****

# 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** | **1/29/2021** | **Kyle Ogle** |  |

## 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

Kyle Ogle

## 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 provides financial plans for their customers. Since this financial information is highly sensitive, all communications between the company and their clients must be secure. Artemis Financial will also need to follow government regulations regarding financial transactions and communications, which will affect data retention policies and security requirements. In general, RESTful APIs such as Artemis Financial’s are vulnerable to data interception if requests and responses are not structured securely. In particular, the service must be sure to use HTTPS for all communications and send confidential information in request and response headers or in the request or response body. Additionally, a secure authentication scheme such as OAuth should be used.

## 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 – because the RESTful API will accept user input, this input will have to be validated
* APIs – the web service includes a RESTful API which will need to communicate securely
* Code error – errors due to improper user input must be handled securely
* Secure coding

## 3. Manual Review

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

* Business names are sent as request parameters in the CRUDController class
* Database connection parameters are hard-coded in DocData
* Request parameters are not validated
* Service does not use HTTPS
* No authentication scheme present

## 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

* Bouncy Castle version 1.46 has several vulnerabilities. Recommendation: update to at least 1.60
  + 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-2016-1000344
  + CVE-2016-1000345
  + CVE-2016-1000346
  + CVE-2016-1000352
  + CVE-2017-13098
  + CVE-2018-1000613
  + CVE-2018-5382
* FasterXML Jackson Databind 2.10.2 has one vulnerability. Recommendation: update to at least 2.10.5.1
  + CVE-2020-25649
* Apache Log4j API 2.12.1 has one vulnerability. Recommendation: update to at least 2.13.2
  + CVE-2020-9488
* SnakeYAML 1.25 has one vulnerability. Recommendation: update to at least 1.26
  + CVE-2017-18640
* Spring Core 5.2.3 has one vulnerability. Recommendation: update to at least 5.2.8
  + CVE-2020-5421
* Apache Tomcat 9.0.30 has several vulnerabilities. Recommendation: update to at least 9.0.40
  + 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

## 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.

* Switch to HTTPS protocol for all communications
* Move request parameters to headers or body rather than URI
* Remove hard-coded database connection credentials
* Implement secure authentication scheme
* Update dependencies as listed above