****

# Artemis Financial Vulnerability Assessment Report

Table of Contents

[Artemis Financial Vulnerability Assessment Report 1](#_Toc162203373)

[Document Revision History 3](#_Toc162203374)

[Client 3](#_Toc162203375)

[Instructions 3](#_Toc162203376)

[Developer 4](#_Toc162203377)

[1. Interpreting Client Needs 4](#_Toc162203378)

[2. Areas of Security 4](#_Toc162203379)

[3. Manual Review 4](#_Toc162203380)

[4. Static Testing 5](#_Toc162203381)

[5. Mitigation Plan 7](#_Toc162203382)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **03/24/2024** | **Michael Kinful** |  |

## 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

Michael Kinful

## Interpreting Client Needs

Artemis Financial, a firm handling sensitive client information, must prioritize both internal and external secure communication. Given its operations in handling transactions globally, it's vital to secure data transfer regardless of the lack of current state or federal mandates on secure communications. This precaution is necessary to protect against the unauthorized disclosure of customer details. The firm is responsible for safeguarding a range of confidential client data, including biometric identifiers, social security numbers, and account details, as well as Artemis Financials’ own trade secrets. This necessitates encryption or other forms of data masking during storage and transmission. Moreover, to keep pace with technological advances and counteract evolving security threats, Artemis Financial must regularly update its software libraries and ensure the deployment of the latest security patches and bug fixes in its applications.

## Areas of Security

Artemis Financial implements stringent security measures in its RESTful API to protect critical financial information. Through meticulous input validation, the system is safeguarded against unauthorized data entries, mitigating potential risks. The API's security framework is reinforced with robust authentication and authorization mechanisms, ensuring that access is strictly regulated. Additionally, the use of generic error messages prevents attackers from exploiting system weaknesses. Adherence to best coding practices and the principle of encapsulation—via private variables and restricted access methods—further secures sensitive client data.

## Manual Review

A screenshot of a computer

Description automatically generated

## Static Testing

**Dependency:** bcprov-jdk15on-1.46.jar

**Vulnerability ID:** cpe:2.3:a:bouncycastle:legion-of-the-bouncy-castle-java-crytography-api:1.46:\*:\*:\*:\*:\*:\*:\*

**Description:** The Bouncy Castle Crypto package is a Java implementation of cryptographic algorithms. This jar contains JCE provider and lightweight API for the Bouncy Castle Cryptography APIs for JDK 1.5 to JDK 1.7.

**Resolution:** Upgrading to the fixed versions mitigates this risk by correcting how the library handles timing information, preventing attackers from potentially decrypting data.

**Dependency:** hibernate-validator-6.0.18.Final.jar

**Vulnerability ID:** cpe:2.3:a:bouncycastle:legion-of-the-bouncy-castle-java-crytography-api:1.46:\*:\*:\*:\*:\*:\*:\*

**Description:** The Bouncy Castle Crypto package is a Java implementation of cryptographic algorithms. This jar contains JCE provider and lightweight API for the Bouncy Castle Cryptography APIs for JDK 1.5 to JDK 1.7.

**Resolution:** Update the lib hibernate-validator-java package if using Ubuntu. For direct management, update Hibernate Validator to a patched version.

**Dependency:** jackson-databind-2.10.2.jar

**Vulnerability ID:** cpe:2.3:a:fasterxml:jackson-databind:2.10.2:\*:\*:\*:\*:\*:\*:\*

**Description:** A flaw was found in FasterXML Jackson Databind, where it did not have entity expansion secured properly. This flaw allows vulnerability to XML external entity (XXE) attacks. The highest threat from this vulnerability is data integrity.

**Resolution:** Update the library to address this XML external entity (XXE) vulnerability. You can find the updated versions on the Jackson project website.

**Dependency:** log4j-api-2.12.1.jar

**Vulnerability ID:** cpe:2.3:a:apache:log4j:2.12.1:\*:\*:\*:\*:\*:\*:\*

**Description:** Improper validation of certificate with host mismatch in Apache Log4j SMTP appender. This could allow an SMTPS connection to be intercepted by a man-in-the-middle attack which could leak any log messages sent through that appender.

**Resolution:** Upgrade to Apache Log4j 2.12.3 or later, or 2.13.1 or later. These fixes address improper certificate validation that could allow attackers to intercept log messages.

**Dependency:** logback-core-1.2.3.jar

**Vulnerability ID:** cpe:2.3:a:qos:logback:1.2.3:\*:\*:\*:\*:\*:\*:\*

**Description:** In logback version 1.2.7 and prior versions, an attacker with the required privileges to edit configurations files could craft a malicious configuration allowing to execute arbitrary code loaded from LDAP servers.

**Resolution:** Upgrade logback to version 1.2.8 or later.

**Dependency:** snakeyaml-1.25.jar

**Vulnerability ID:** cpe:2.3:a:snakeyaml\_project:snakeyaml:1.25:\*:\*:\*:\*:\*:\*:\*

cpe:2.3:a:yaml\_project:yaml:1.25:\*:\*:\*:\*:\*:\*:\*

**Description:** The Alias feature in SnakeYAML before 1.26 allows entity expansion during a load operation, a related issue to CVE-2003-1564.

**Resolution:** Upgrade to a non-vulnerable version.

**Dependency:** spring-beans-5.2.3.RELEASE.jar

**Vulnerability ID:** cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*

cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*

**Description:** A Spring MVC or Spring WebFlux application running on JDK 9+ may be vulnerable to remote code execution (RCE) via data binding. The specific exploit requires the application to run on Tomcat as a WAR deployment. If the application is deployed as a Spring Boot executable jar, i.e. the default, it is not vulnerable to the exploit. However, the nature of the vulnerability is more general, and there may be other ways to exploit it.

**Resolution:** Upgrade to Spring Boot 2.6.6, 2.5.12 and Spring Framework 5.3.16, 5.2.20

**Dependency:**  spring-boot-2.2.4.RELEASE.jar

**Vulnerability ID:** cpe:2.3:a:vmware:spring\_boot:2.2.4:release:\*:\*:\*:\*:\*:\*

**Description:** Spring-boot versions prior to version v2.2.11.RELEASE was vulnerable to temporary directory hijacking. This vulnerability impacted the org.springframework.boot.web.server.AbstractConfigurableWebServerFactory.createTempDir method. NOTE: This vulnerability only affects products and/or versions that are no longer supported by the maintainer

**Resolution:** Update to Spring Boot version 2.2.11

**Dependency:** spring-context-5.2.3.RELEASE.jar

**Vulnerability ID:** cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*

cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*

**Description:** In Spring Framework versions 5.3.0 - 5.3.18, 5.2.0 - 5.2.20, and older unsupported versions, the patterns for disallowedFields on a DataBinder are case sensitive which means a field is not effectively protected unless it is listed with both upper and lower case for the first character of the field, including upper and lower case for the first character of all nested fields within the property path.

**Resolution:** Upgrade to new versions.

**Dependency:** spring-expression-5.2.3.RELEASE.jar

**Vulnerability ID:** cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*

cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*

**Description:** In Spring Framework versions 5.3.0 - 5.3.16 and older unsupported versions, it is possible for a user to provide a specially crafted SpEL expression that may cause a denial of service condition.

**Resolution:** Upgrade the affected Spring products to a non vulnerable version.

**Dependency:** spring-web-5.2.3.RELEASE.jar

**Vulnerability ID:** cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*

cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*

**Description:** Pivotal Spring Framework through 5.3.16 suffers from a potential remote code execution (RCE) issue if used for Java deserialization of untrusted data. Depending on how the library is implemented within a product, this issue may or not occur, and authentication may be required. NOTE: the vendor's position is that untrusted data is not an intended use case. The product's behavior will not be changed because some users rely on deserialization of trusted data.

**Resolution:** Update to the latest version.

**Dependency:** spring-webmvc-5.2.3.RELEASE.jar

**Vulnerability ID:** cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*

cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*

**Description:** n Spring Framework versions 5.3.0 - 5.3.16 and older unsupported versions, it is possible for a user to provide a specially crafted SpEL expression that may cause a denial of service condition.

**Resolution:** Update to a patched version of the spring-webmvc.

**Dependency:** tomcat-embed-core-9.0.30.jar

**Vulnerability ID:** cpe:2.3:a:apache:tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*

cpe:2.3:a:apache\_tomcat:apache\_tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*

**Description:** The refactoring present in Apache Tomcat 9.0.28 to 9.0.30, 8.5.48 to 8.5.50 and 7.0.98 to 7.0.99 introduced a regression. The result of the regression was that invalid Transfer-Encoding headers were incorrectly processed leading to a possibility of HTTP Request Smuggling if Tomcat was located behind a reverse proxy that incorrectly handled the invalid Transfer-Encoding header in a particular manner. Such a reverse proxy is considered unlikely.

**Resolution:** Upgrade to the latest version of tomcat-embed.

## Mitigation Plan

To maintain both security and operational efficiency, a thorough update of the codebase is necessary. Begin by upgrading all dependent libraries, which may entail rewriting parts of the code to align with the new updates. Once updates are integrated and the code compiles successfully, proceed to internal testing. Post-testing, refine the code to include clearer error messages, enhance data input checks and cleansing, and introduce security improvements such as password salting and hashing. To further reduce security vulnerabilities, adopting code review practices or collaborative programming during this overhaul is advisable.