****

# 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** | **9/10/2023** | **Arica Bryant** | **Updated “Interpret client needs” section** |
| **1.1** | **9/12/2023** | **Arica Bryant** | **“Areas of Security” Updated** |
| **1.2** | **9/13/2023** | **Arica Bryant** | **“Manual Review” Updated** |
| **1.3** | **9/15/2023** | **Arica Bryant** | **“Static Testing” and “Mitigation Plan” Updated** |

## Client



## Developer

Arica Bryant

## Interpreting Client Needs

Secure communications will be extremely valuable to the client, Artemis Financial. They are a consulting company that will be handling the personal financial information of customers. They create financial plans for customers that include savings, retirements, investments, and insurance plans. This data should be secure and safe as it is transferred so that the private information of customers is protected. Artemis Financial will likely be making international transactions with customers. This form of transactional communication would also need to be secured.

Possible external threats that might occur now and in the immediate future are DoS (Denial of Service) attacks. Denial of Service attacks occur when hackers are able to restrict access to a system due to a vulnerability. Since the company works with APIs, data breaches are another potential threat where information can be stolen or spied upon during communication. Examples include the “Man In The Middle” attack where an unauthorized third-party is spying on information that is transferred through the API.

Since web application technologies are constantly evolving, the company should make sure that any frameworks, resources, or APIs are kept up to date. Old software can open up the application to more vulnerabilities.

## Areas of Security

Code Quality: Code Quality is important when working on any software development project. Poorly written code can make any issues that occur difficult to troubleshoot and fix. Errant code can also be taken advantage of by hackers who will try and exploit these vulnerabilities.

APIs: The company will be using an API to transfer important financial information back and forth. The communication performed by the API should be monitored to prevent unauthorized access to private information such as the previously mentioned “Man In The Middle” attack.

Input Validation: The company will most likely be accepting user inputs for their web application. Any form of unput that is received by the web application should be verified and checked for security.

Code Error: It is important for there to secure error handling within the project so that it will be easier to tell where a problem is occurring. This decreases the amount of time it takes to find and fix a security issue when it happens.

## Manual Review

When manually analyzing the pom.xml file for the project, a lot of the frameworks are out of date. The program is more susceptible to the vulnerabilities of these older installations. In the “DocData.java” file there is no form of error handling implemented. In the same file, this document is opened and read but the document is not closed after use. There is also a lack of user input validation in the program.

## Static Testing

|  |  |  |
| --- | --- | --- |
| Dependency Name | Description | CVE Codes |
| bcprov-jdk15on-1.46.jar | 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. | [**CVE-2016-1000352**](https://nvd.nist.gov/vuln/detail/CVE-2016-1000352)  [**CVE-2016-1000346**](https://nvd.nist.gov/vuln/detail/CVE-2016-1000346)  [**CVE-2016-1000345**](https://nvd.nist.gov/vuln/detail/CVE-2016-1000345)  [**CVE-2016-1000344**](https://nvd.nist.gov/vuln/detail/CVE-2016-1000344)  [**CVE-2016-1000343**](https://nvd.nist.gov/vuln/detail/CVE-2016-1000343)  [**CVE-2016-1000342**](https://nvd.nist.gov/vuln/detail/CVE-2016-1000342)  [**CVE-2016-1000341**](https://nvd.nist.gov/vuln/detail/CVE-2016-1000341)  [**CVE-2016-1000339**](https://nvd.nist.gov/vuln/detail/CVE-2016-1000339)  [**CVE-2016-1000338**](https://nvd.nist.gov/vuln/detail/CVE-2016-1000338)  [**CVE-2018-5382**](https://nvd.nist.gov/vuln/detail/CVE-2018-5382)  [**CVE-2017-13098**](https://nvd.nist.gov/vuln/detail/CVE-2017-13098)  [**CVE-2013-1624**](https://nvd.nist.gov/vuln/detail/CVE-2013-1624) |
| hibernate-validator-6.0.18.Final.jar | Hibernate's Bean Validation (JSR-380) reference implementation. | [**CVE-2020-10693**](https://nvd.nist.gov/vuln/detail/CVE-2020-10693) |
| jackson-databind-2.10.2.jar | General data-binding functionality for Jackson: works on core streaming API | [**CVE-2023-35116**](https://nvd.nist.gov/vuln/detail/CVE-2023-35116)  [**CVE-2021-46877**](https://nvd.nist.gov/vuln/detail/CVE-2021-46877)  [**CVE-2022-42004**](https://nvd.nist.gov/vuln/detail/CVE-2022-42004)  [**CVE-2022-42003**](https://nvd.nist.gov/vuln/detail/CVE-2022-42003)  [**CVE-2020-36518**](https://nvd.nist.gov/vuln/detail/CVE-2020-36518)  [**CVE-2020-25649**](https://nvd.nist.gov/vuln/detail/CVE-2020-25649) |
| log4j-api-2.12.1.jar | The Apache Log4j API | [**CVE-2021-44832**](https://nvd.nist.gov/vuln/detail/CVE-2021-44832)  [**CVE-2021-45105**](https://nvd.nist.gov/vuln/detail/CVE-2021-45105)  [**CVE-2021-45046**](https://nvd.nist.gov/vuln/detail/CVE-2021-45046)  [**CVE-2021-44228**](https://nvd.nist.gov/vuln/detail/CVE-2021-44228)  [**CVE-2020-9488**](https://nvd.nist.gov/vuln/detail/CVE-2020-9488) |
| logback-core-1.2.3.jar | logback-core module | [**CVE-2021-42550**](https://nvd.nist.gov/vuln/detail/CVE-2021-42550) |
| snakeyaml-1.25.jar | YAML 1.1 parser and emitter for Java | [**CVE-2022-1471**](https://nvd.nist.gov/vuln/detail/CVE-2022-1471)  [**CVE-2022-41854**](https://nvd.nist.gov/vuln/detail/CVE-2022-41854)  [**CVE-2022-38752**](https://nvd.nist.gov/vuln/detail/CVE-2022-38752)  [**CVE-2022-38751**](https://nvd.nist.gov/vuln/detail/CVE-2022-38751)  [**CVE-2022-38750**](https://nvd.nist.gov/vuln/detail/CVE-2022-38750)  [**CVE-2022-38749**](https://nvd.nist.gov/vuln/detail/CVE-2022-38749)  [**CVE-2022-25857**](https://nvd.nist.gov/vuln/detail/CVE-2022-25857)  [**CVE-2017-18640**](https://nvd.nist.gov/vuln/detail/CVE-2017-18640) |
| spring-boot-2.2.4.RELEASE.jar | Spring Boot | [**CVE-2023-20883**](https://nvd.nist.gov/vuln/detail/CVE-2023-20883)  [**CVE-2023-20873**](https://nvd.nist.gov/vuln/detail/CVE-2023-20873)  [**CVE-2022-27772**](https://nvd.nist.gov/vuln/detail/CVE-2022-27772) |
| spring-boot-starter-web-2.2.4.RELEASE.jar | Starter for building web, including RESTful, applications using Spring MVC. Uses Tomcat as the default embedded container | [**CVE-2023-20883**](https://nvd.nist.gov/vuln/detail/CVE-2023-20883)  [**CVE-2023-20873**](https://nvd.nist.gov/vuln/detail/CVE-2023-20873)  [**CVE-2022-27772**](https://nvd.nist.gov/vuln/detail/CVE-2022-27772) |
| spring-core-5.2.3.RELEASE.jar  spring-web-5.2.3.RELEASE.jar  spring-webmvc-5.2.3.RELEASE.jar | Spring Core  Spring Web  Spring Web MVC | [**CVE-2023-20863**](https://nvd.nist.gov/vuln/detail/CVE-2023-20863)  [**CVE-2023-20861**](https://nvd.nist.gov/vuln/detail/CVE-2023-20861)  [**CVE-2022-22971**](https://nvd.nist.gov/vuln/detail/CVE-2022-22971)  [**CVE-2022-22970**](https://nvd.nist.gov/vuln/detail/CVE-2022-22970)  [**CVE-2022-22968**](https://nvd.nist.gov/vuln/detail/CVE-2022-22968)  [**CVE-2022-22965**](https://nvd.nist.gov/vuln/detail/CVE-2022-22965)  [**CVE-2022-22950**](https://nvd.nist.gov/vuln/detail/CVE-2022-22950)  [**CVE-2021-22060**](https://nvd.nist.gov/vuln/detail/CVE-2021-22060)  [**CVE-2021-22096**](https://nvd.nist.gov/vuln/detail/CVE-2021-22096)  [**CVE-2021-22118**](https://nvd.nist.gov/vuln/detail/CVE-2021-22118)  [**CVE-2020-5421**](https://nvd.nist.gov/vuln/detail/CVE-2020-5421)  [**CVE-2016-1000027**](https://nvd.nist.gov/vuln/detail/CVE-2016-1000027) |
| tomcat-embed-core-9.0.30.jar  tomcat-embed-websocket-9.0.30.jar | Core Tomcat implementation | [**CVE-2023-41080**](https://nvd.nist.gov/vuln/detail/CVE-2023-41080)  [**CVE-2023-28708**](https://nvd.nist.gov/vuln/detail/CVE-2023-28708)  [**CVE-2022-42252**](https://nvd.nist.gov/vuln/detail/CVE-2022-42252)  [**CVE-2022-42252**](https://nvd.nist.gov/vuln/detail/CVE-2022-42252)  [**CVE-2021-43980**](https://nvd.nist.gov/vuln/detail/CVE-2021-43980)  [**CVE-2022-34305**](https://nvd.nist.gov/vuln/detail/CVE-2022-34305)  [**CVE-2022-29885**](https://nvd.nist.gov/vuln/detail/CVE-2022-29885)  [**CVE-2021-41079**](https://nvd.nist.gov/vuln/detail/CVE-2021-41079)  [**CVE-2021-33037**](https://nvd.nist.gov/vuln/detail/CVE-2021-33037)  [**CVE-2021-30640**](https://nvd.nist.gov/vuln/detail/CVE-2021-30640)  [**CVE-2021-25329**](https://nvd.nist.gov/vuln/detail/CVE-2021-25329)  [**CVE-2021-25122**](https://nvd.nist.gov/vuln/detail/CVE-2021-25122)  [**CVE-2021-24122**](https://nvd.nist.gov/vuln/detail/CVE-2021-24122)  [**CVE-2020-17527**](https://nvd.nist.gov/vuln/detail/CVE-2020-17527)  [**CVE-2020-13943**](https://nvd.nist.gov/vuln/detail/CVE-2020-13943)  [**CVE-2020-13935**](https://nvd.nist.gov/vuln/detail/CVE-2020-13935)  [**CVE-2020-13934**](https://nvd.nist.gov/vuln/detail/CVE-2020-13934)  [**CVE-2020-8022**](https://nvd.nist.gov/vuln/detail/CVE-2020-8022)  [**CVE-2020-11996**](https://nvd.nist.gov/vuln/detail/CVE-2020-11996)  [**CVE-2020-9484**](https://nvd.nist.gov/vuln/detail/CVE-2020-9484)  [**CVE-2020-1938**](https://nvd.nist.gov/vuln/detail/CVE-2020-1938)  [**CVE-2020-1935**](https://nvd.nist.gov/vuln/detail/CVE-2020-1935)  [**CVE-2019-17569**](https://nvd.nist.gov/vuln/detail/CVE-2019-17569) |

## Mitigation Plan

According to the manual review and static testing report, most of the vulnerabilities are from outdated frameworks. Here is a suggested action plan to decrease the number of vulnerabilities:

**bcprov-jdk15on-1.46.jar**

In the first dependency “bcprov-jdk15on-1.46.jar”, the CVE vulnerabilities are all mostly caused from errors in the Bouncy Castle JCE Provider version 1.55 and would be fixed by upgrading to a more current version.

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

For the “hibernate-validator-6.0.18.Final.jar” dependency and its “CVE-2020-10693” vulnerability, the NVD database provides the mitigation technique of unwrapping the context to HibernateConstraintValidatorContext and passing user input as an expression variable (*CVE-2020-10693 Detail,* 2020).

**jackson-databind-2.10.2.jar**

The Jackson databind and FasterXML Jackson databind need to be updated.

**log4j-api-2.12.1.jar**

The Apache Log4j needs to be updated to a more current version.

**logback-core-1.2.3.jar**

The logback core needs to be updated.

**snakeyaml-1.25.jar**

Using Snakeyaml to parse untrusted YAML files will make the program vulnerable to Denial of Service attacks. The parser should not be used on user supplied input.

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

**spring-boot-starter-web-2.2.4.RELEASE.jar**

These versions of spring boot are out of date and unsupported. An update to a more current version is needed.

**spring-core-5.2.3.RELEASE.jar**

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

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

These versions of spring core, web, and webmvc are out of date and unsupported. An update to a more current version is needed.

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

**tomcat-embed-websocket-9.0.30.jar**

Older versions of tomcat are prone to Denial of Service attacks and data breaches. An update to a more current version is needed.

References

National Vulnerability Database. (2020, May 6). *CVE-2020-10693 Detail*. NVD. <https://nvd.nist.gov/vuln/detail/CVE-2020-10693>