****

# 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** | **9/17/2023** | **Sean Glass** |  |

## Client



## Developer

Sean Glass

## Interpreting Client Needs

Artemis Financial develops financial plans for their clients, with plans including savings, retirement, investments, and insurance. Due to the sensitivity of the client’s data, secure communication is of the utmost importance. They will be making financial transactions, domestically and internationally, so there are certain governmental restrictions in relation to secure communications. Because the client is dealing with sensitive financial data, external threats are almost guaranteed to happen now or in the future. The client has also expressed the desire to modernize their operations by using the most current and effective software security.

## Areas of Security

* Input Validation
  + Users must be validated to ensure they are accessing only the information they are allowed to see. If validation is not used, someone could easily access other users’ sensitive financial information.
* APIs
  + APIs are relevant because of the sensitive data that is transmitted from the server to the client. The APIs help provide authentication and authorization to keep the user’s information confidential.
* Cryptography
  + Without encryption, bad actors can get access to sensitive financial information by intercepting the API request. In these situations, if encryption is used appropriately, the data is only readable by the sender and the recipient.
* Code Error
  + Code errors are relevant because errors can lead to data corruption, privacy concerns, unauthorized access, data leakage, etc.
* Code Quality
  + High-quality code is well-structured and is, therefore, less prone to certain vulnerabilities, such as SQL injections. Also, appropriate input validation helps make the application less susceptible to attacks.
* Encapsulation
  + Encapsulation ensures that sensitive financial information is restricted to authorized individuals, which reduces the likelihood of a successful attack.

## Manual Review

In CRUDController.java, there is no input validation currently being implemented. Input should be validated before accessing DocData. In DocData.java, the code attempts to connect to a database using hard-coded username and password. In GreetingController.java, there is no input validation being implemented. Users should be properly validated to prevent security issues. In myDateTime.java, class fields should be encapsulated.

## Static Testing

|  |  |
| --- | --- |
| Vulnerability ID | Description |
| CVE-2016-1000352 | In the Bouncy Castle JCE Provider version 1.55 and earlier the ECIES implementation allowed the use of ECB mode. This mode is regarded as unsafe and support for it has been removed from the provider. |
| CVE-2023-20883 | In Spring Boot versions 3.0.0 - 3.0.6, 2.7.0 - 2.7.11, 2.6.0 - 2.6.14, 2.5.0 - 2.5.14 and older unsupported versions, there is potential for a denial-of-service (DoS) attack if Spring MVC is used together with a reverse proxy cache. |
| CVE-2021-42550 | 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. |
| CVE-2021-44832 | Apache Log4j2 versions 2.0-beta7 through 2.17.0 (excluding security fix releases 2.3.2 and 2.12.4) are vulnerable to a remote code execution (RCE) attack when a configuration uses a JDBC Appender with a JNDI LDAP data source URI when an attacker has control of the target LDAP server. This issue is fixed by limiting JNDI data source names to the java protocol in Log4j2 versions 2.17.1, 2.12.4, and 2.3.2. |
| CVE-2022-1471 | SnakeYaml's Constructor() class does not restrict types which can be instantiated during deserialization. Deserializing yaml content provided by an attacker can lead to remote code execution. We recommend using SnakeYaml's SafeConsturctor when parsing untrusted content to restrict deserialization. We recommend upgrading to version 2.0 and beyond. |
| CVE-2022-3064 | Parsing malicious or large YAML documents can consume excessive amounts of CPU or memory. |
| CVE-2023-35116 | \*\* DISPUTED \*\* jackson-databind through 2.15.2 allows attackers to cause a denial of service or other unspecified impact via a crafted object that uses cyclic dependencies. NOTE: the vendor's perspective is that this is not a valid vulnerability report, because the steps of constructing a cyclic data structure and trying to serialize it cannot be achieved by an external attacker. |
| CVE-2023-41080 | URL Redirection to Untrusted Site ('Open Redirect') vulnerability in FORM authentication feature Apache Tomcat.This issue affects Apache Tomcat: from 11.0.0-M1 through 11.0.0-M10, from 10.1.0-M1 through 10.0.12, from 9.0.0-M1 through 9.0.79 and from 8.5.0 through 8.5.92. The vulnerability is limited to the ROOT (default) web application. |
| CVE-2020-10693 | 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. |

## Mitigation Plan

To mitigate the identified security vulnerabilities, I recommend the following actions:

* Use the latest Bouncy Castle JCE Provider version.
* Update Spring Boot to the latest version.
* Implement user validation within program.
* Ensure high-quality code is written.
* Do not hard-code sensitive information.
* Ensure encapsulation is used appropriately.
* Do not parse malicious or large YAML documents.