****

# 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 5](#_Toc32574613)

[4. Static Testing 5](#_Toc32574614)

[5. Mitigation Plan 6](#_Toc32574615)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **5/21/2022** | **Steven Estrada** |  |

## 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

Steven Estrada

## 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?

Looking over the client’s summary we can find useful information regarding what type of services would be required. Artemis Financial is a financial consulting company that develops individualized financial planning for a variety of individuals with different financial goals. Because of the type of work that is being done at Artemis Financial, cyber security should be priority. The company deals with the financial details of individuals which can be very beneficial and therefore may call the attention of attackers trying to steal money. From the company’s description, it is not fully understood if there are any international transactions that the company produces. One may imply if the software is on the internet, then the client may be international. One governmental restriction that may apply to secure communications is due to governmental export restrictions. External threats that are presents now and a possibility include: inadequate encryption, lack of security of the servers, lack of customer password privacy, and faulty authentication. Some “modernization” requirements that will need considerations include larger passwords and two-step authentication.

## 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.

Using the Vulnerability Assessment Process Flow Diagram, we have a few of the areas of security that are applicable to Artemis Financial’s software application. The first one will be cryptography. Cryptography is important when it comes to secure communication of finances for both domestic and international government and is a regulated export item. The other area of security is input validation. Clients of Artemis Financial will be accessing their financial profiles through a log in page. This log in page can be the target of SQL injections and will need to be secured enough to keep attackers away, but allow the customers to access their information.

## 3. Manual Review

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

When searching through the code base manually, there are areas of improvement that could be made to improve the security of the web application. One simple improvement that could be made is to secure the way that the sensitive data is being transferred. Solution for this will be referred to in the mitigation plan. Lack of access control can also be an issue when creating the web application. Lastly, we would need to include query parameterization to build a dynamic SQL statement.

## 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. [bcprov-jdk15on-1.46.jar](file:///C:\Users\typel\eclipse-workspace\Project%201\target\dependency-check-report.html#l1_991c96a4e31e6c19e2b9136c8955bd423f2dc4c7)
3. [hibernate-validator-6.0.18.Final.jar](file:///C:\Users\typel\eclipse-workspace\Project%201\target\dependency-check-report.html#l3_7fd00bcd87e14b6ba66279282ef15efa30dd2492)
4. [jackson-databind-2.10.2.jar](file:///C:\Users\typel\eclipse-workspace\Project%201\target\dependency-check-report.html#l5_0528de95f198afafbcfb0c09d2e43b6e0ea663ec)
5. [log4j-api-2.12.1.jar](file:///C:\Users\typel\eclipse-workspace\Project%201\target\dependency-check-report.html#l10_a55e6d987f50a515c9260b0451b4fa217dc539cb)
6. [logback-core-1.2.3.jar](file:///C:\Users\typel\eclipse-workspace\Project%201\target\dependency-check-report.html#l12_864344400c3d4d92dfeb0a305dc87d953677c03c)
7. [snakeyaml-1.25.jar](file:///C:\Users\typel\eclipse-workspace\Project%201\target\dependency-check-report.html#l14_8b6e01ef661d8378ae6dd7b511a7f2a33fae1421)
8. [spring-aop-5.2.3.RELEASE.jar](file:///C:\Users\typel\eclipse-workspace\Project%201\target\dependency-check-report.html#l15_9cdd9a1dd636331767fffcc7fe49a7bb00e7b34b)
9. [spring-boot-2.2.4.RELEASE.jar](file:///C:\Users\typel\eclipse-workspace\Project%201\target\dependency-check-report.html#l16_225a4fd31156c254e3bb92adb42ee8c6de812714)
10. [spring-core-5.2.3.RELEASE.jar](file:///C:\Users\typel\eclipse-workspace\Project%201\target\dependency-check-report.html#l17_3734223040040e8c3fecd5faa3ae8a1ed6da146b)
11. [tomcat-embed-core-9.0.30.jar](file:///C:\Users\typel\eclipse-workspace\Project%201\target\dependency-check-report.html#l18_ad32909314fe2ba02cec036434c0addd19bcc580)
12. [tomcat-embed-websocket-9.0.30.jar](file:///C:\Users\typel\eclipse-workspace\Project%201\target\dependency-check-report.html#l20_33157f6bc5bfd03380ebb5ac476db0600a04168d)
13. A brief description and recommended solutions provided by the dependency check report

Looking at the Dependency Check Report for the code base we find: externally-controlled input, bypassing input sanitation controls, vulnerability to XML external entity attacks, man-in-the-middle attacks, execution of arbitrary code loaded from LDAP servers, entity expansion during a load operation, and potential remote code execution. The most obvious solution to a majority of these vulnerabilities is to download and update the version of the dependencies. Updates take care of most of the issues found in the dependency report.

1. Attribution (if any) that documents how this vulnerability has been identified or documented previously

The following attributions that document how this vulnerability has been identified include: github.com, redhat.com, oracle.com, nvd.com, and ubuntu.com

## 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. Securing the sensitive data being transfer is top priority. For this we should include the HTTP POST request, it helps by putting the parameters in the body of the request instead of the URL blocking attackers from viewing data in the referrer headers or server access logs. Creating access controls or authorizations provide limitations for users to access only the data that is permitted for use. Though it is not perfect is it worth putting time into building and utilizing all the features. Dynamic SQL statements will help reduce the possibility of SQL injections. We need to rely on “injection safe” database queries. In Java, parameterization can be provided with the following class: java.sql.PreparedStatement. Lastly, updating the dependencies which are the external Java libraries that our project uses, to the latest versions will help mitigate most of the vulnerabilities in the project.