****

# Artemis Financial Vulnerability Assessment Report

Table of Contents

[Artemis Financial Vulnerability Assessment Report 1](#_Toc770000428)

[Document Revision History 2](#_Toc1082694960)

[Client 3](#_Toc1507556973)

[Instructions 3](#_Toc2024018442)

[Developer 3](#_Toc1430057416)

[1. Interpreting Client Needs 4](#_Toc916960538)

[2. Areas of Security 4](#_Toc1550324078)

[3. Manual Review 4](#_Toc498402392)

[4. Static Testing 4](#_Toc2117924537)

[5. Mitigation Plan 7](#_Toc24537497)

[References 7](#_Toc1762669568)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **07/10/2023** | **Dominic Drury** | **Initial documentation** |

## 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

Dominic Drury

## Interpreting Client Needs

Artemis Financial is a consulting company that develops individualized financial plans for their customers. These plans include savings, retirements, investments, and insurance. Considering the company's primary data is the financial information of their clients, which could also include relevant identity information like copies of IDs, social security numbers, birth certificates, death certificates, marriage licenses, etc., secure communication is paramount.

Artemis Financial does not specify that they are a worldwide organization, but with the global market and commonplace of international transactions, it would be safe to assume that they do have some international transactions. Further questioning would be recommended.

For this project there are several governmental restrictions about secure communications that need to be considered, especially if we are assuming that Artemis Financial operates worldwide. The is the mandatory EU-GDPR for operating in the European Union, the UK’s UK-GDPR which is similar to the EU-GDPR but does contain some differences for domestic law, since Artemis Financial is a public US organization it is expected to comply with the mandatory SOX act, the PCI DSS if Artemis Financial uses credit card information, the US mandatory BSA and GLBA, the PSD 2 in the EU, and the international mandatory FFIEC. (Kost, 2023)

External threats that may be present now, or in the immediate future, are primarily going to be unauthorized data access. With all the highly sensitive data that the company will be storing it will be a highly sought after target for criminals seeking to obtain SSN, account and routing numbers, and people who have the funds available to target for fraud scams.

The modernization requirements we must consider are maintaining encryption algorithms to ensure that the data being communicated is secure, safe anti phishing practices for all employees and potentially their clients, regular security checks either from a dedicated department or reliable third party, regular maintenance checks for the application to prevent bugs, glitches, and exploits, and account security like password strength and password change frequency for all employees and potentially clients.

## Areas of Security

Based on the client needs as mentioned above the areas of security that should be focused on by Artemis Financial are input validation, APIs, cryptography, code error, and code quality. Input validation to protect against SQL injections, APIs since Artemis Financial uses a RESTful web API, cryptography to ensure that the communications being sent and received are properly encoded and protected, code error to undermine anyone trying to use error handling as a means of bypassing security protocol or using error handling as an opening for an attack, and code quality to maintain the application.

## Manual Review

After a careful manual review of the files that were sent, I found that there is improper input validation in GreetingController.java. The Greeting greeting object being created uses a string obtained from the user, this can be an opening for SQL injections and many other attacks. The same problem appears again in the CRUDController.java

## Static Testing

Ater running a dependency check using the Maven-Dependency-Check Plug-in I found the following issues and charted them in the table below:

|  |  |  |  |
| --- | --- | --- | --- |
| Dependency | Vulnerability ID | Severity | Comments |
| log4j-api-2.12.1.jar | cpe:2.3:a:apache:log4j:2.12.1 | Critical | vulnerable to a 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. |
| tomcat-embed-core-9.0.30.jar | cpe:2.3:a:apache:tomcat:9.0.30 | Critical | When using the RemoteIpFilter with requests received from a reverse proxy via HTTP that include the X-Forwarded-Proto header set to https, session cookies created by Apache Tomcat 11.0.0-M1 to 11.0.0.-M2, 10.1.0-M1 to 10.1.5, 9.0.0-M1 to 9.0.71 and 8.5.0 to 8.5.85 did not include the secure attribute. This could result in the user agent transmitting the session cookie over an insecure channel. |
| cpe:2.3:a:apache\_tomcat:apache\_tomcat:9.0.3 | No information available |
| snakeyaml-1.25.jar | cpe:2.3:a:snakeyaml\_project:snakeyaml:1.25 | Critical | 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. |
| cpe:2.3:a:yaml\_project:yaml:1.25 | Uncaught Exception in GitHub repository eemeli/yaml prior to 2.0.0-5. |
| spring-beans-5.2.3.RELEASE.jar | cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release | High | No information available |
| cpe:2.3:a:springsource:spring\_framework:5.2.3:release | No information available |
| spring-web-5.2.3.RELEASE.jar | cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release | High | No information available |
| cpe:2.3:a:springsource:spring\_framework:5.2.3:release | No information available |
| spring-boot-2.2.4.RELEASE.jar | cpe:2.3:a:vmware:spring\_boot:2.2.4:release | High | 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. |
| jackson-databind-2.10.2.jar | cpe:2.3:a:fasterxml:jackson-databind:2.10.2 | High | \*\* DISPUTED \*\* An issue was discovered jackson-databind thru 2.15.2 allows attackers to cause a denial of service or other unspecified impacts via crafted object that uses cyclic dependencies. NOTE: the vendor's perspective is that the product is not intended for use with untrusted input. |
| bcprov-jdk15on-1.46.jar | cpe:2.3:a:bouncycastle:legion-of-the-bouncy-castle-java-crytography-api:1.46 | High | 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. |
| logback-core-1.2.3.jar | cpe:2.3:a:qos:logback:1.2.3 | Medium | 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. |
| spring-expression-5.2.3.RELEASE.jar | cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release | Medium | No information available |
| cpe:2.3:a:springsource:spring\_framework:5.2.3:release | No information available |
| spring-context-5.2.3.RELEASE.jar | cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release | Medium | No information available |
| cpe:2.3:a:springsource:spring\_framework:5.2.3:release | No information available |
| hibernate-validator-6.0.18.Final.jar | cpe:2.3:a:redhat:hibernate\_validator:6.0.18 | Medium | 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. |
| spring-webmvc-5.2.3.RELEASE.jar | cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release | Medium | No information available |
| cpe:2.3:a:springsource:spring\_framework:5.2.3:release | No information available |

## Mitigation Plan

After carefully reviewing the security vulnerabilities documented here on this vulnerability assessment report, I would suggest the following solutions:

* Ensure that all untrusted information is not sent directly to the server as a string
* limit JNDI data source names to the java protocol in Log4j2 versions 2.17.1, 2.12.4, and 2.3.2
* upgrade snkeyaml to version 2.0 and beyond
* ensure that all dependencies are as up to date as possible

References

Kost, E. (2023, June 27). Top 8 Cybersecurity Regulations for Financial Services: Upguard. RSS. <https://www.upguard.com/blog/cybersecurity-regulations-financial-industry>