****

# 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** | **3/27/2022** | **Phillip Shirah** |  |

## 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

Phillip Shirah

## 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?

Artemis Financial is a financial consulting company. The very nature of their business is dealing with customer financial information. When dealing with this information, security is of great importance. Therefore, secure communications are a must for this company. Without secure communications, their client’s information is extremely vulnerable to falling into the wrong hands. Any financial company that deals in investments, must have at least a few transactions performed on the international stage. This company would have to comply with the Gramm-Leach-Bliley Act (GLBA), the Electronic Communications Privacy Act (ECPA), Federal Trade Commission Act (FTC Act), California Privacy Rights Act (CPRA), Virginia Consumer Data Protection Act (VCDPA), and Colorado Privacy Act (ColoPA). Artemis Financial, as with all companies, will need to constantly fight against external threats such as: phishing, malware, spyware, ransomware, denial of service attacks. Software and web applications are always changing and updating. It is a requirement of every business to continuously update and maintain systems in line with these changes and updates, failure to do so will result in creating vulnerabilities in the system.

## 2. Areas of Security

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

Input Validation – As a financial services company, input validation is very important. If the financial information is not validated, there could be serious financial implications down the road.

APIs – Secure API interactions are important for any web-based company as this can provide a serious avenue for vulnerabilities to hacking.

Cryptography – User’s information and financial information needs to be kept safe and secure. This is another level of safety if the system is hacked, then at least the data will be encrypted.

Client/Server – If this connection is not secure, it will offer yet another vulnerability for a hacker to exploit to gain access to the application.

Code Error – This is an often-overlooked area of security, but without proper error handling this can cause a vulnerability to be exploited in the application.

Code Quality – Poorly executed coding practices can lead to holes that a hacker can exploit to crash the software or gain access to unprotected information.

Encapsulation – Without proper encapsulation critical data or functionality can be bundled within components of the system making it vulnerable to hackers.

## 3. Manual Review

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

* There is no input validation for the message values in the application
* The version for the spring-data-rest-webmvc is out of date

## 4. Static Testing

Run a dependency check on Artemis Financial’s 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

* bcprov-jdk15on-1.46.jar, update to the latest version
  + CVE-2013-1624
  + CVE-2015-6644
  + CVE-2015-7940
  + CVE-2016-1000338
  + CVE-2016-1000339
  + CVE-2016-1000341
  + CVE-2016-1000342
  + CVE-2016-1000343
  + CVE-2016-1000345
  + CVE-2016-1000344
  + CVE-2016-1000346
  + CVE-2016-1000352
  + CVE-2017-13098
  + CVE-2018-1000613
  + CVE-2018-5382
  + CVE-2020-15522
  + CVE-2020-26939
* hibernate-validator-6.0.18.Final.jar, update to the latest version
  + CVE-2020-1069
* Jackson-databind-2.10.2.jar, update to the latest version
  + CVE-2020-25649
* log4j-api-2.12.1.jar, update to the latest version
  + CVE-2020-9488
* logback-core-1.2.3.jar, update to the latest version
  + CVE-2021-42550
* snakeyaml-1.25.jar, update to the latest version
  + CVE-2017-18640
* spring-aop-5.2.3.RELEASE.jar, update to the latest version
  + CVE-2020-5421
  + CVE-2021-22060
  + CVE-2021-22096
  + CVE-2021-22118
* spring-core-5.2.3.RELEASE.jar, update to the latest version
  + CVE-2020-5421
  + CVE-2021-22060
  + CVE-2021-22096
  + CVE-2021-22118
* tomcat-embed-core-9.0.30.jar, update to the latest version
  + CVE-2019-17569
  + CVE-2020-11996
  + CVE-2020-13934
  + CVE-2020-13935
  + CVE-2020-13943
  + CVE-2020-17527
  + CVE-2020-1935
  + CVE-2020-1938
  + CVE-2020-9484
  + CVE-2021-24122
  + CVE-2021-25122
  + CVE-2021-25329
  + CVE-2021-30640
  + CVE-2021-33037
  + CVE-2021-41079
  + CVE-2021-42340
* tomcat-embed-websocket-9.0.30.jar, update to the latest version
  + CVE-2019-17569
  + CVE-2020-11996
  + CVE-2020-13934
  + CVE-2020-13935
  + CVE-2020-13943
  + CVE-2020-17527
  + CVE-2020-1935
  + CVE-2020-1938
  + CVE-2020-8022
  + CVE-2020-9484
  + CVE-2021-24122
  + CVE-2021-25122
  + CVE-2021-25329
  + CVE-2021-30640
  + CVE-2021-33037
  + CVE-2021-41079
  + CVE-2021-42340

## 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 Financial’s software application.

* We will need to add input validation into the GreetingController to limit the length of the data input so as not to have massive data sizes allowed.
* We will need to update our spring-data-rest-webmvc. Keeping software and dependencies updated helps prevent security vulnerabilities.
* Update all the dependencies from the dependency check.