****

# 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** | **[Date]** | **[Your name]** |  |

## 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

Marcus Zambito

## Interpreting Client Needs

Artemis Financial has expressed the need to update its operations and to implement new and effective software security measures to protect its client base. As a consultation and financial planning company, secure communications are necessary for client data protection – this includes savings, finances, insurance, and sensitive personal information. Artemis Financial is not a Global company, requiring less adherence to different global laws. Assuming Artemis Financial is a U.S based company, the most important laws to consider are under The Electronic Communications Privacy Act (ECPA). Current external threats are Blackhat hacker groups, which may use malware attacks, phishing attacks, and SQL injection attacks. Open-source libraries are a great resource, but can provide these groups with information that may allow them to use such attacks if said libraries are used.

## Areas of Security

As a communication and finance-based company, Artemis Financial has several areas to consider:

1. Secure communication. HTTPS protocols can be used to protect data transmission, while employees must be kept up to speed on phishing scams. Keeping clients up to speed and informed on these forms of attack are also imperative.
2. Session Management. Artemis Financials software must consider user sessions security at the highest level- This includes maintenance and termination to prevent attacks from targeting these sessions.
3. Input Validation and encoding. SQL injection is a common form of attack, and proper input validation can lessen the risk of this form of attack working. Output Encoding is also important here.
4. Data protection. This, in my opinion, is the most important area. As a financial advisor and counselor, Artemis Financial will have access to huge amounts of personal data and financial information. A breach or leak of this information can impact thousands- Encryption will help keep this data secure and safe.
5. Authentication. Artemis Financial must inform users of password management and the importance of authorized users.

## Manual Review

No major vulnerabilities jump out in manual review- though I am unable to directly view “main”

## Static Testing

Static testing shows at least 13 Vulnerable dependencies. These include:

[bcprov-jdk15on-1.46.jar](file:///C:\Users\marcc\eclipse-workspace\JavaWorkspace\rest-service\target\dependency-check-report.html#l2_991c96a4e31e6c19e2b9136c8955bd423f2dc4c7)- Some versions are vulnerable to timing side-channel attacks, plaintext recovery attacks, and crafted elliptic curve key exchanges. Documentation for these vulnerabilities was provided by SHA1

[spring-boot-2.2.4.RELEASE.jar](file:///C:\Users\marcc\eclipse-workspace\JavaWorkspace\rest-service\target\dependency-check-report.html#l3_225a4fd31156c254e3bb92adb42ee8c6de812714)- Unable to detect directory Hijacking in some versions. This is no longer an issue, but should be considered.

[logback-core-1.2.3.jar](file:///C:\Users\marcc\eclipse-workspace\JavaWorkspace\rest-service\target\dependency-check-report.html#l4_864344400c3d4d92dfeb0a305dc87d953677c03c)- versions 1.2.7 and below allowed an attacker with required privileges to edit config files. Be sure to keep up to date.

[log4j-api-2.12.1.jar](file:///C:\Users\marcc\eclipse-workspace\JavaWorkspace\rest-service\target\dependency-check-report.html#l5_a55e6d987f50a515c9260b0451b4fa217dc539cb)-improper validation allowed “man in the middle” attacks to leak log messages. Fixxed with apache log4j 2.12.3

[snakeyaml-1.25.jar](file:///C:\Users\marcc\eclipse-workspace\JavaWorkspace\rest-service\target\dependency-check-report.html#l8_8b6e01ef661d8378ae6dd7b511a7f2a33fae1421)- Versions below 1.26 allow entity expansion during loading. Keep up to date

[jackson-databind-2.10.2.jar](file:///C:\Users\marcc\eclipse-workspace\JavaWorkspace\rest-service\target\dependency-check-report.html#l9_0528de95f198afafbcfb0c09d2e43b6e0ea663ec)- A flaw in FasterXML Databind allowed entity expansion, which created vulnerabilities to XXE attacks. Disabling this function should be enough to prevent this flaw.

[tomcat-embed-core-9.0.30.jar](file:///C:\Users\marcc\eclipse-workspace\JavaWorkspace\rest-service\target\dependency-check-report.html#l13_ad32909314fe2ba02cec036434c0addd19bcc580)- Unlikely issue involved with reverse proxy Transfer-encoding. Highly unlikely, simply do not try to replicate.

[hibernate-validator-6.0.18.Final.jar](file:///C:\Users\marcc\eclipse-workspace\JavaWorkspace\rest-service\target\dependency-check-report.html#l16_7fd00bcd87e14b6ba66279282ef15efa30dd2492)- Hibernate Validator in version 6.1.2 allowed attackers to bypass input sanitation. Recommendation to not use.

[spring-web-5.2.3.RELEASE.jar](file:///C:\Users\marcc\eclipse-workspace\JavaWorkspace\rest-service\target\dependency-check-report.html#l19_dd386a02e40b915ab400a3bf9f586d2dc4c0852c)-Remote code execution issue if used for Java deserialization. Do not use with untrusted data, according to vendor.

[spring-beans-5.2.3.RELEASE.jar](file:///C:\Users\marcc\eclipse-workspace\JavaWorkspace\rest-service\target\dependency-check-report.html#l20_0250c8c641433dc06b1b44e4563fa08a2fbf8954)-Older versions susceptible to malicious input from users. Keep up to date according to Vendor releases.

[spring-context-5.2.3.RELEASE.jar](file:///C:\Users\marcc\eclipse-workspace\JavaWorkspace\rest-service\target\dependency-check-report.html#l22_7750c95c96c7a1885c8b1b503ba915bc33ca579a)- Older versions allow for characters to be placed in disallowed fields. Recommended to keep up to date according to Vendor releases.

[spring-expression-5.2.3.RELEASE.jar](file:///C:\Users\marcc\eclipse-workspace\JavaWorkspace\rest-service\target\dependency-check-report.html#l23_d0c6bb10758805b2153c589686b8045554bfac2d)-Denial of service issues with unsupported versions. As before, keep up to date according to Vendor releases.

## Mitigation Plan

Mitigation solutions are listed alongside vulnerabilities.