****

# 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 9, 2023** | **Carisma Carter** | **Vulnerability 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

Carisma Carter

## Interpreting Client Needs

Artemis Financial is a consulting company that develops individualized financial plans for their customers. The financial plans include savings, retirement, investments, and insurance. This involves them holding highly sensitive information that is constantly susceptible to attacks. Therefore, secure communication is the top priority and holds great value in this company. As a financial institution, Artemis Financial is obligated to follow governmental regulations and limitations for all transactions and communications. Failing to do so can pose a security risk for both the company and its clients. Any external threats might be present now and in the immediate future would most certainly be an attack on gaining the clients and the company’s financial information. Open-source libraries play a significant role for Artemis Financial. They provide valuable resources and tools that can be used to enhance the company's software development processes and infrastructure through security, reliability, and timesaving. The forever-evolving web application technologies play a role in the company’s modernization requirements by web application technologies evolving to address emerging security threats while complying with industry regulations. Artemis Financial can take advantage of advancements in authentication, encryption, and secure coding practices to protect sensitive financial data and ensure compliance.

## Areas of Security

The client/server involves communication between the client-side and server-side. Security measures like secure communication protocols, HTTPS, and access controls should be implemented to protect integrity. Cryptography plays a critical role in securing sensitive data in transit and at rest. Artemis Financials’ web application should utilize strong cryptographic algorithms and protocols to encrypt sensitive information, such as user credentials, financial transactions, and other confidential data. Proper input validation is crucial to prevent malicious input or unauthorized data from being processed by the application. By implementing strong input validation practices, Artemis Financial can protect against common security vulnerabilities like SQL injection and command injection.

## Manual Review

During manual review, the following vulnerabilities were identified in Artemis Financials’ web-based software application:

* The lack of validation for requests exposes the system to potential external threats.
* Verification mechanisms for user authentication are absent.
* The service does not employ HTTP, which is advisable for secure transmission of sensitive data.
* The request parameters in the CRUD Controller class include business names, which can be manipulated by unauthorized individuals due to the sensitive information being exposed.

## Static Testing

A screenshot of a computer

Description automatically generated

1. bcprov-jdk15on-1.46.jar - multiple vulnerabilities, update to the latest version

* CVE-2016-1000338
* CVE-2016-1000342
* CVE-2016-1000343
* CVE-2016-1000344
* CVE-2016-1000352
* CVE-2016-1000341
* CVE-2016-1000345
* CVE-2017-13098
* CVE-2020-15522
* CVE-2016-1000339
* CVE-2015-7940
* CVE-2018-5382
* CVE-2013-1624
* CVE-2016-1000346

1. hibernate-validator-6.0.18.Final.jar - only one vulnerability, update to the latest version

* CVE-2020-10693

1. jackson-databind-2.10.2.jar - multiple vulnerabilities, update to the latest version

* CVE-2020-25649
* CVE-2020-36518
* CVE-2021-46877
* CVE-2022-42003
* CVE-2022-42004
* CVE-2023-35116

1. log4j-api-2.12.1.jar- only one vulnerability, update to the latest version

* CVE-2020-9488

1. logback-core-1.2.3.jar - only one vulnerability, update to the latest version

* CVE-2021-42550

1. snakeyaml-1.25.jar - multiple vulnerabilities, update to the latest version

* CVE-2022-1471
* CVE-2017-18640
* CVE-2022-25857
* CVE-2022-38749
* CVE-2022-38751
* CVE-202238752
* CVE-2022-41854
* CVE-202238750

1. spring-boot-2.2.4.RELEASE.jar- only two vulnerabilities, update to the latest version

* CVE-2022-27772
* CVE-2023-20883

1. spring-boot-starter-web-2.2.4.RELEASE.jar - only two vulnerabilities, update to the latest version

* CVE-2022-27772
* CVE-2023-20883

1. spring-core-5.2.3.RELEASE.jar - multiple vulnerabilities, update to the latest version

* CVE-2022-22965
* CVE-2021-22118
* CVE-2020-5421
* CVE-2022-22950
* CVE-2022-22971
* CVE-2023-20861
* CVE-2023-20863
* CVE-2022-22968
* CVE-2022-22970
* CVE-2021-22060
* CVE-2021-22096

1. spring-web-5.2.3.RELEASE.jar - multiple vulnerabilities, update to the latest version

* CVE-2016-1000027
* CVE-2022-22965
* VE-2021-22118
* CVE-2020-5421
* CVE-2022-22950
* CVE-2022-22971
* CVE-2023-20861
* CVE-2023-20863
* CVE-2022-22968
* CVE-2022-22970
* CVE- 2021-22060
* CVE-2021-22096

1. spring-webmvc-5.2.3.RELEASE.jar- multiple vulnerabilities, update to the latest version

* CVE-2022-22965
* CVE-2020-5421
* CVE-2022-22971
* CVE-2023-20861
* CVE-2023-20863
* CVE-2022-22968
* CVE-2022-22970
* CVE-2021-22060
* CVE-2021-22096

1. tomcat-embed-core-9.0.30.jar – multiple vulnerabilities, update to the latest version

* CVE-2020-1938
* CVE-2020-11996
* CVE-2020-13934
* CVE-2020-13935
* CVE-2020-17527
* CVE-2021-25122
* CVE-2021-41079
* CVE-2022-29885
* CVE-2022-42252
* CVE-2020-9484
* CVE-2020-1935

1. tomcat-embed-websocket-9.0.30.jar – multiple vulnerabilities, update to the latest version

* CVE-2020-1938
* CVE-2020-8022
* CVE-2020-11996
* CVE-2020-13934
* CVE-2020-13935
* CVE-2021-25122
* CVE-2021-43980

## Mitigation Plan

Listed are some vulnerabilities found from the manual review and steps to mitigate the identified security vulnerabilities:

Cross-Site Scripting (XSS):

* Implement proper input validation and sanitization techniques to filter out malicious input.
* Encode or escape user-generated data before rendering it in web pages.
* Utilize security libraries or frameworks that offer built-in protections against XSS attacks.

Exposure of Sensitive Information to an Unauthorized Actor:

* Identify and classify sensitive information within the web application, such as user credentials, financial data, or personally identifiable information (PII).
* Implement strong access controls to restrict access to sensitive data based on user roles and privileges.
* Regularly review and audit access controls to ensure they are configured correctly and enforced consistently.

Missing Release of Memory after Effective Lifetime:

* Conduct a code review to identify areas where memory is allocated but not properly released.
* Implement automated tools or frameworks that assist in identifying memory leaks or improper memory management.
* Follow best practices for memory management, including deallocating memory when it is no longer needed.

.