****

# 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** | **1/23/2022** | **Matthew Trembley** |  |

## 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

Matthew Trembley

## 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 finds great value in secure communications due to the fact they deal with both internal and external information for each of their clients. Seeing as Artemis Financial is a financial firm, a few special cases of abroad clients may arise, in which international transactions will occur. Currently there are no governmental restrictions for the secure communications to be considered, but that by no means there is a place for leniency. Artemis Financial will be handling a lot of private information for their clients, thus all the information should really be encrypted while stored and in any transmission of data. Artemis Financial will need to ensure that the library they use stays up to date with their application, that way any fixes for security breaches or bugs will be fixed.

## 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.

There are a few areas of security that Artemis Financial should be concerned about.

* Input validation – Validating user input is incredibly important. User input validation by means of input string validation will avoid any security breaches like SQL injection.
* Cryptography – Cryptography will be one of the largest areas of concern. As Artemis Financial will be dealing with a lot of private information for their clients, cryptography will be the largest protector of this information.
* Error handling – Any error should be handled correctly, and this will work with input validation very closely. This will help prevent unauthorized access by means of validating input, and if input is erroneous, dealt with correctly.
* Code quality – Code quality is very important that way information is not exposed on accident when working with error handling and input validation.

## 3. Manual Review

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

Manually inspecting the code, I found a few vulnerabilities worth mentioning. First of all, I noticed that the GreetingController did not use any kind of validation. Next, I noticed the program access data with a URL with the POST method, which is quite dangerous because data can be leaked into the user’s browser’s history. It also seems there is no real API within the program, so the end user will could be pretty confused using it because they would have to directly access the code – which is dangerous for any malicious user

## 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 | * An information disclosure vulnerability in Bouncy Castle could enable a local malicious application to gain access to users private information * Bouncy Castle library before 1.51 does not validate a point which makes it easier for remote attackers to obtain private keys * Bouncy Castle before 1.61, attackers can obtain sensitive information about a private exponent because of Observable Differences in Behavior to Error Inputs | Update to latest information of Bouncy Castle library Java |
| Hibernate-validator-6.0.18.Final.jar | * A flaw was found in Hibernate Validator version 6.1.2 Final. A bug in the message interpolation processor enables invalid EL expressions to be evaluated as if they were valid. This flaw allows attackers to bypass input sanitation (escaping, stripping) controls that developers may have put in place when handling user-controlled data in error messages. | Update to latest version of Hibernate-validator |
| Jackson-databind-2.10.2.jar | * A flaw was found in FasterXML Jackson Databind, where it did not have entity expansion secured properly. This flaw allows vulnerability to XML external entity (XXE) attacks. The highest threat from this vulnerability is data integrity. | Update to latest version of Jackson-databind |
| log4j-api-2.12.1.jar | * Improper validation of certificate with host mismatch in Apache Log4j SMTP appender. This could allow an SMTPS connection to be intercepted by a man-in-the-middle attack which could leak any log messages sent through that appender. | Update to latest version of log4j-api |
| logback-core-1.2.3.jar | In logback version 1.2.7 and prior versions, an attacker with the required privileges to edit configurations files could craft a malicious configuration allowing to execute arbitrary code loaded from LDAP servers. | Update to latest version of logback |
| snakeyaml-1.25.jar | The Alias feature in SnakeYAML 1.18 allows entity expansion during a load operation, a related issue to CVE-2003-1564. | Update to latest version of snakeyaml |
| spring-aop-5.2.3.RELEASE.jar | In Spring Framework versions 5.2.0 - 5.2.8, 5.1.0 - 5.1.17, 5.0.0 - 5.0.18, 4.3.0 - 4.3.28, and older unsupported versions, the protections against RFD attacks from CVE-2015-5211 may be bypassed depending on the browser used through the use of a jsessionid path parameter. | Update to latest version of Spring |
| spring-core-5.2.3.RELEASE.jar | In Spring Framework versions 5.3.0 - 5.3.13, 5.2.0 - 5.2.18, and older unsupported versions, it is possible for a user to provide malicious input to cause the insertion of additional log entries. This is a follow-up to CVE-2021-22096 that protects against additional types of input and in more places of the Spring Framework codebase. | Update to latest version of Spring |
| tomcat-embed-core-9.0.30.jar  tomcat-embed-websocket-9.0.30.jar | * The refactoring present in Apache Tomcat 9.0.28 to 9.0.30, 8.5.48 to 8.5.50 and 7.0.98 to 7.0.99 introduced a regression. The result of the regression was that invalid Transfer-Encoding headers were incorrectly processed leading to a possibility of HTTP Request Smuggling if Tomcat was located behind a reverse proxy that incorrectly handled the invalid Transfer-Encoding header in a particular manner. Such a reverse proxy is considered unlikely. * A specially crafted sequence of HTTP/2 requests sent to Apache Tomcat 10.0.0-M1 to 10.0.0-M5, 9.0.0.M1 to 9.0.35 and 8.5.0 to 8.5.55 could trigger high CPU usage for several seconds. If a sufficient number of such requests were made on concurrent HTTP/2 connections, the server could become unresponsive. | Update to latest version of tomcat |

## 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.

After reviewing the static test on the code base and a manual review of the code, the following recommendations are as follows: update all databases/libraries in use with existing program. These dependencies can easily be corrected by making sure the program uses up-to-date security protocols. Next, introduce an API for the user, that way there is no direct interaction with the code in itself. Finally, include more error handling for clearer input validation.