****

# 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** | **September 15, 2022** | **Joshua Perez** |  |

## 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

Joshua Perez

## Interpreting Client Needs

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

The value of secure communications is always critical, no matter the industry. However, in this case, when one considers the industry of Artemis Financial, this becomes especially important. Financial companies of any sort must be at the forefront of the cyber security game, specifically when it comes to communications, transactions, and data. Since they are a consulting company, they likely have lots and lots of information pertaining to their client’s personal information.

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

The company focuses on consulting, and while it is not specified that they make international transactions, it can be assumed that during the course of investing for their customers, they will likely need to make transactions internationally at some point. We can also consider that their customers may be located worldwide and want to access their information.

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

While I don’t believe there is an end all, be all, federal law regarding secure communications in an organization, certain patchwork laws may apply such as the Electronic Communications Privacy Act of 1986. This act “protects wire, oral, and electronic communications while those communications are being made, are in transit, and when they are stored on computers. The Act applies to email, telephone conversations, and data stored electronically.” as stated on the DOJ website. This law should be considered when consulting for Artemis Financial.

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

The external threats that may be present now and in the future are the many greedy people that wish to steal from the corporation and their clients. There is a lot of money to be made from breaking into these systems and stealing personal information and sometimes even actual money.

1. **What are the modernization requirements that you must consider? For example:**
   * 1. The role of open-source libraries
     2. Evolving web application technologies

The modernization requirements that must be considered are many in number. From securing API access, to securing transactions both local and international. Keeping everything up to date and making sure that all known exploits are patched out of the web application is a must.

## Areas of Security

**Input Validation:** Input validation is an important area to consider here due to the fact that Artemis Financial will have an online portal where users can access their information and see whatever confidential information Artemis may have pertaining to their accounts.

**APIs:** Since Artemis Financial will utilize a RESTful web API that will likely need to be accessible outside of the system, we will probably need to implement these APIs in a fashion that will not allow easy and unexpected intrusion into both user and administrative accounts.

**Cryptography:** Since we will be using the RESTful web API, as I stated previously, we would be justified in ensuring proper cryptography guidelines are utilized in order to safeguard Artemis Financials’ data, and their client’s data.

**Client/Server:** Artemis Financial must ensure that proper certification verification is occurring between Artemis’ servers and their clients as data is transferred via HTTP requests, this becomes even more relevant because of the use of Restful API.

**Code Error:** Artemis Financial and in turn us, as Global Rain, will have to extensively review for RESTful API interactions, as any potential API access may become a vulnerability.

**Encapsulation:** Since the RESTful API Artemis is using will likely be accessing data stored in the system, we need to encapsulate the code within the spring framework so that it can handle certain aspects of the security and initialization automatically.

## Manual Review

This section of the assignment highly confused me, everything seems unfinished and therefore full of vulnerabilities to begin with. In Crud.Java it is odd that the first two variables are constants in the form of “private final” while then being manipulated in the very same script. The first constructor is also out of place and sets both contents as the same thing, which doesn’t really make sense. The CrudController.Java does not ever run the 2nd constructor and therefore will always set them to the same thing. In DocData the information being requested in DocData.Java is never actually integrated, and it would never receive the correct information to begin with as it is reading for the business name. In Greeting.Java the content being retrieved in the constructor would be wrong. In the GreetingController.Java, incrementing is not limited and can continue on infinitum until the program is compromised. There is no verification, validation, or type checking that occurs within any of the Java files, including for string length, string case sensitivity, and forced variable conversion.

## Static Testing

* **The names or vulnerability codes of the known vulnerabilities**
* **A brief description and recommended solutions that are found in the dependency-check report**
* **Attribution (if any) that documents how this vulnerability has been identified or how it was documented in the past**

Graphical user interface, text, application

Description automatically generated

(**bcprov-jdk15on-1.46.jar**) CVE-2016-1000352 / CVE-2016-1000346 / CVE-2016-1000345 / CVE-2016-1000344 / CVE-2016-1000343 – A lot of problems here stemmed from DHIES / ECIES mode and the old version of the JCE provider under version 1.55. This could be solved by getting a newer version of the Bouncy Castle JCE Provider.

(**logback-core-1.2.3.jar**) CVE-2021-42550 – This exploit can be used if the intruder manages to gain the required privileges to exploit it. According to the Jira forms this is something of a moot exploit as an attacker who fulfills the requirements can simply perform an RCE.

(**snakeyaml-1.25.jar**) CVE-2022-38751 / CVE-2022-38750 / CVE-2022-38749 – If untrusted YAML files are parsed, the system may be vulnerable to Denial-of-Service attacks. The fix for this one seems to be just limiting the length of the value that can be parsed.

(**jackson-databind-2.10.2.jar**) CVE-2020-36518 – allows a StackOverflow exception and DOS via nested objects. This was fixed by optimizing “UntypedObjectDeserializer” as shown in the following link: <https://github.com/FasterXML/jackson-databind/issues/2816>

The rest of the vulnerabilities can be fixed by simply updating the relevant items. Ex. Spring boot update, spring framework update, Apache update.

## Mitigation Plan

Our mitigation plan begins with upgrading all outdated facets of our application, so that they are no longer an issue. This gets rid of the exploits we can solve right away and allows us to find the real problems that we need to fix. The next step in this process is to simply write better code, I don’t know if it was purposeful, but the code provided for this project was an unfinished mess that didn’t make much sense and was filled with vulnerabilities.