****

# 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 5](#_Toc32574614)

[5. Mitigation Plan 5](#_Toc32574615)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **11/11/22** | **Cristian Chavez** | **Vulnerability Assessment Report Updated** |

## 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

Cristian Chavez

## Interpreting Client Needs

Artemis Financial is a consulting company that works with their clients and helps them with their financial goals in the short and long term. Clients are entrusting their personal data as well as their money with Artemis Financial, so there is an expectation that communications within the company as well as to each client is secure. While we are not completely sure that they have international clients, it would be best to proceed as if they do, since it’ll allow them to securely expand their business in the future if they aren’t international already. Since Artemis consults with clients with retirement and investment plans, there is a lot of capital being moved and stored. This makes them and their clients targets for Phishing attacks, because the attackers would already have an idea of what information they could extract and the financial gain they could receive. Our objective is to modernize their current systems so they can further grow, though as stated already because of they’re a financial company more care might be needed. Open-source libraries are an easy way to add improvements to the system that are already streamlined. Though open-source code is out in the open, it’s not exactly easy to find vulnerabilities in since due there are many developers and contributors that have gone through and helped improve it and fix vulnerabilities. For peace of mind and knowing what is at stake however, it would be best to keep open-source code usage to a minimum.

## Areas of Security

Artemis Financial has a RESTful API, which has the potential to be exploited if not secured properly. Areas of security worth focusing on would be Input Validation, Cryptography, APIs, and Code Quality.

The system needs to have proper input validation to protect it from injection attacks and put the system’s database at risk. Cryptography would be important since Artemis is dealing with important client information, it should be encrypted as an additional layer of security in case, they are breached. The API should be up to date on security measures to ensure there are no data leaks and that the system is protected from Denial-of-Service attacks. Lastly code quality should be as best as possible and can be achieved by implementing secure patterns and following best practice during development.

## Manual Review

In Greeting.java the method “Greeting” accepts a long id and a string content from the users. There is not any input validation present. In DocData.java, the method “getId” returns the id inputted from before, an extra layer of security could be added by verifying the id being returned before doing so.

## Static Testing

[bcprov-jdk15on-1.46.jar](file:///C:\Users\Cristian\eclipse-workspace\CS%20305%20Project%20One%20Code%20Base.zip_expanded\rest-service\target\dependency-check-report.html#l1_991c96a4e31e6c19e2b9136c8955bd423f2dc4c7) / [**CVE-2016-1000338**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000338)

DSA does not fully validate encrypted data, allows for extra elements to be added to allow it to validate when it shouldn’t have.

[hibernate-validator-6.0.18.Final.jar](file:///C:\Users\Cristian\eclipse-workspace\CS%20305%20Project%20One%20Code%20Base.zip_expanded\rest-service\target\dependency-check-report.html#l3_7fd00bcd87e14b6ba66279282ef15efa30dd2492) / [**CVE-2020-10693**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-10693)

A bug allowed invalid EL expressions to be evaluated as if they were valid. Allowed attackers to bypass input sanitation placed by developers.

[jackson-databind-2.10.2.jar](file:///C:\Users\Cristian\eclipse-workspace\CS%20305%20Project%20One%20Code%20Base.zip_expanded\rest-service\target\dependency-check-report.html#l5_0528de95f198afafbcfb0c09d2e43b6e0ea663ec) / [**CVE-2020-25649**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-25649)

Flaw that left system open to XML external entity attacks, data integrity at risk.

[log4j-api-2.12.1.jar](file:///C:\Users\Cristian\eclipse-workspace\CS%20305%20Project%20One%20Code%20Base.zip_expanded\rest-service\target\dependency-check-report.html#l10_a55e6d987f50a515c9260b0451b4fa217dc539cb) / [**CVE-2020-9488**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-9488)

Improper validation allows connection to be intercepted by man-in-the-middle attack.

[logback-core-1.2.3.jar](file:///C:\Users\Cristian\eclipse-workspace\CS%20305%20Project%20One%20Code%20Base.zip_expanded\rest-service\target\dependency-check-report.html#l12_864344400c3d4d92dfeb0a305dc87d953677c03c) / [**CVE-2021-42550**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-42550)

An attacker with needed privileges can edit configuration files to execute code loaded from LDAP servers.

[snakeyaml-1.25.jar](file:///C:\Users\Cristian\eclipse-workspace\CS%20305%20Project%20One%20Code%20Base.zip_expanded\rest-service\target\dependency-check-report.html#l14_8b6e01ef661d8378ae6dd7b511a7f2a33fae1421) / [**CVE-2017-18640**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2017-18640)

Some versions of package are vulnerable to DoS attacks

[spring-boot-2.2.4.RELEASE.jar](file:///C:\Users\Cristian\eclipse-workspace\CS%20305%20Project%20One%20Code%20Base.zip_expanded\rest-service\target\dependency-check-report.html#l15_225a4fd31156c254e3bb92adb42ee8c6de812714) / [**CVE-2022-27772**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-27772)

Version left open to temporary directory hijacking; vulnerability affects only versions no longer supported

[spring-core-5.2.3.RELEASE.jar](file:///C:\Users\Cristian\eclipse-workspace\CS%20305%20Project%20One%20Code%20Base.zip_expanded\rest-service\target\dependency-check-report.html#l16_3734223040040e8c3fecd5faa3ae8a1ed6da146b) / [**CVE-2022-22965**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22965)

An application running on JDK+9 may be vulnerable to remove code execution through data binding

[spring-web-5.2.3.RELEASE.jar](file:///C:\Users\Cristian\eclipse-workspace\CS%20305%20Project%20One%20Code%20Base.zip_expanded\rest-service\target\dependency-check-report.html#l17_dd386a02e40b915ab400a3bf9f586d2dc4c0852c) / [**CVE-2016-1000027**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000027)

Spring framework through 5.3.16 open to remote code execution if used for untrusted data or java deserialization

[tomcat-embed-core-9.0.30.jar](file:///C:\Users\Cristian\eclipse-workspace\CS%20305%20Project%20One%20Code%20Base.zip_expanded\rest-service\target\dependency-check-report.html#l18_ad32909314fe2ba02cec036434c0addd19bcc580) / [**CVE-2020-1938**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-1938)

Exploit possible that allows for remote code execution

[tomcat-embed-websocket-9.0.30.jar](file:///C:\Users\Cristian\eclipse-workspace\CS%20305%20Project%20One%20Code%20Base.zip_expanded\rest-service\target\dependency-check-report.html#l20_33157f6bc5bfd03380ebb5ac476db0600a04168d) / [**CVE-2020-1938**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-1938)

Exploit possible that allows for remote code execution

## Mitigation Plan

While there were some large vulnerabilities found, it’s great that they were found early so they can be properly handled. Enhanced input validation would be the first thing to implement in order to prevent injection attacks as well as to see if there are a better encryption system to keep data safe. Permissions should be used sparingly and be given only to those that need them. In addition, if the Spring Framework is being used, it should be always be a version that is still being supported, along with using the framework for the purposes intended by the developers.