****

# 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** | **5/25/2022** | **Tierney Shaughnessy** |  |

## 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

Tierney shaughnessy

## 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 company that creates financial plans for clients. Their services range from retirement to investments to savings. Secure communications is vital for a business such as this, as the information the company handles is deeply sensitive – tax ID’s, Social Security Numbers, the list goes on. There is no suggestion that Artemis Financial only services the US, so it’s likely the company deals with transactions both domestic and international. A major governmental restriction that the company will need to consider about secure communications in this field is regarding prevention of exposure to trade secrets. As the information is so sensitive, a major external threat would be targeting client information. To protect this, most if not all of the data will need to be greatly encrypted from those not in the company. As for modernized reqiuirements, keeping up to date with maintenance in terms of fixing bugs and handling security threats and vulnerabilities will be needed.

## 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.

* Cryptography
  + Thanks to the sensitive nature of the information Artemis Financial handles, cryptography would be absolutely vital. This is because it ensures that user information will be secure and outside attacks would have as little chance as possible to access it.
* Code Quality
  + The quality of the code, especially in terms of security, is paramount. In this case, implementing methods that control access based on the user would be an example of maintaining the quality of the code itself.
* API’s
  + An API would be needed as it is running both internally and externally. This allows for monitoring which data is appropriately accessed.
* Input Validation
  + This information is private and therefore, it is necessary to validate which information belongs to who so that the users are protected.
* Code Error
  + Handling errors is imperative to creating good code and a good product. Testing also will help with things such as the API to see what needs fixing.

## 3. Manual Review

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

In examining the code, I looked at the Greeting Controller as well as the POM.XML file. With the XML I wanted to see if there was an Apache Validator implemented. I could not find one, but I would certainly have an associate double check to ensure more than one set of eyes was on that. As for the Greeting Controller file, I couldn’t find any Input Validation. That is something we’ll need to keep track of going forward. I also tried to look for any code involving cryptography, but I was not able to locate it.

## 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

* [hibernate-validator-6.0.18.Final.jar](file:///C:\Users\slain\Desktop\Module2.2\target\dependency-check-report.html#l2_7fd00bcd87e14b6ba66279282ef15efa30dd2492)
  + Hibernate's Bean Validation (JSR-380) reference implementation, CWE-20 Improper Input Validation
* [jackson-databind-2.10.2.jar](file:///C:\Users\slain\Desktop\Module2.2\target\dependency-check-report.html#l4_0528de95f198afafbcfb0c09d2e43b6e0ea663ec)
  + General data-binding functionality for Jackson: works on core streaming API, CWE-611 Improper Restriction of XML External Entity Reference ('XXE')
* [log4j-api-2.12.1.jar](file:///C:\Users\slain\Desktop\Module2.2\target\dependency-check-report.html#l9_a55e6d987f50a515c9260b0451b4fa217dc539cb)
  + The Apache Log4j API, CWE-295 Improper Certificate Validation
* [logback-core-1.2.3.jar](file:///C:\Users\slain\Desktop\Module2.2\target\dependency-check-report.html#l11_864344400c3d4d92dfeb0a305dc87d953677c03c)
  + logback-core module, CWE-502 Deserialization of Untrusted Data
* [snakeyaml-1.25.jar](file:///C:\Users\slain\Desktop\Module2.2\target\dependency-check-report.html#l14_8b6e01ef661d8378ae6dd7b511a7f2a33fae1421)
  + YAML 1.1 parser and emitter for Java, CWE-776 Improper Restriction of Recursive Entity References in DTDs ('XML Entity Expansion')
* [spring-aop-5.2.3.RELEASE.jar](file:///C:\Users\slain\Desktop\Module2.2\target\dependency-check-report.html#l15_9cdd9a1dd636331767fffcc7fe49a7bb00e7b34b)
  + Spring AOP, CWE-502 Deserialization of Untrusted Data
* [spring-boot-2.2.4.RELEASE.jar](file:///C:\Users\slain\Desktop\Module2.2\target\dependency-check-report.html#l16_225a4fd31156c254e3bb92adb42ee8c6de812714)
  + Spring Boot, CWE-668 Exposure of Resource to Wrong Sphere
* [spring-core-5.2.3.RELEASE.jar](file:///C:\Users\slain\Desktop\Module2.2\target\dependency-check-report.html#l17_3734223040040e8c3fecd5faa3ae8a1ed6da146b)
  + Spring Core, CWE-502 Deserialization of Untrusted Data
* [tomcat-embed-core-9.0.30.jar](file:///C:\Users\slain\Desktop\Module2.2\target\dependency-check-report.html#l18_ad32909314fe2ba02cec036434c0addd19bcc580)
  + Core Tomcat implementation, CWE-269 Improper Privilege Management
* [tomcat-embed-websocket-9.0.30.jar](file:///C:\Users\slain\Desktop\Module2.2\target\dependency-check-report.html#l20_33157f6bc5bfd03380ebb5ac476db0600a04168d)
  + Core Tomcat implementation, CWE-269 Improper Privilege Management

## 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.

Depending on what the Client needs, the mitigation could be varied – but a swift plan could very much be implementing the up-to-date versions of the above vulnerabilities.