****

# 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** | **9/11/2022** | **Matthew Stebbins** |  |

## 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

Matthew Stebbins

## Interpreting Client Needs

Customers go to Artemis for financial plans. With customers financial information communication between Artemis and their customers needs to be secure and government regulations will need to be followed when it comes to how the data is transmitted and stored. If not setup correctly their RESTful API could be vulnerable to data interception. HTTPS is a must for this type of traffic.

## Areas of Security

* Input validation – the RESTful API will accept user input; this input will have to be sanitized
* APIs – the web service includes a RESTful API which will need to communicate securely
* Code error – errors due to improper user input must be handled securely and without giving information to hackers
* Secure coding – to Remove loopholes in the service that may create vulnerabilities

## Manual Review

Need authentication and HTTPS

Data needs to be validated

## Static Testing

* Bouncy Castle version 1.46 has several vulnerabilities. Recommendation: update to at least 1.60
  + CVE-2013-1624
  + CVE-2015-6644
  + CVE-2015-7940
  + CVE-2016-1000338
  + CVE-2016-1000339
  + CVE-2016-1000341
  + CVE-2016-1000342
  + CVE-2016-1000343
  + CVE-2016-1000344
  + CVE-2016-1000345
  + CVE-2016-1000346
  + CVE-2016-1000352
  + CVE-2017-13098
  + CVE-2018-1000613
  + CVE-2018-5382
* FasterXML Jackson Databind 2.10.2 has one vulnerability. Recommendation: update to at least 2.10.5.1
  + CVE-2020-25649
* Apache Log4j API 2.12.1 has one vulnerability. Recommendation: update to at least 2.13.2
  + CVE-2020-9488
* SnakeYAML 1.25 has one vulnerability. Recommendation: update to at least 1.26
  + CVE-2017-18640
* Spring Core 5.2.3 has one vulnerability. Recommendation: update to at least 5.2.8
  + CVE-2020-5421
* Apache Tomcat 9.0.30 has several vulnerabilities. Recommendation: update to at least 9.0.40
  + CVE-2019-17569
  + CVE-2020-11996
  + CVE-2020-13934
  + CVE-2020-13935
  + CVE-2020-13943
  + CVE-2020-17527
  + CVE-2020-1935
  + CVE-2020-1938
  + CVE-2020-8022
  + CVE-2020-9484
  + CVE-2021-24122

## Mitigation Plan

Add authentication and switch to HTTPS

Implement data validation

Update the dependencies that create vulnerabilities in the system and remove the ones that don’t from future static tests.