****

# 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** | **3/19/2023** | **Brody Robinson** |  |

## Client



## Developer

Brody Robinson

## Interpreting Client Needs

1. The value of secure for communications for this company should be of the utmost importance. Given the type of clients of they serve, the data they safeguard is sensitive. They will want to prevent all unwanted access and keep their confidentiality and integrity intact.
2. Being a web-based company, Artemis Financial will have international transactions. This indicates the need for following rules and regulations in regards to international trading.
3. There will be governmental restrictions regarding secure communications, and each will need to be considered, given that there will be many different forms as the company deals with money, stocks, savings and retirements. These different entities open the doors to all types of communications where each will likely be address with certain governmental restrictions.
4. The threats that are in abundance now, will also be in abundance in the future. In fact, in may be even worse. These would include threats such as the amount of hackers. Population is growing, and so is the knowledge of those in the cyber world, good and bad. These threats will also include technological advancement that empower these hackers to find and exploit the vulnerabilities in any industry or company they target.
5. The role of open-source libraries is a modernization requirement that can provide much help to Artemis Financial. From an open-source library, the company can add multiple layers of security and functionality to their program, adding more obstacles to make it difficult for hackers, thus reducing the risk of major attacks.

## Areas of Security

* API:

An Application Programming Interface, or API, enables the communication of two or more applications. As Artemis Financial uses RESTful APIs, we need to ensure that the relationship between the applications is secure and protected. A tool to be used here will be Open Authorization, or OAuth, which will allow users access to certain web resources without the need for sharing passwords or login credentials.

* Cryptography:

Cryptography is the means for transmitting and accessing data via the internet. It also provides the scientific means for converting data into a secure format. This is encryption. If this process is not followed, or not followed correctly, the confidentiality and integrity of the program can quickly become compromised. The use of cryptography will allow the proper communication between parties in a trusted manner to reduce the risk of vulnerabilities for all.

* Client/Server:

This is the communication between the clients and servers. When the client sends a request to the server and the server delivers, we need to ensure the information that is transferred back and forth is secured to prevent any discrepancies and to prevent a portal for possible attacks.

* Code Quality:

Code quality applies to the security of the application in multiple factors. By sanitizing the code, we can decrease the amount of ways that the system can be exploited. The excess of code, in terms that it is of poor quality, can present and bring about vulnerability by error and/or mistake. Thus, the code should be written/revised of great quality with security in mind.

## Manual Review

* The program does not use HTTPS. When working with sensitive data and information, this is recommended.
* Requests are not validated. This renders a vulnerability to an attack on the system.
* In regard to verifications, there is no authentication system in place.
* Within the CRUDController class, business names are sent as request parameters. This creates a vulnerability by allowing information to be accessed by unauthorized entities.

## Static Testing

bcprov-jdk15on-1.46.jar (CVE-2016-1000338)

In Bouncy Castle JCE Provider version 1.55 and earlier the DSA does not fully validate ASN.1 encoding of signature on verification.

hibernate-validator-6.0.18.Final.jar

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.

jackson-databind-2.10.2.jar

A flaw was found in FasterXML Jackson Databind, where it did not have entity expansion secured properly.

log4j-api-2.12.1.jar

Improper validation of certificate with host mismatch in Apache Log4j SMTP appender.

logback-core-1.2.3.jar

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.

snakeyaml-1.25.jar

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. It is recommended to use SnakeYaml's SafeConsturctor when parsing untrusted content to restrict deserialization.

spring-boot-2.2.4.RELEASE.jar

\*\* UNSUPPORTED WHEN ASSIGNED \*\* 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.

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

\*\* UNSUPPORTED WHEN ASSIGNED \*\* 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.

spring-core-5.2.3.RELEASE.jar

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-web-5.2.3.RELEASE.jar

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.

spring-webmvc-5.2.3.RELEASE.jar

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.

tomcat-embed-core-9.0.30.jar

When using the Apache JServ Protocol (AJP), care must be taken when trusting incoming connections to Apache Tomcat. Tomcat treats AJP connections as having higher trust than, for example, a similar HTTP connection. If such connections are available to an attacker, they can be exploited in ways that may be surprising.

tomcat-embed-websocket-9.0.30.jar

When using the Apache JServ Protocol (AJP), care must be taken when trusting incoming connections to Apache Tomcat. Tomcat treats AJP connections as having higher trust than, for example, a similar HTTP connection. If such connections are available to an attacker, they can be exploited in ways that may be surprising.

## Mitigation Plan

* Update all dependencies listed above with the latest versions, respectively, to ensure that we are using the most up to date products and taking advantage of the security upgrades.
* Move all direct communications to HTTPS. This will increase the security of the company and client’s information by preventing the data from being viewed by an outside entity.
* Deploy a standard for stronger and more complex alphanumerical combination for credentials of the users.
* Deploy two-factor system for authentication to secure sensitive data of that of the company and the clients.
* Revise and remove any instance where any identities are present in the hard-code database.
* Ensure the quality of the coding has been developed and/or revised to fit modern standards for security practices.