

# 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** | **11/13/2020** | **Caio De Morais** | **vulnerability assessment** |

## 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

Caio De Morais

## 1. Interpreting Client Needs

* Artemis Financial is a financial consulting company that develops individualized financial plans for retirement, saving, investments and others for their patrons. They desire to modernize its operating and to improve data security, they want to implement the newest and effective software security. Artemis Financial currently has RESTful web application programming interface and we will be helping them to protect their organization from external threats. We will be examining Artemis Financials’ web-based software application to identify any types of vulnerabilities and threats in their current software.
* There are vulnerabilities while leveraging APIs today, the most common REST API Security Threats are: Injection Attacks, DoS attacks, broken authentication, Man-In-The-Middle-Attack and broken access control.

## 2. Areas of Security

* Input Validation:

Input validation should happen as early as possible in the data flow. Data from untrusted sources should be subjected to input validation, although input validation helps preventing SQL injection attacks, it should not be used as the primary method of prevention. Input validation is crucial to protect customer’s data.

* APIs:

The use of REST API is recommended since it support transport layer security which send encrypted messages between server and client and keeps an internet connection private. API security is crucial since hacked APIs are behind major data breaches. Artemis Financial currently has RESTful web application programming interface.

* Cryptography:

Identify all sensitive data and encrypt it, ensure that data cannot be easily overwriting, protect keys and in some cases hide the keys even from the administrator. We can implement some

* Client/Server:

Connection between client and server must be secure and encrypted.

* Code Quality:

Test code to ensure it is secure and accurate.

* Encapsulation

Declare the variable of some classes as private, no one outside the class can access private data fields via public methods.

## 3. Manual Review

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

[Include your findings here.]

## 4. Static Testing

* 27 vulnerabilities were found.
* Dependency log4j-api-2.12.1.jar had one CVE (Low Severity) - Published Vulnerabilities: CVE-2020-9488.
* Dependency spring-core-5.2.3.RELEASE.jar had one CVE (Medium Severity) - Published Vulnerabilities: CVE-2020-5421
* Dependency snakeyaml-1.25.jar had one CVE (High Severity) - Published Vulnerabilities: CVE-2017-18640.
* Dependency tomcat-embed-core-9.0.30.jar had nine CVE (Critical Severity) - Published Vulnerabilities: CVE-2019-17569, CVE-2020-11996, CVE-2020-13934, CVE-2020-13935, CVE-2020-13943, CVE-2020-1935, CVE-2020-1938, CVE-2020-8022, CVE-2020-9484.
* Dependency bcprov-jdk15on-1.46.jar had fifteen CVE (Unknown Severity) - Published Vulnerabilities: C CVE-2018-1000613, CVE-2016-1000352, CVE-2016-1000346, CVE-2016-1000345, CVE-2016-1000344, CVE-2016-1000343, CVE-2016-1000342, CVE-2016-1000341, CVE-2016-1000339, CVE-2016-1000338, CVE-2018-5382, CVE-2017-13098, CVE-2013-1624.
* 15 vulnerabilities from dependency “bcprov-jdk15on-1.46.jar” had unknown severity (Possible false positives). If confirmed that it is a false positive, suppress using HTML report.

## 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.

• Solution for vulnerabilities found on Dependency log4j-api-2.12.1.jar: Upgrade dependency to version 2.13.2, the upgrade will reduce the chances of an attacker with man-in-the-middle access to intercept log messages sent through SMTPS.

• Solution for vulnerabilities found on Dependency snakeyaml-1.25.jar: Upgrade dependency to version 1.27 will solve this vulnerability.

• Solution for vulnerabilities found on Dependency tomcat-embed-core-9.0.30.jar: Install openSUSE Security Update. To install this openSUSE Security Update use the SUSE recommended

installation methods like YaST online\_update or "zypper patch".

* Solution for vulnerabilities found on Dependency bcprov-jdk15on-1.46.jar: Upgrade bouncycastle to version 1.60 or add full filtering for BDS data.
* Solution for vulnerabilities found on Dependency spring-core-5.2.3.RELEASE.jar: Upgrade to version 5.2.9, 5.1.18, 5.0.19, or 4.3.29