****

# 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** | **03/20/2024** | **Logan King** |  |

## 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

Logan King

## Interpreting Client Needs

Artemis Financial is a firm specializing in crafting personalized financial strategies for clients, covering a spectrum from savings and retirement planning to investments and insurance. Given the sensitive nature of the information it handles, including social security numbers and tax data, secure communication channels are of paramount importance. While the company's geographical scope isn't explicitly mentioned, it's reasonable to assume it deals with international transactions. One significant governmental consideration revolves around safeguarding against the exposure of trade secrets. Hence, Artemis Financial must ensure stringent measures are in place to protect all client data from external threats. Encryption emerges as a crucial tool in fortifying data against unauthorized access. Furthermore, staying abreast of the latest security protocols and conducting regular maintenance checks to address vulnerabilities and software bugs are essential modern requirements for Artemis Financial. This proactive approach is vital in maintaining the integrity of their systems and safeguarding client information effectively.

## Areas of Security

Artemis Financial recognizes the importance of input validation as a means to authenticate the owner of information, enhancing user protection. This validation process is designed to verify strings of data, ensuring the legitimacy of user input. In terms of code quality, stringent measures are implemented to control access to methods based on user permissions. For instance, users are only granted access to their own information, preventing unauthorized access to other users' data or server resources. The development of APIs is essential for Artemis Financial, given its operation both internally and externally. APIs facilitate controlled access to acceptable data, streamlining interactions between various systems and ensuring secure data exchange. Error handling implementation is crucial to identify and rectify issues within the API promptly. By effectively managing errors, Artemis Financial can mitigate the risk of exposing user information and maintain the integrity of its systems. So, cryptography plays a pivotal role within Artemis Financials’ infrastructure, safeguarding user information across different regions and currencies. By employing robust encryption techniques, the company ensures the confidentiality and security of sensitive data, regardless of geographical location.

## Manual Review

Upon conducting a Vulnerability Assessment on the code base, I examined both the POM.XML file and the Greeting Controller. In the XML file, I searched for evidence of an Apache Validator to ensure secure data validation. However, I did not find any indication of its presence. In the Greeting Controller, I observed a notable vulnerability related to input validation. There was a lack of input validation, which poses a potential security risk. This issue needs to be addressed promptly to enhance the overall security posture. Regarding code quality, it was generally satisfactory; however, there was a glaring omission in error handling. The absence of proper error handling mechanisms leaves the system vulnerable to potential exploitation and undermines its reliability. Moving on to the API, several deficiencies were identified. One significant vulnerability pertains to data exposure due to the absence of secure data transmission protocols. Specifically, user input was not processed through a POST method, increasing the risk of unauthorized access to sensitive information. Lastly, I attempted to identify cryptography measures within the code base but was unable to find any evidence of their implementation. This represents a critical gap in data security, as cryptography is essential for safeguarding user information, particularly in scenarios involving diverse geographical regions and currencies. Addressing these vulnerabilities is imperative to fortify the system's security posture and mitigate potential risks.

## Static Testing

|  |  |  |  |
| --- | --- | --- | --- |
| Dependency | Vulnerability | Description | Published Vulnerabilities |
| [bcprov-jdk15on-1.46.jar](file:///C:\Users\Logan's%20PC\Desktop\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l2_991c96a4e31e6c19e2b9136c8955bd423f2dc4c7) | [cpe:2.3:a:bouncycastle:legion-of-the-bouncy-castle-java-crytography-api:1.46:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Abouncycastle&cpe_product=cpe%3A%2F%3Abouncycastle%3Alegion-of-the-bouncy-castle-java-crytography-api&cpe_version=cpe%3A%2F%3Abouncycastle%3Alegion-of-the-bouncy-castle-java-crytography-api%3A1.46) | 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. | The TLS implementation in the Bouncy Castle Java library before 1.48 and C# library before 1.8 does not properly consider timing side-channel attacks on a noncompliant MAC check operation during the processing of malformed CBC padding, which allows remote attackers to conduct distinguishing attacks and plaintext-recovery attacks via statistical analysis of timing data for crafted packets, a related issue to CVE-2013-0169. |
| [spring-boot-2.2.4.RELEASE.jar](file:///C:\Users\Logan's%20PC\Desktop\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l3_225a4fd31156c254e3bb92adb42ee8c6de812714) | [cpe:2.3:a:vmware:spring\_boot:2.2.4:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Avmware&cpe_product=cpe%3A%2F%3Avmware%3Aspring_boot&cpe_version=cpe%3A%2F%3Avmware%3Aspring_boot%3A2.2.4) | Spring Boot | 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 |
| [logback-core-1.2.3.jar](file:///C:\Users\Logan's%20PC\Desktop\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l4_864344400c3d4d92dfeb0a305dc87d953677c03c) | [cpe:2.3:a:qos:logback:1.2.3:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aqos&cpe_product=cpe%3A%2F%3Aqos%3Alogback&cpe_version=cpe%3A%2F%3Aqos%3Alogback%3A1.2.3) | logback-core module | 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. |
| [log4j-api-2.12.1.jar](file:///C:\Users\Logan's%20PC\Desktop\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l5_a55e6d987f50a515c9260b0451b4fa217dc539cb) | [cpe:2.3:a:apache:log4j:2.12.1:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aapache&cpe_product=cpe%3A%2F%3Aapache%3Alog4j&cpe_version=cpe%3A%2F%3Aapache%3Alog4j%3A2.12.1) | The Apache Log4j API | 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. Fixed in Apache Log4j 2.12.3 and 2.13.1 |
| [snakeyaml-1.25.jar](file:///C:\Users\Logan's%20PC\Desktop\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l8_8b6e01ef661d8378ae6dd7b511a7f2a33fae1421) | [cpe:2.3:a:snakeyaml\_project:snakeyaml:1.25:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Asnakeyaml_project&cpe_product=cpe%3A%2F%3Asnakeyaml_project%3Asnakeyaml&cpe_version=cpe%3A%2F%3Asnakeyaml_project%3Asnakeyaml%3A1.25) [cpe:2.3:a:yaml\_project:yaml:1.25:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Ayaml_project&cpe_product=cpe%3A%2F%3Ayaml_project%3Ayaml&cpe_version=cpe%3A%2F%3Ayaml_project%3Ayaml%3A1.25) | YAML 1.1 parser and emitter for Java | The Alias feature in SnakeYAML before 1.26 allows entity expansion during a load operation, a related issue to CVE-2003-1564. |
| [jackson-databind-2.10.2.jar](file:///C:\Users\Logan's%20PC\Desktop\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l9_0528de95f198afafbcfb0c09d2e43b6e0ea663ec) | [cpe:2.3:a:fasterxml:jackson-databind:2.10.2:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Afasterxml&cpe_product=cpe%3A%2F%3Afasterxml%3Ajackson-databind&cpe_version=cpe%3A%2F%3Afasterxml%3Ajackson-databind%3A2.10.2) | General data-binding functionality for Jackson: works on core streaming API | 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. |
| [tomcat-embed-core-9.0.30.jar](file:///C:\Users\Logan's%20PC\Desktop\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l13_ad32909314fe2ba02cec036434c0addd19bcc580) | [cpe:2.3:a:apache:tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aapache&cpe_product=cpe%3A%2F%3Aapache%3Atomcat&cpe_version=cpe%3A%2F%3Aapache%3Atomcat%3A9.0.30) [cpe:2.3:a:apache\_tomcat:apache\_tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aapache_tomcat&cpe_product=cpe%3A%2F%3Aapache_tomcat%3Aapache_tomcat&cpe_version=cpe%3A%2F%3Aapache_tomcat%3Aapache_tomcat%3A9.0.30) | Core Tomcat implementation | 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. |
| [hibernate-validator-6.0.18.Final.jar](file:///C:\Users\Logan's%20PC\Desktop\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l16_7fd00bcd87e14b6ba66279282ef15efa30dd2492) | [cpe:2.3:a:redhat:hibernate\_validator:6.0.18:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aredhat&cpe_product=cpe%3A%2F%3Aredhat%3Ahibernate_validator&cpe_version=cpe%3A%2F%3Aredhat%3Ahibernate_validator%3A6.0.18) | Hibernate's Bean Validation (JSR-380) reference implementation. | A flaw was found in Hibernate Validator version 6.1.2.Final. A bug in the message interpolation processor enables invalid EL expressions to be evaluated as if they were valid. This flaw allows attackers to bypass input sanitation (escaping, stripping) controls that developers may have put in place when handling user-controlled data in error messages. |
| [spring-web-5.2.3.RELEASE.jar](file:///C:\Users\Logan's%20PC\Desktop\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l19_dd386a02e40b915ab400a3bf9f586d2dc4c0852c) | [cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Apivotal_software&cpe_product=cpe%3A%2F%3Apivotal_software%3Aspring_framework&cpe_version=cpe%3A%2F%3Apivotal_software%3Aspring_framework%3A5.2.3) [cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aspringsource&cpe_product=cpe%3A%2F%3Aspringsource%3Aspring_framework&cpe_version=cpe%3A%2F%3Aspringsource%3Aspring_framework%3A5.2.3) | Spring Web | 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. |
| [spring-beans-5.2.3.RELEASE.jar](file:///C:\Users\Logan's%20PC\Desktop\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l20_0250c8c641433dc06b1b44e4563fa08a2fbf8954) | [cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Apivotal_software&cpe_product=cpe%3A%2F%3Apivotal_software%3Aspring_framework&cpe_version=cpe%3A%2F%3Apivotal_software%3Aspring_framework%3A5.2.3) [cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aspringsource&cpe_product=cpe%3A%2F%3Aspringsource%3Aspring_framework&cpe_version=cpe%3A%2F%3Aspringsource%3Aspring_framework%3A5.2.3) | Spring Beans | 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. |
| [spring-webmvc-5.2.3.RELEASE.jar](file:///C:\Users\Logan's%20PC\Desktop\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l21_745a62502023d2496b565b7fe102bb1ee229d6b7) | [cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Apivotal_software&cpe_product=cpe%3A%2F%3Apivotal_software%3Aspring_framework&cpe_version=cpe%3A%2F%3Apivotal_software%3Aspring_framework%3A5.2.3) [cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aspringsource&cpe_product=cpe%3A%2F%3Aspringsource%3Aspring_framework&cpe_version=cpe%3A%2F%3Aspringsource%3Aspring_framework%3A5.2.3) | Spring Web MVC | In Spring Framework versions 5.3.0 - 5.3.13, 5.2.0 - 5.2.18, and older unsupported versions, it is possible for a user to provide malicious input to cause the insertion of additional log entries. This is a follow-up to CVE-2021-22096 that protects against additional types of input and in more places of the Spring Framework codebase. |
| [spring-context-5.2.3.RELEASE.jar](file:///C:\Users\Logan's%20PC\Desktop\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l22_7750c95c96c7a1885c8b1b503ba915bc33ca579a) | [cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Apivotal_software&cpe_product=cpe%3A%2F%3Apivotal_software%3Aspring_framework&cpe_version=cpe%3A%2F%3Apivotal_software%3Aspring_framework%3A5.2.3) [cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aspringsource&cpe_product=cpe%3A%2F%3Aspringsource%3Aspring_framework&cpe_version=cpe%3A%2F%3Aspringsource%3Aspring_framework%3A5.2.3) | Spring Context | 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. |
| [spring-expression-5.2.3.RELEASE.jar](file:///C:\Users\Logan's%20PC\Desktop\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l23_d0c6bb10758805b2153c589686b8045554bfac2d) | [cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Apivotal_software&cpe_product=cpe%3A%2F%3Apivotal_software%3Aspring_framework&cpe_version=cpe%3A%2F%3Apivotal_software%3Aspring_framework%3A5.2.3) [cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aspringsource&cpe_product=cpe%3A%2F%3Aspringsource%3Aspring_framework&cpe_version=cpe%3A%2F%3Aspringsource%3Aspring_framework%3A5.2.3) | Spring Expression Language (SpEL) | 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. |

## Mitigation Plan

To swiftly address the identified security vulnerabilities, an initial step involves updating all outdated frameworks to their latest versions. This proactive measure ensures that known vulnerabilities present in each dependency are patched effectively. By staying current with updates, Artemis Financial can mitigate potential risks associated with outdated software and bolster the overall security of its systems.