****

# 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** | **21-Jan-2023** | **Luke Peters** |  |

## 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

Luke Peters

## Interpreting Client Needs

The client, Artemis Financial, is a consulting company that develops individualized financial plans for its customers. The financial plans include savings, retirement, investments, and insurance. The client wants to modernize their operations and use the most current and effective software security to protect their organization from external threats.

The value of secure communications to the company is high, as the financial plans include sensitive information such as savings and investments. It is important that this information is protected from unauthorized access or tampering.

It is not specified in the scenario if the company makes any international transactions, but if they do, it is important to consider any additional regulations or restrictions that may apply.

It is also not specified if any governmental restrictions on secure communications exist. However, it is important to research and comply with any relevant regulations and laws regarding data privacy and security.

Potential external threats that may be present now and in the immediate future include hacking, phishing, and denial of service attacks. It is also important to consider the potential for insider threats.

Modernization requirements that must be considered include the role of open-source libraries and evolving web application technologies. It is important to ensure that any libraries or technologies used are secure and up to date to minimize vulnerabilities. Additionally, it may be necessary to implement new security measures such as two-factor authentication or encryption to protect sensitive financial information.

## Areas of Security

In conducting a vulnerability assessment for Artemis Financials’ web-based software application, the relevant areas of security to consider would likely include authentication and access control, data validation, input sanitation, and secure communication. These areas are relevant to the software application as they ensure secure access to the application and protect clients' sensitive financial information. Additionally, any identified vulnerabilities in these areas should be documented in a vulnerability assessment report and mitigation strategies should be proposed.

## Manual Review

During the manual review, several vulnerabilities were identified in the Project One Code Base. These include business names being sent as request parameters in the CRUDController class, which could result in a leak of sensitive information; database connection parameters being hard-coded in DocData, which could lead to unauthorized access; request parameters not being validated, which could result in injection attacks; the service not using HTTPS, which could lead to man-in-the-middle attacks; and no authentication scheme present, which could allow unauthorized access to the system. These vulnerabilities should be addressed in the vulnerability assessment report and appropriate measures should be taken to mitigate them.

## Static Testing

The static testing using the dependency-check plug-in has identified several vulnerabilities in the code base of Artemis Financial's software application. These vulnerabilities are present in various libraries and dependencies used by the application, including Bouncy Castle, FasterXML Jackson Databind, Apache Log4j API, SnakeYAML, Spring Core, and Apache Tomcat. The specific vulnerabilities identified along with the recommended solutions are as follows:

* 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

These vulnerabilities should be addressed as soon as possible by updating the affected libraries and dependencies to the recommended versions or later. Additionally, it is important to regularly check for updates and vulnerabilities in all libraries and dependencies used by the application to ensure its continued security.

## Mitigation Plan

1. Business names should not be sent as request parameters in the CRUDController class. Implement proper validation and sanitization of request parameters to prevent potential injection attacks.
2. Database connection parameters should not be hard coded in DocData. Use environment variables or a configuration file to store sensitive information.
3. Implement HTTPS to ensure secure communication between client and server.
4. Implement an authentication scheme to ensure only authorized users can access the application.
5. Update Bouncy Castle to at least version 1.60 to address known vulnerabilities.
6. Update FasterXML Jackson Databind to at least version 2.10.5.1 to address known vulnerabilities.
7. Update Apache Log4j API to at least version 2.13.2 to address known vulnerabilities.
8. Update SnakeYAML to at least version 1.26 to address known vulnerabilities.
9. Update Spring Core to at least version 5.2.8 to address known vulnerabilities.
10. Update Apache Tomcat to at least version 9.0.40 to address known vulnerabilities.