****

# 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** | **5/22/2021** | **Danny Faught** | **neato** |

## 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

Danny Faught

## 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?

Secure communications over the system that Artemis Financial uses will be of utmost importance. They handle sensitive financial data. They also have a RESTful API that will need to be secured. Data transferred will need to be secured according to any government and international (if applicable) standards. As AF uses an API, this means that some or all of the endpoints may be exposed to the internet and may be vulnerable to an attacker. Threats in this realm could range from DoS attacks to SQL injections depending on the function of these APIs.

## 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.

APIs will need to be investigated  
Cryptography could be involved if data is very sensitive  
Client/Server interactions need to be secured for the API

## 3. Manual Review

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

Comments left in DocData with sensitive info

Sql connection left open

DocData constructor empty – would never set ID

No toString override implemented which is called in CRUDController

Assignment to final variables in Greeting  
myDateTime not finished  
no error handling

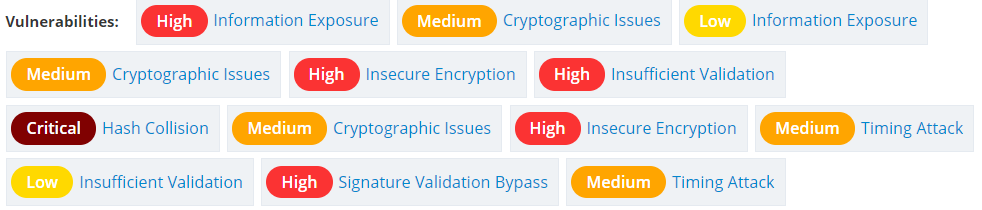
Unused classes.

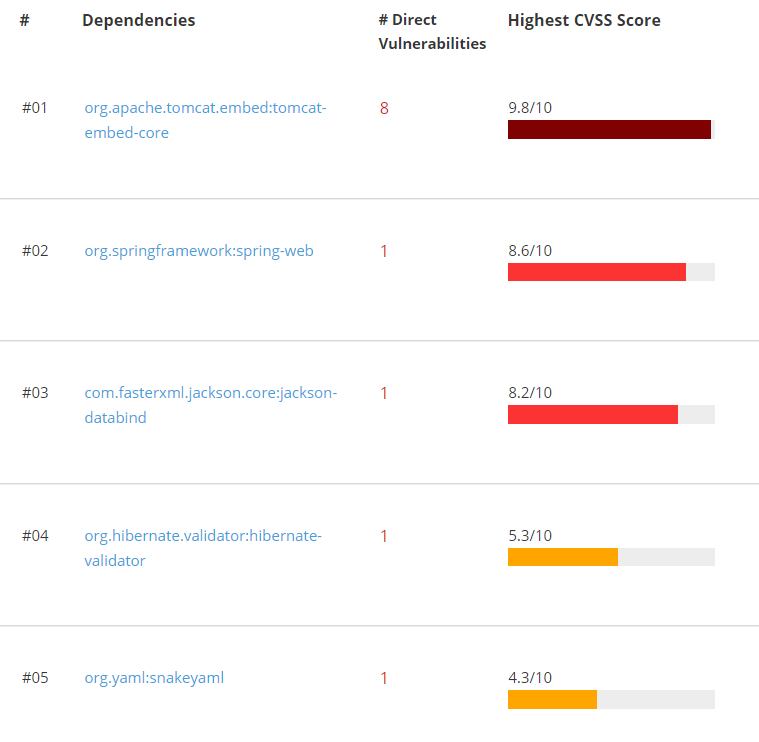
## 4. Static Testing

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

13 direct vulnerabilities detected for bouncycastle of various severities.

  
Solution: Upgrade to version 1.68 (no vulnerabilities reported so far)

5 transitive vulnerabilities found in spring-boot-starter-web of various severities  


Solution: All dependent packages can be updated to latest versions (upgrade spring-boot-starter-web)

## 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.

1. Update all packages with outstanding vulnerabilities. This will ensure that the application is not vulnerable due to the vulnerabilities of the packages it is using.
2. Fix code errors. Several things are wrong with this application as far as code goes. Very severe is the sql username and password left in the comments of this application and the sql connection being left open. There are a few syntactical errors in this code that will need to be addressed. There are unused classes in this code. There is no error handling.