****

# 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** | **11/9/23** | **Cooper Brien** |  |

## Client



## Developer

Cooper Brien

## Interpreting Client Needs

Artemis Financial will need a great deal of cyber security to protect its customers’ information. The company works with customers to develop individualized financial plans. To organize a financial plan, personal financial information is required to maximize the effectiveness of these plans. This information could include a customer’s income, expenditures, banking information, etc. It is important to develop strong security measures to prevent any possibility of threats intercepting this sensitive information.

## Areas of Security

Regarding areas of security, the areas that this application should be focused on are Input Validation, Secure APIs, Cryptography, and Client/Server

Input validation is important to make sure that user inputs are valid and correct. This can also be extended to securing logins and using multi-factor authentication to protect the sensitivity of the data that will be distributed in this application.

The application is using an API and it will be important to tightly secure these interactions with the API.

It is important to encrypt data that is being transported between the client and the server as an extra layer of protection. If a hacker were to break through and intercept data, the encryption would make it useless to them unless they knew how to decrypt it.

Securing communications between the client and the server is also very important. Hackers can intercept sensitive data that travels between the client and the server, so it is important to make sure that there are secure boundaries put in place to prevent this.

It is also always important to maintain secure coding practices and patterns. Unsecure code can be exploited or can damage the applications efficiency and performance.

## Manual Review

In the code there were multiple instances of cryptography errors. This can be dangerous because if information is intercepted, encrypted data prevents it from being interpreted by interceptor. This would make the data that they intercepted useless unless they could decrypt it. There were also some problems involving input validation. Some inputs were being invalidated improperly. There were some poor coding structures including instances of data leaks and infinite loops.

## Static Testing

A screenshot of a computer

Description automatically generated

**Bcprov-jdk15on-1.46**

* CVE-2013-1624
  + CWE-310: Cryptography Issues
  + SOL: Update the library to fix these security issues
* CVE-2015-6644
  + CWE-200: Information Exposure
  + SOL: Update the library to fix these security issues
* CVE-2015-7940
  + Information Exposure
  + SOL: Implement patch provided by NVD
* CVE-2016-1000338
  + CWE-347: Improper Verification of Cryptographic Signature
  + SOL: Implement patch provided by NVD
* CVE-2016-1000339
  + CWE-310: Cryptographic Issues
  + SOL: Implement patch provided by NVD
* CVE-2016-1000341
  + 7PK - Time and State
  + SOL: Implement patch provided by found in the NVD
* CVE-2016-1000342
  + Improper Verification of Cryptographic Signature
  + SOL: Implement patch provided by NVD
* CVE-2016-1000343
  + Cryptographic Issues
  + SOL: Implement patch provided by NVD
* CVE-2016-1000344
  + Cryptographic Issues
  + SOL: Implement patch provided by NVD
* CVE-2016-1000345
  + CWE-361: 7PK- Time and State
  + SOL: Implement patch provided by NVD
* CVE-2016-1000346
  + CWE-320: Key Management Errors
  + SOL: Implement patch provided by NVD
* CVE-2016-1000352
  + Cryptographic Issues
  + SOL: Implement patch provided by NVD
* CVE-2017-13098
  + CWE-203: Information Exposure Through Discrepancy
  + SOL: Implement patch provided by NVD
* CVE-2018-5382
  + CWE-354: Improper Validation of Integrity Check Value
  + SOL: Implement more robust keystore format.
* CVE-2020-0187
  + Cryptographic Issues
  + SOL: Implement patch provided by NVD
* CVE-2020-26939
  + Information Exposure Through Discrepancy
  + SOL: Implement patch provided by NVD
* CVE-2023-33201
  + CWE-203: Improper Certificate Validation
  + SOL: Implement patch provided by NVD

**Spring-boot-2.2.4.RELEASE.jar**

* CVE-2022-27772
  + CWE-668: Exposure of Resource to Wrong Sphere
  + SOL: Implement patch provided by NVD
* CVE-2023-20873
  + SOL: Update spring-boot
* CVE-2023-20883
  + CWE-400: Uncontrolled Resource Consumption
  + SOL: Update spring-boot

**Logback-core-1.2.3.jar**

* CVE-2021-42550
  + CWE-502 Deserialization of Untrusted Data
  + SOL: Implement patch provided by NVD

**Log4j-api-2.12.1.jar**

* CVE-2020-9488
  + CWE-295 Improper Certificate Validation
  + SOL: Implement patch provided by NVD
* CVE-2021-44228
  + Uncontrolled Resource Consumption
  + CWE-502 Deserialization of Untrusted Data
  + CWE-20 Improper Input Validation
  + SOL: Implement patch provided by NVD
* CVE-2021-44832
  + Improper Input Validation
  + SOL: Implement patch provided by NVD
* CVE-2021-45046
  + CWE-917 Improper Neutralization of Special Elements used in Expression Language Statement
  + SOL: Implement patch provided by NVD
* CVE-2021-45105
  + Improper Input Validation
  + CWE-674 Uncontrolled Recursion
  + SOL: Implement patch provided by NVD

**Snakeyaml-1.25.jar**

* CVE-2017-18640
  + CWE-776 Improper Restriction of Recursive Entity References in DTDs
  + SOL: Implement patch provided by NVD
* CVE-2021-4235
  + SOL: Implement patch provided by NVD
* CVE-2022-1471
  + Deserialization of Untrusted Data
  + No concrete solution yet found, suggests using an alternative to Snakeyaml or updating Snakeyaml to fix this
* CVE-2022-25857
  + Improper Restriction of Recursive Entity References DTDs
  + SOL: Implement patch provided by NVD
* CVE-2022-3064
  + Uncontrolled Resource Consumption
  + SOL: Implement patch provided by NVD
* CVE-2022-38749
  + CWE-787 Out-of-bounds Write
  + No concrete solution yet found, suggests using an alternative to Snakeyaml or updating Snakeyaml to fix this
* CVE-2022-38750
  + Out-of-bounds Write
  + No concrete solution yet found, suggests using an alternative to Snakeyaml or updating Snakeyaml to fix this
* CVE-2022-38751
  + Out-of-bounds Write
  + No concrete solution yet found, suggests using an alternative to Snakeyaml or updating Snakeyaml to fix this
* CVE-2022-38752
  + Out-of-bounds Write
  + No concrete solution yet found, suggests using an alternative to Snakeyaml or updating Snakeyaml to fix this
* CVE-2022-41854
  + Out-of-bounds Write
  + No concrete solution yet found, suggests using an alternative to Snakeyaml or updating Snakeyaml to fix this

**Jackson-databind-2.10.2.jar**

* CVE-2020-25649
  + CWE-611 Improper Restriction of XML External Entity Reference
  + SOL: Implement patch provided by NVD
* CVE-2020-36518
  + Out-of-bounds Write
  + SOL: Implement patch provided by NVD
* CVE-2021-46877
  + CWE-770 Allocation of Resources without limits or throttling
  + SOL: Implement patch provided by NVD
* CVE-2022-42003
  + Deserialization of Untrusted Data
  + SOL: Implement patch provided by NVD
* CVE-2022-42004
  + Deserialization of Untrusted Data
  + SOL: Implement patch provided by NVD
* CVE-2023-35116
  + Allocation of resources without limits or throttling
  + SOL: Implement patch provided by NVD

**Tomcat-embed-core-9.0.30**

* CVE-2019-17569
  + CWE-444 Inconsistent Interpretation of HTTP Requests
  + SOL: Implement patch provided by NVD
* CVE-2020-11996
  + SOL: Implement patch provided by NVD
* CVE-2020-13934
  + CWE-401 Improper Release of Memory Before removing last reference (‘memory leak’)
  + CWE-476 NULL Pointer dereference
  + SOL: Implement patch provided by NVD
* CVE-2020-13935
  + CWE-835 Loop with unreachable exit condition
  + SOL: Implement patch provided by NVD
* CVE-2020-13943
  + SOL: Implement patch provided by NVD
* CVE-2020-17527
  + Information Exposure
  + SOL: Implement patch provided by NVD
* CVE-2020-1935
  + Inconsistent Interpretation of HTTP requests
  + SOL: Implement patch provided by NVD
* CVE-2020-1938
  + SOL: Implement patch provided by NVD
* CVE-2020-8022
  + CWE-276 Incorrect Default Permissions
  + SOL: Implement patch provided by NVD
* CVE-2020-9484
  + Deserialization of Untrusted Data
  + SOL: Implement patch provided by NVD
* CVE-2021-24122
  + CWE-706 Use of Incorrectly resolved name or reference.
  + SOL: Implement patch provided by NVD
* CVE-2021-25122
  + Information Exposure
  + SOL: Implement patch provided by NVD
* CVE-2021-25329
  + SOL: Implement patch provided by NVD
* CVE-2021-30640
  + CWE-116 Improper Encoding or Escaping of output
  + SOL: Implement patch provided by NVD
* CVE-2021-33037
  + Inconsistent interpretation of HTTP requests.
  + SOL: Implement patch provided by NVD
* CVE-2021-41079
  + Infinite Loop
  + SOL: Update Tomcat to latest version
* CVE-2021-43980
  + CWE-362 Concurrent Execution using shared resource with improper synchronization.
  + SOL: Update Tomcat to latest version
* CVE-2022-29885
  + Uncontrolled resource consumption
  + SOL: Implement patch provided by NVD
* CVE-2022-34305
  + CVE-79 Improper Neutralization of Input During Web page generation
  + SOL: Update Tomcat to latest version
* CVE-2022-42252
  + Inconsistent Interpretation of HTTP Requests
  + SOL: Update Tomcat to latest version
* CVE-2023-28708
  + CWE-523 Unprotected Transport of Credentials
  + SOL: Update Tomcat to latest version
* CVE-2023-41080
  + CWE-601 URL redirection to untrusted site
  + SOL: Implement patch provided by NVD
* CVE-2023-42795
  + CWE-459 Incomplete Cleanup
  + SOL: Update Tomcat to latest version
* CVE-2023-44487
  + Uncontrolled Resource Consumption
  + SOL: Implement patch provided by NVD
* CVE-2023-45648
  + Improper Input Validation
  + SOL: Update Tomcat to latest version

Hibernate-validator-6.0.18.Final.jar

* CVE-2020-10693
  + Improper Input Validation
  + SOL: Implement patch provided by NVD

Spring-web-5.2.3.RELEASE.jar

* CVE-2016-1000027
  + Deserialization of Untrusted Data
  + SOL: Update Spring Framework
* CVE-2021-22096
  + CWE-117 Improper Output of Neutralization for logs
  + SOL: Update Spring Framework
* CVE-2021-22118
  + Exposure of Resource to Wrong Sphere
  + SOL: Implement patch provided by NVD

Spring-beans-5.2.3.RELEASE.jar

* CVE-2022-22965
  + Improper Control of Generation of Code (Code Injection)
  + SOL: Update Spring Framework

spring-webmvc-5.2.3.RELEASE.jar

* CVE-2021-22060
  + CWE-117 Improper Output Neutralization for Logs
  + SOL: Implement patch provided by NVD

Spring-context-5.2.3.RELEASE.jar

* CVE-2022-22968
  + CWE-178 Improper Handling of Case Sensitivity
  + SOL: Implement patch provided by NVD

Spring-expression-5.2.3.RELEASE.jar

* CVE-2022-22950
  + Allocation of resources without limits or throttling
  + SOL: Update Spring Framework
* CVE-2023-20861
  + SOL: Update Spring Framework
* CVE-2023-20863
  + Improper Neutralization of special elements used in expression language statement
  + SOL: Update Spring Framework

## Mitigation Plan

The first step in mitigating the vulnerabilities above would be to update the libraries and dependencies that the application is using. Many of the solutions to the CVE codes discovered from the dependency check can be solved by using the latest updates of the dependencies. Secondly, I would make sure that the development team employs safe and secure coding practices. I would go through the code and fix the vulnerabilities that are returning infinite loops and causing data leaks. Finally, I would go through any other of the present vulnerabilities that were not fixed during the first two steps. Many of these vulnerabilities will need certain patches to prevent these vulnerabilities. Luckily, there are sometimes multiple patches for each vulnerability that we detected in the vulnerability report. Many of these solutions can be found on the NVD website which offers multiple possible solutions to these vulnerabilities.