****

# 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** | **03-31-2024** | **Glasgow, Brandon** |  |

## 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

Brandon Glasgow

## Interpreting Client Needs

The client holds communication security of the utmost importance. Being that the client is in the finance industry the highest levels of security are imperative because there is a great deal of personal and financial information that is being shared and the financial well-being of both the client and its customers could be put at risk with any type of security issue. While the information was not shared in the introduction as to whether Artemis Financial makes international transactions, I will assume that like many other financial institutions they would.

In 2021, several executive orders to improve cybersecurity within the U.S. Communication security within the U.S. is also regulated by the DoD and Department Commerce. One such instance is laid out clearly in DoD Instruction 8523.01 “Communications Security”. There is, and quiet frankly, always be external threats posed to the client’s security, especially within the finance industry because many hackers are looking to gain something, and the financial industry is a literal “gold mine” to them. We must always consider the security improvements to technology that come about every single day. New levels of security like Two-Factor Authentication, Biometric Authentication, and other likewise added layers of security are constantly being created and improved. Allowing the security features of the system to improve to include the latest features is important to allow the client to protect their customers and themselves best.

## Areas of Security

*APIs:* As stated in the introduction, Artemis Financial already uses a RESTful API. The API will limit security compromises as well as connection to the API. Being that using an API allows for it to be enabled for two or more applications to communicate with we may also use open authorization (OAuth) that is supported by the API.

*Client/Server:* Communication between client and server would certainly be present. Client/Server also includes relationships between the various programs within an application as well as physical security which means protecting of hardware, networks, and data from actions that can be done physically across all components of the computer system.

*Cryptography:* Cryptography is imperative when accessing and transmit information throughout the internet. Without the data being properly encrypted it could compromise the integrity and/or confidentiality of the data of either the client (Artemis Financial) or their customers.

*Secure Coding:* Secure coding helps to protect the system from accidentally introduced vulnerabilities. Errors in code including bugs and logic flows within the coding process can create detrimental vulnerabilities within the system.

## Manual Review

ISSUE 1: Within the CRUDController.java file there is a vulnerability where the application may be exposing inside objects and then being retrieved and passed through code injection. For instance while “business\_name” is being passed through the CRUD method the DocData object is exposed creating the vulnerability.

ISSUE 2: Within the DocData.java file, within the data access method the method is to access data involving the description of the location of said data which is the username and password that has numerous vulnerabilities. This could allow for unauthorized users to easily gain system access through a brute force attack.

## Static Testing

The Following are the results of the dependency check along with their descriptions and CVE Codes, recommended solutions I will outline within the mitigation plan section:

Bouncy Castle, CVE-2015-6644.

The Bouncy Castle allows for malicious application to access the private data stored on the database. The recommended solution for vulnerability is to frequently update the application among the operating system to prevent a compromise of the security of the application.

Apache Log4j SMTP appender, CVE-2020-9488.

The Apache Log4j SMTP appender allows for log messages to be revealed if they are sent through the appended. The recommended solution for vulnerability is to upgrade to the newer version being 2.13.2 and having a built-in feature for verification.

Core Tomcat, CVE-2020-13935.

The Core Tomcat allows for high consumption of memory that can crash the system. The recommended solution for vulnerability is to use an updated version to address the dependency and the vulnerabilities.

## 

## Mitigation Plan

The first part of my suggested mitigation plan includes addressing the issues described in the dependency check results above. To address the issue in Bouncy Castle we must update the application among the operating system often to avoid malicious applications from compromising the security of the application. To address the Core Tomcat issue, updating to match the correct validation found within Apache Tomcat will support WebSocket Framework when triggering infinite loops in DoS attacks. Finally, to address the Apache SMTP Log4j appender issue we should update the server to improve the support of security measures. These remedies for the dependency check were provided by the NVD (National Vulnerability Database)

Another thing I suggest within my mitigation plan is that we must ensure that strong combinations of alphanumeric characters are being used in both the username and password to help mitigate the risk of a brute force attack. We also must use secure coding practices and vigorous code review so that errors are found and corrected before the system is launched. I also suggest that validation and verification of all certificates take place for the application and webserver. This will help to avoid the exploitation of some of the issues outlined in the dependency check, this would include the mutual checking of the TLS Certificates which also helps mitigate the vulnerability found within Bouncy Castle.