****

# 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** | **11/12/2020** | **Nicholas Wood** |  |

## 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

Nicholas Wood

## 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 is a company that has decided to modernize their operations, they wanted to apply the most current and effective software security for their operations. Since they are a financial company, they will be dealing with sensitive information of each of their clients. Their clients will also need to have good communication through the web with the financial services to see their information or to change their investments. Communications is vital to the company as many of their clients may interact through the web. Artemis Financial may be dealing with organizations or clients around the globe so it is vital be up to date with all the security software. I understand with government agencies their may be certain software requirements to meet and if you are dealing with government money then they would have many restrictions that need to follow. External threats can include other countries or institutions looking to take control of assets from individuals or organizations. Before they try to modernize their web applications, they need to make sure that they are completely secure, with many of the newer applications some of them are not completely safe from attacks and having financial information to protect should be of the utmost importance.

## 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.

The areas of security that are applicable to Artemis Financial software application are input validation, secure API interactions, and cryptography. I believe the other areas may also be relevant, but I feel these elements are much more relevant to Artemis Financial. The input validation is highly desirable as it makes sure that the users accessing the account is authorized. With the organization involved with finances that deal with user’s secure information, the log in authorization must be protected. The secure API’s also must be secured properly, with the increase in injections hacking we must make sure the program is secure. The security of the application is needed as it makes sure the users have confidence user the product. We must also make sure that the site is has encryption so that when the user is using a public computer than the secure access will not leak. It must also be encrypted so that no over the shoulder hacker can take the link and access the users account.

## 3. Manual Review

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

Reviewing the code has given me some insight into what some things they may need to change or add to the system. I like how they use CRUD for creating, reading, updating and deleting information of the users and storing this information. I did not see any protection to this; however, it seems that anyone can access a user’s information and there doesn’t seem to be any protection with this. It also seems that SQL is implemented in sections, but it is not complete which could cause many errors in the program. I did not see any information regarding the security of the application when it comes to the html, there doesn’t seem to be any encryption to the code as well which would lead to some over the shoulder hacking and maybe some leakage as well. I also did not see any area that authorized users as well which would lead to hacking of financial information of the 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

CVE-2020-9488: 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. This has been documented and considered resolved as well.

CVE-2018-18640: YAML 1.1 parser and emitter for Java. This issue was documented 05-2017 and a solution was developed much later.

CVE-2020-5421: In Spring Framework versions 5.20-5.2.8, 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 on the browser used through the use of a jsessionid path parameter. This was discovered by and resolved by Keitaro Yamazaki from Ierae Security.

The other vulnerabilities I would consider is 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, and CVE-2018-5382. I am considering these as vulnerabilities because it talks about the Bouncy Castle Crypto package and its issues and typically I would consider this a false positive but after reading through it, I believe there are many underlying issues with the Bouncy Castle Crypto package that could be vulnerable to the system.

## 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.

The solution to CVE-2020-9488 is to upgrade to 2.13.2 which supports this feature. Previous versions can set the system property mail.smto.ssl.checkserveridentity to true to globally enable hostname verification for SMTPS connections.

The solution to CVE-2018-18640 is to update to version 1.26 which resolves this issue.

To fix this solution Spring Framework need to be updated to the following versions, 5.2.9, 5.1.18, 5.0.19, and 4.3.29.

For the Bouncy Castle Crypto I would consider looking at another Crypto program to implement due to the large amount of issues with the Bouncy Castle Crypto. There are many issues that are not resolved with the Bouncy Castle Crypto and some of the issues that are identified by the dependency report are possible issues and not documented as actual flaws with the crypto package.