****

# 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** | **1/20/2023** | **Nick Franklin** |  |

## 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

Nick Franklin

## Interpreting Client Needs

* **Client summary and needs.**

The client is Artemis Financial who is a consulting company that develops individualized financial plans for their customers including savings, retirement, investments, and insurance. As part of their success with their custom software, they would also like the most current and effective software security to protect their organization from external threats.

* **What is the value of secure communications to the company?**

Because Artemis Financial deals with not only customers funds but also with their personal information such as tax information, Social Security numbers, etc., secure communications are of the highest value. Since our task is to assess their web-based software application and identify any security vulnerabilities, all possible avenues must be checked thoroughly as security is the task here and thus the highest priority.

* **Does the company make any international transactions?**

Although there is no mention in the briefing of whether Artemis Financial is global or solely domestic, they will be working with customers that may well have international funds or transactions so we will work under the pretenses that the company will be making international transactions.

* **Are there governmental restrictions about secure communications to consider?**

Other than the proper use and security of customers’ personal data, there are not any government restrictions that we need to account for that I know of.

* **What external threats might be present now and in the immediate future?**

Many external threats exist such as hackers, malware, and other types of virtual attacks and malicious programs that seek to obtain client or business data or funds. Besides these external threats, there is also the internal threats like employees committing theft or exposing data whether on accident or with intent. Just as software and technology are constantly evolving, so are the threats and so must security.

* **What are the modernization requirements that you must consider?**

Through the use of open-source libraries and the evolving web applications technologies, security can be handled fairly well. Simply by keeping all systems up to date and current, they should be able to fend off most attacks. Because of open-sourced libraries and the shared cumulative knowledge that they contain, especially that pertaining to security, as new threats come to be and evolve, new updated security methods are made available to combat them. Also, the use of software that can check for security vulnerabilities can greatly improve the process of creating a secure application.

## Areas of Security

* **Input Validation :** This applies to the client’s needs because without proper input validation, a client could use input to infiltrate or disrupt the system. This important for any only application that allows for user input.
* **API’s :** Because the client has an online application, remote access to the system and it’s functionalities is required and allows for user inputs. This means that a REST API to prevent any unwanted access to the client’s systems and have more input validation through the API.
* **Cryptography :** Proper encryption is necessity when transferring data, especially sensitive data.
* **Client/Server :** Validating certifications when transferring data is also a necessity in order to keep data secure and properly handled.
* **Code Error :** Secure error handling is important for inputs and API to keep the system running and needs necessary maintenance and review.
* **Encapsulation :** This is necessary in order to keep different functionality separate and safe. Data needs to be kept secure and only accessed under the proper function and with proper authority.

## Manual Review

I first noticed a flaw in the DocData.java file because I was drawn their first due to the alert about an unused variable in the file. This is where I noticed that it is accessing the data using it’s location and the username and password which can expose the system to unwanted access or data theft. There is also no input validation or authentication processes to ensure the proper access. The CRUDController.java class also has the name in the request which is vulnerable.

## Static Testing

* **Bcprov-jdk15on-1.46.jar :** Needs updating or is vulnerable to injections, weak private keys, ECB mode, timing attacks, padding oracle attacks, local information disclosure, and information leaks. **Codes :** **CVE-2016-1000338, CVE-2016-1000342, CVE-2016-1000343, CVE-2016-1000344, CVE-2016-1000352, CVE-2016-1000341, CVE-2016-1000345, CVE-2017-13098, CVE-2020-15522, CVE-2020-0187 (OSSINDEX), CVE-2016-1000339, CVE-2020-26939 (OSSINDEX), CVE-2015-7940, CVE-2018-5382, CVE-2013-1624, CVE-2016-1000346, CVE-2015-6644 (OSSINDEX)**
* **Hibernate-validator-6.0.18.Final.jar :** Bug in the message interpolation processor enables invalid EL expressions to be evaluated as if they were valid. Allows attackers to bypass input sanitation. **Code : CVE-2020-10693**
* **Jackson-databind-2.10.2.jar :** Flaw where it did not have entity expansion secured properly. Needs to be updated. Allows vulnerability to XML external entity attacks, Java StackOverflow exception, DoS, and resource exhaustion. **Codes : CVE-2020-25649, CVE-2020-36518, CVE-2022-42003, CVE-2022-42004**
* **Log4j-api-2.12.1.jar :** Improper validation of certificate with host mismatch in Apache. Allows connection to be intercepted. Needs update. **Code : CVE-2020-9488**
* **Logback-core-1.2.3.jar :** Needs to be updated. Attacker with privileges to edit configurations files could craft malicious configurations. **Code : CVE-2021-42550**
* **Snakeyaml-1.25.jar :** Constructor class does not restrict types. Use SnakeYaml’s SafeConstructor when parsing untrusted content to restrict deserialization. **Code : CVE-2022-1471** Vulnerable to DoS due to missing nested depth limitation for collections. **Code : CVE-2022-25857** Parsing untrusted YAML files gives vulnerabilities to DoS and other attacks. **Codes : CVE-2022-38749, CVE-2022-38751, CVE-2022-38752, CVE-2022-41854, CVE-2022-38750**
* **Spring-boot-2.2.4.RELEASE.jar :** Needs update to avoid hijacking. **Code : CVE-2022-27772**
* **Spring-boot-starter-web-2.2.4.RELEASE.jar :** Needs update to avoid hijacking. **Code : CVE-2022-27772**
* **Spring-core-5.2.3.RELEASE.jar :** Needs update for vulnerabilities. **Codes : CVE-2022-22965, CVE-2021-22118, CVE-2020-5421,** **CVE-2022-22950, CVE-2022-22971, CVE-2022-22968, CVE-2022-22970, CVE-2021-22060, CVE-2021-22096**
* **Spring-web-5.2.3.RELEASE.jar :** Needs update or suffers from potential remote code execution issue, privilege escalation, RFD attacks, DoS, and others. **Codes : CVE-2016-1000027, CVE-2022-22965, CVE-2021-22118, CVE-2020-5421, CVE-2022-22950, CVE-2022-22971, CVE-2022-22968, CVE-2022-22970, CVE-2021-22060, CVE-2021-22096**
* **Spring-webmvc-5.2.3.RELEASE.jar :** Needs update. **Codes : CVE-2022-22965, CVE-2021-22118, CVE-2020-5421, CVE-2022-22950, CVE-2022-22971, CVE-2022-22968, CVE-2022-22970, CVE-2021-22060, CVE-2021-22096**
* **Tomcat-embed-core-9.0.30.jar :** Needs update for vulnerabilities. **Codes : CVE-2020-1938, CVE-2020-11996, CVE-2020-13934, CVE-2020-13935,** **CVE-2020-17527, CVE-2021-25122, CVE-2021-41079, CVE-2022-29885, CVE-2022-42252, CVE-2020-9484, CVE-2021-25329, CVE-2021-30640, CVE-2022-34305, CVE-2021-24122, CVE-2021-33037, CVE-2019-17569, CVE-2020-1935, CVE-2020-13943, CVE-2021-43980**
* **Tomcat-embed-websocket-9.0.30.jar :** Needs update for vulnerabilities. **Codes : CVE-2020-1938, CVE-2020-8022, CVE-2020-11996, CVE-2020-13934, CVE-2020-13935, CVE-2020-17527, CVE-2021-25122, CVE-2021-41079, CVE-2022-29885, CVE-2022-42252, CVE-2020-9484, CVE-2021-25329, CVE-2021-30640, CVE-2022-34305, CVE-2021-24122, CVE-2021-33037, CVE-2019-17569, CVE-2020-1935, CVE-2020-13943, CVE-2021-43980**

## Mitigation Plan

First off, getting and keeping everything up-to-date is very important and will fix many security risks and vulnerabilities. That is based off of the static testing and the many results that would be solved by updating the software in use and dependencies. All parameters for calls or requests should be changed in order to keep data and locations safe. Authentication should be implemented for added security and ensuring proper credentials for access. Also, there should be input validation and parameters at all levels in order to prevent any malicious input.