****

# 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** | **24 Jan 2024** | **Tyler Barnes** |  |

## 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

Tyler Barnes

## Interpreting Client Needs

## Determine your client’s needs and potential threats and attacks associated with their application and software security requirements. Document your findings in your vulnerability assessment report. Consider the scenario information and the following questions regarding how companies protect against external threats:

* 1. What is the value of secure communications to the company?
  2. Does the company make any international transactions?
  3. Are there governmental restrictions about secure communications to consider?
  4. What external threats might be present now and in the immediate future?
  5. What are the modernization requirements that you must consider? For example:
     1. The role of open-source libraries
     2. Evolving web application technologies

Artemis Financial wants to modernize their operations of financial planning for their customers. Financial information is highly sensitive, communication between the company and their customers must be secure. The company must follow government regulations for financial transactions and communications, which will affect security requirements and policies. Artemis Finacial’s API’s are vulnerable to interception if the responses and requests are not secured. They must use HTTPS for all communications and send confidential information for all responses.

## Areas of Security

Refer to the Vulnerability Assessment Process Flow Diagram provided. Think about the functionality of the software application to identify which areas of security apply to Artemis Financial’s web application. Document your findings in your vulnerability assessment report and justify why each area is relevant to the software application.

* API’s – RESTful API will need to communicate securely for the web service
* Code Error – User input errors must be secure
* Input Validation - The input will need to be validated so RESTful API will accept the user input.
* Secure Coding

## Manual Review

Refer to the seven security areas outlined in the Vulnerability Assessment Process Flow Diagram. Use what you’ve learned in steps 1 and 2 to guide your manual review. Identify all vulnerabilities in the Project One Code Base, linked in Supporting Materials, by manually inspecting the code. Document your findings in your vulnerability assessment report. Be sure to include a description that identifies where the vulnerabilities are found (specific class file, if applicable).

* Service doesn’t use HTTPS
* Request parameters aren’t validated
* Authentication scheme isn’t present
* In the CRUDController class, business names are sent as request parameters

## Static Testing

## Integrate the dependency-check plug-in into Maven by following the instructions outlined in the Integrating the Maven Dependency-Check Plug-in tutorial provided in Supporting Materials. Run a dependency check on Artemis Financial’s software application to identify all security vulnerabilities in the code. Specifically, identify all vulnerabilities in the code base by analyzing results from running the code through a static test. Include these items from the dependency-check report in your vulnerability assessment report:

* 1. The names or vulnerability codes of the known vulnerabilities
  2. A brief description and recommended solutions that are found in the dependency-check report
  3. Attribution (if any) that documents how this vulnerability has been identified or how it was documented in the past.
* Bouncy castle has several vulnerabilities. Recommend updating .
  + - 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
    - CVE-2013-1624
* FasterXML Jackson Databind 2.10.2 Recommend updating .
  + - CVE-2023-35116
    - CVE-2021-46877
    - CVE-2022-42004
    - CVE-2022-42003
    - CVE-2020-36518
    - CVE-2020-25649
* Apache Log 4j API 2.12.1 has 5 vulnerabilities. Recommend updating.
  + - CVE-2021-44832
    - CVE-2021-45105
    - CVE-2021-45046
    - CVE-2021-44228
    - CVE-2020-9488
* SnakeYAML 1.25 has 8 vulnerabilities. Recommend updating.
  + - CVE-2022-1471
    - CVE-2022-41854
    - CVE-2022-38752
    - CVE-2022-38751
    - CVE-2022-38750
    - CVE-2022-38749
    - CVE-2022-25857
    - CVE-2017-18640
* Apache Tomcat 9.0.30 has 20 vulnerabilities. Recommend updating.
  + - [CVE-2023-46589](https://nvd.nist.gov/vuln/detail/CVE-2023-46589)
    - [CVE-2023-45648](https://nvd.nist.gov/vuln/detail/CVE-2023-45648)
    - [CVE-2023-42795](https://nvd.nist.gov/vuln/detail/CVE-2023-42795)
    - [CVE-2023-44487](https://nvd.nist.gov/vuln/detail/CVE-2023-44487)
    - [CVE-2023-41080](https://nvd.nist.gov/vuln/detail/CVE-2023-41080)
    - [CVE-2023-28708](https://nvd.nist.gov/vuln/detail/CVE-2023-28708)
    - [CVE-2022-42252](https://nvd.nist.gov/vuln/detail/CVE-2022-42252)
    - [CVE-2022-34305](https://nvd.nist.gov/vuln/detail/CVE-2022-34305)
    - [CVE-2021-43980](https://nvd.nist.gov/vuln/detail/CVE-2021-43980)
    - [CVE-2022-29885](https://nvd.nist.gov/vuln/detail/CVE-2022-29885)
    - [CVE-2021-41079](https://nvd.nist.gov/vuln/detail/CVE-2021-41079)
    - [CVE-2021-33037](https://nvd.nist.gov/vuln/detail/CVE-2021-33037)
    - [CVE-2021-30640](https://nvd.nist.gov/vuln/detail/CVE-2021-30640)
    - [CVE-2021-25329](https://nvd.nist.gov/vuln/detail/CVE-2021-25329)
    - [CVE-2021-25122](https://nvd.nist.gov/vuln/detail/CVE-2021-25122)
    - [CVE-2021-24122](https://nvd.nist.gov/vuln/detail/CVE-2021-24122)
    - [CVE-2020-17527](https://nvd.nist.gov/vuln/detail/CVE-2020-17527)
    - [CVE-2020-13943](https://nvd.nist.gov/vuln/detail/CVE-2020-13943)
    - [CVE-2020-13935](https://nvd.nist.gov/vuln/detail/CVE-2020-13935)
    - [CVE-2020-13934](https://nvd.nist.gov/vuln/detail/CVE-2020-13934)

## Mitigation Plan

Interpret the results from the manual review and static testing report. Identify steps to mitigate the identified security vulnerabilities by creating an action list that documents how to fix each vulnerability in your vulnerability assessment report.

* Use HTTPS protocol for communications
* Instead of URI move request parameters to headers or body
* Implement authentication scheme
* Update dependencies