****

# CS 305 Project One

**Artemis Financial Vulnerability Assessment Report**

**Table of Contents**

[Document Revision History 3](#_gjdgxs)

[Client 3](#_30j0zll)

[Instructions 3](#_3znysh7)

[Developer 4](#_2et92p0)

[1. Interpreting Client Needs 4](#_tyjcwt)

[2. Areas of Security 4](#_3dy6vkm)

[3. Manual Review 4](#_1t3h5sf)

[4. Static Testing 4](#_4d34og8)

[5. Mitigation Plan 4](#_2s8eyo1)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **1/23/22** | **Ayden Hooke** |  |

## 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

Ayden Hooke

## 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 deals with customers, their money, and the future of their money. This important customer information should be classified and only able to be accessed by those who are authorized to access it. Thus, it is absolutely critical that this company has secure communications to avoid any sort of data leak. Since Artemis Financial is an engineering company that services entrepreneurs, businesses, and government agencies around the world, there are certainly going to be international transactions made by the company. Like any other financial company, there are plenty of governmental restrictions and regulations that must be followed in order to protect the company, the customers, and anyone else who is involved with the company. The primary external threats that Artemis Financial would face are attacks on their system seeking information about their clients and their financial information. This would be both a threat they would have to worry about now and forever. Some of the modernization requirements that must be considered would be the role of open source libraries. Libraries such as these see updates more often and can be more secure than other options. Therefore, it should be important to Artemis Financial that their software is continually updated. In addition to this, there are other web technologies that would help to keep information safe, such as two-factor authentication for logins and secure access to their website using https instead of http.

## 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 most important areas of security for Artemis Financial’s software application are apis, cryptography, client/server, and code quality. APIs are important because without a secure use of APIs, client information can be leaked when applications are communicating with one another. Cryptography is incredibly important because of the importance that should be placed on keeping the client’s data secure. By encrypting and decrypting clients’ data, Artemis Financial would cause hackers to have a much more difficult time decoding data in the case that they were able to access it unauthorized. Client/server interaction is important because the server must safely store client’s data and the client must also store data safely and securely. Additionally, the passing of data between the two could be a security risk if it is not properly protected and coded. This is also the last important area of security -- code quality. Code quality is not only important because proper code implements standard security features (like setting things to public vs private), but also because without well written code, there is a potential for data loss if the data is not handled properly.

## 3. Manual Review

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

A massive vulnerability in the base code is that the default username and password are both “root.” Given that this is a common default password and username, it would likely be one of the first ones checked, hence the severity of this vulnerability. In CRUDController, the business name is requested as an input, and it is a public function in a public class. This could potentially be a vulnerability, as well as the showInfo() function in customer.java, since that is a public function that could give out information to unauthorized people if it is able to be manipulated maliciously.

## 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

bcprov-jdk15on-1.46.jar - update to the latest version

* CVE-2013-1624
* CVE-2015-6644 (OSSINDEX)
* 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 (OSSINDEX)
* CVE-2018-5382
* CVE-2020-15522
* CVE-2020-26939 (OSSINDEX)

hibernate-validator-6.0.18.Final.jar - update to the latest version

* CVE-2020-10693

jackson-databind-2.10.2.jar - update to the latest version

* CVE-2020-25649

log4j-api-2.12.1.jar - update to the latest version

* CVE-2020-9488

logback-core-1.2.3.jar - update to the latest version

* CVE-2021-42550

snakeyaml-1.25.jar - update to the latest version

* CVE-2017-18640

spring-aop-5.2.3.RELEASE.jar - update to the latest version

* CVE-2020-5421
* CVE-2021-22060
* CVE-2021-22096
* CVE-2021-22118

spring-core-5.2.3.RELEASE.jar - update to the latest version

* CVE-2020-5421
* CVE-2021-22060
* CVE-2021-22096
* CVE-2021-22118

tomcat-embed-core-9.0.30.jar - update to the latest version

* 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-2020-9484
* CVE-2021-24122
* CVE-2021-25122
* CVE-2021-25329
* CVE-2021-30640
* CVE-2021-33037
* CVE-2021-41079
* CVE-2021-42340

tomcat-embed-websocket-9.0.30.jar - update to the latest version

* 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
* CVE-2021-25122
* CVE-2021-25329
* CVE-2021-30640
* CVE-2021-33037
* CVE-2021-41079
* CVE-2021-42340

For all dependencies, the common solution is to update to the latest version to remove the security risks.

## 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.

The first step to mitigate risks for Artemis Financial is to update all their software to the latest versions that are available. By using the most updated versions of everything, Artemis Financial can avoid the security exploits that have been shown to exist in prior, outdated versions. Changing the default username and password from “root” would also remedy security risks, as well as ensuring that what functions need to be public are public, but everything else is private. Artemis Financial can also implement two-factor authentication and the use of https with their website if they aren’t using them already. Ensuring that there are authentication checks everywhere feasible will greatly increase security.