# CS 305 Project One

**Victor Landi**

**Southern New Hampshire University**

[**CS-305-T4502 Software Security 22EW4**](https://learn.snhu.edu/d2l/home/1008989)

**Dr. Vivian Lyon**

**March 2022**

****

# 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** | **March 2022** | **Victor Landi** |  |

## 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

Victor Landi

## 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 their customers with financial plans. This information will need to be securely communicated between the company and their client base. The firm also needs to follow government regulations concerning financial transactions and communications worldwide. RESTful APIs , used by the firm’s data retention policies and security requirements are extremely high on the priority list for Artemis Financial. Data is vulnerable to data interception if requests and responses are not structured securely. It is recommended that HTTPS is used for all communications. Confidential information needs to be sent in request and response headers or in the request or response body. A secure authorization scheme such as OAuth needs to be used.

## 2. Areas of Security

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

Input Validation – Input will have to be validated, since the RESTful API accepts user input

“One potentially dangerous scenario that we see frequently in web application code is missing input validation of untrusted data. When the user submits a web form in the browser, input parameters must be validated on the server to ensure all data is consistent with business requirements ” (Manico & Detlefsen, 2014).

APIs – Secure communication will be needed with the RESTful API used by the web service

Code Error – Errors can occur from improper user input; this will need to be managed 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.

Authentication scheme is not present

HTTPS is not used by the service

Request parameters are not validated

“By putting the parameters in the body of the request instead of the URL, the submitted data is not leaked in the browser history list, referer headers, or server access logs” (Manico & Detlefsen, 2014).

CRUDController class accepts Business names sent as request parameters

Lines 13,14

Database connection parameters are Hard coded in DocData

Lines 26 - 30

Research by Manico and Detlesfen (2014)) supports this,” The critical design flaw here, as well as one of the most prominent features in many software frameworks, is the merging of access control policy and application code. This “hard coding” of access control policy leads to a variety of challenges.

When you “hard code” policy in your application code like this, you need to push a new version of your software any time your access control policy changes. It also makes it very difficult to ensure that access control is consistent across every place that a similar check has to be made.”

## 4. Static Testing

Run a dependency check on Artemis Financials’ 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 many 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 many 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 Financials’ software application.

All communication will need to use HTTPS protocol

Updated dependencies as listed above

Removal of hard-coded database connection credentials

Relocate request parameters to headers or body rather than URL

Implement a secure authentication scheme

**References:**

Jim Manico and August Detlefsen. 2014. Iron-Clad Java: Building Secure Web Applications (Oracle Press) (1st. ed.). McGraw-Hill Education Group.