****

# CS 305 Project One

**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** | **11/20/2020** | **Jeff Caldarelli** |  |

## Client



## Instructions

Deliver this completed vulnerability assessment report, identifying your findings of security vulnerabilities and articulating recommendations for next steps to remedy the issues you have found.

Respond to the five steps outlined below and include your findings. Replace the bracketed text on all pages with your own words. If you choose to include images or supporting materials, be sure to insert them throughout.

## Developer

Jeff Caldarelli

## 1. Interpreting Client Needs

Determine your client’s needs and potential threats and attacks associated with their application and software security requirements. Consider the following regarding how companies protect against external threats based on the scenario information:

* What is the value of secure communications to the company?
* Are there any international transactions that the company produces?
* Are there governmental restrictions about secure communications to consider?
* What external threats might be present now and in the immediate future?
* What are the “modernization” requirements that must be considered, such as the role of open source libraries and evolving web application technologies?

[Include your findings here.]

My client, Artemis Financial, is concerned with the software security of their web based application. Since this application deals with sensitive data, it is extremely important to maintain secure software. For this reason, a high value is placed on securing company communications. I do not predict there will be international transactions, however this is a scenario to be aware of moving forward. Also, if there are any governmental restrictions regarding secure communications, we need to be aware of these as well. External threats could be wide ranging, for example false identification to gain access to a customers sensitive data, or a hacker infiltrating the system in another way. Some modernization requirements should also be considered, such as providing a framework to be compatible with any future updates or additions we might want to implement.

## 2. Areas of Security

Referring to the Vulnerability Assessment Process Flow Diagram, identify which areas of security are applicable to Artemis Financials’ software application. Justify your reasoning for why each area is relevant to the software application.

[Include your findings here.]

Cryptography – utilize encryption to ensure security of data

API’s – Artemis Financial is utilizing a RESTful API, we need to make sure it does not allow unexpencted access. If applicable, input validation can be fortified here, though there was no input commands noted in the code.

Client/Server – Due to the implementation of the API access, it is important to maintain proper certificates for protecting data transfers during https requests.

## 3. Manual Review

Continue working through the Vulnerability Assessment Process Flow Diagram. Identify all vulnerabilities in the code base by manually inspecting the code.

[Include your findings here.]

The spring-boot-starter-parent we are dependent on is out of date. The most recent is version 2.4.0

## 4. Static Testing

Run a dependency check on Artemis Financials’ software application to identify all security vulnerabilities in the code. Record the output from dependency check report. Include the following:

1. The names or vulnerability codes of the known vulnerabilities
2. A brief description and recommended solutions provided by the dependency check report
3. Attribution (if any) that documents how this vulnerability has been identified or documented previously

[Include your findings here.]

The Bouncy Castle Crypto package, represented as the [bcprov-jdk15on-1.46.jar](file:///C:\Users\jacal\OneDrive\Desktop\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l1_991c96a4e31e6c19e2b9136c8955bd423f2dc4c7) dependency. This dependency has a CVE count of 15, with most issues solved through upgrades.

The Apache Log4j API, represented as [log4j-api-2.12.1.jar](file:///C:\Users\jacal\OneDrive\Desktop\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l10_a55e6d987f50a515c9260b0451b4fa217dc539cb) on the dependency list. This is summarized as incorrect validation of certificate with host in Apache Log4j appender, which could allow interception of connection by a man-in-the-middle attack.

YAML 1.1 parser and emitter for Java, shown as [snakeyaml-1.25.jar](file:///C:\Users\jacal\OneDrive\Desktop\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l13_8b6e01ef661d8378ae6dd7b511a7f2a33fae1421) dependency. According to the National Vulnerability Database(NVD), this involves the Alias feature in SnakeYAML version 1.18, which allows entity expansion during load operation. Upgrade to latest version.

Spring Core, displayed as [spring-core-5.2.3.RELEASE.jar](file:///C:\Users\jacal\OneDrive\Desktop\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l15_3734223040040e8c3fecd5faa3ae8a1ed6da146b) dependency. The NVD states that with certain older Spring Framework versions, the protection against RFD attacks could be bypassed. Upgrade to latest version.

[Core Tomcat Implementation, as tomcat-embed-core-9.0.30.jar](file:///C:\Users\jacal\OneDrive\Desktop\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l16_ad32909314fe2ba02cec036434c0addd19bcc580). This dependency is described through the NVD as a CVE count of 8, with vulnerabilities such as avenues for pseudo headers as opposed to intended requested headers.

Core Tomcat implementation, as the [tomcat-embed-websocket-9.0.30.jar](file:///C:\Users\jacal\OneDrive\Desktop\CS%20305%20Module%20Two%20Static%20Testing%20Code%20Base\Module2.2\Module2.1\target\dependency-check-report.html#l18_33157f6bc5bfd03380ebb5ac476db0600a04168d) dependency. This dependency has a CVE count of 9, with vulnerabilities such as avenues for pseudo headers as opposed to intended requested headers.

## 5. Mitigation Plan

After interpreting your results from the manual review and static testing, identify the steps to remedy the identified security vulnerabilities for Artemis Financials’ software application.

[Include your findings here.]

Update the spring-boot-starter-parent to the latest version, 2.4.0. Also, apply the known solutions for the 6 vulnerabilities presented by the dependency check. Many of these solutions could be to update the various versions of utilized software. This seems to be the case for the first vulnerability, which I would plan to mitigate or correct by upgrading the utilized Legion of the Bouncy Castle Legion of the Bouncy Castle Java Cryptography API to version 1.60 or higher.