invisible data into a signed structure - CVE-2016-1000338
* hibernate-validator-6.0.18.Final.jar
  + A bug in the interpolation processor enables invalid EL expressions to be evaluated as if they were valid. This allows for attackers to bypass input sanitation controls that are put in place. - CVE-2020-10693
* jackson-databind-2.10.2.jar
  + Flaw discovered in FasterXML Jackson Databind where the entity expansion was not secured properly. This vulnerability affects XXE attacks threatening data integrity. - CVE-2020-25649
* log4j-api-2.12.1.jar
  + improper validation of certificate with a host mismatch In Apache Log4j SMTP appender allowing for man-in-the-middle attacks which can leak log messages. - CVE-2020-9488
* logback-core-1.2.3.jar
  + logback version 1.2.7 and earlier allows for attackers with required privileges to edit configurations which can execute arbitrary code from LDAP servers - CVE-2021-42550
* snakeyaml-1.25.jar
  + The constructor class does not restrict types that can be instantiated which can lead to remote code execution. Recommend using the safeconsturctor when parsing untrusted content, recommend upgrade to version 2.0 and above. - CVE-2022-1471
* spring-boot-2.2.4.RELEASE.jar
  + Spring boot versions prior to 3.0.5 could cause applications deployed to Cloud Foundry to be vulnerable to security bypass. Recommend upgrading to 3.0.6 or higher - CVE-2023-20873
* spring-boot-starter-web-2.2.4.RELEASE.jar
  + Versions 3.0.5 could be vulnerable to security bypass when deployed to Cloud foundry. Recommend upgrading to 3.0.6 or above - CVE-2023-20873
* spring-core-5.2.3.RELEASE.jar
  + Spring MVC or WebFlux running on JDK9 or above could be vulnerable to remote code execution through databinding. CVE-2022-22965
* spring-web-5.2.3.RELEASE.jar
  + Untrusted data can cause remote code execution on pivotal Spring Framework through 5.3.16 depending on how the library is implemented within a product. CVE-2016-1000027
* spring-webmvc-5.2.3.RELEASE.jar
  + Spring MVC or WebFlux applications running on JDK9 or above may be remote code execution through databinding. This exploit is not applicable if the application is run as a Spring executable JAR file. - CVE-2022-22965
* tomcat-embed-core-9.0.30.jar
  + When using the Apache JServ protocol, Tomcat treats these connections with higher trust than a similar HTTP connection. Recommend disabling this connection unless necessary and upgrade to versions 7.0.100 or later. - CVE-2020-1938
* tomcat-embed-websocket-9.0.30.jar
  + When using the Apache JServ protocol, Tomcat treats these connections with higher trust than a similar HTTP connection. Recommend disabling this connection unless necessary and upgrade to versions 7.0.100 or later. - CVE-2020-1938

## Mitigation Plan

In order to mitigate the vulnerabilities in this program, it is essential to implement strong input validation techniques, whitelisting would be a good methodology to follow. It is also important to make sure that JDK, Tomcat, and Spring are all updated to the latest versions especially when not using an executable JAR file. Ensuring encryption is secure is also invaluable to the company to keep customers’ data safe.