****

# 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** | **07/17/2022** | **Raven Oyunjargal** | **Added main contents** |

## 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

Raven Oyunjargal

## 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?

Secure communications is critically important and valuable to the company because they handle variety of individual financial plans for savings, retirement, investments and insurance for their patrons. Having secure communications means they won’t lose or leak confidential data to the public.

* Are there any international transactions that the company produces?

They interact with patrons all over the world to prepare their financial plans so Artemis Financial produces many countries currency transactions and their economy information.

* Are there governmental restrictions about secure communications to consider?

Yes. Some governments wants the company to operate within their jurisdiction by their laws meaning secure communications may not be fully secure as the government wants their own citizen patron’s information. Restrictions include sharing information and communications data to the government or risk losing operations and legal lawsuits.

* What external threats might be present now and in the immediate future?

There is always the threat of people inside leaking data and potential security vulnerabilities that will allow attackers to breach in communications and database. In future, hackers will find new ways of breaching into the system from outside and these attacks needs to be counteracted with upgraded security measures and firewalls.

* What are the “modernization” requirements that must be considered, such as the role of open source libraries and evolving web application technologies?

Open source libraries are frequently updated with new improvements to the system but it also makes it vulnerable to attacks since anyone can contribute and they can use it. Web application technologies are getting simpler and easier to understand following modernization and security and secure communications must be made high priority to ensure people taking advantage of vulnerability and exploits can’t breach into the system. Determine vulnerabilities from documentation and make steps to block the vulnerability path until it is fixed by library developers.

## 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.

* Code quality and error: This area is important in every application and since the client servers many patrons from around the world, the code quality must be high grade and as error proof as possible at all times.
* Input validation: This is relevant as any application especially web application being accessible by anyone with internet access, input validation must be secure and ensure no attacker can breach into the system with input validation exploit or vulnerabilities.
* API: Company using producing variety of financial plans for different purposes, web application will be using APIs of various types such as currency and time zone. API transactions and information exchange needs to be secure to ensure exploits or vulnerabilities aren’t used.
* Client/Server: Web application has to be accessible with 99% uptime, by patrons and development, company team servers are run by cloud or traditional database with maintenance scheduled closely. It is important the servers are maintained regularly and updated with security and quality code.

## 3. Manual Review

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

* On some instances of object and variable calling, some input validation is needed.
* Other than it, code base looks secure and void of any vulnerability.
* ApplicationTests.java has empty function called contextLoads which is unnecessary.

## 4. Static Testing

Graphical user interface, text, application, email

Description automatically generated

Vulnerabilities

* Hibernate validator.jar - A flaw was found in Hibernate Validator version 6.1.2.Final. A bug in the message interpolation processor enables invalid EL expressions to be evaluated as if they were valid. This flaw allows attackers to bypass input sanitation (escaping, stripping) controls that developers may have put in place when handling user-controlled data in error messages.
* [jackson-databind-2.10.2.jar](file:///C:\Users\johno\eclipse-workspace\CS%20305%20Module%20Two%20Static%20Testing%20Code%20Base.zip_expanded\Module2.2\target\dependency-check-report.html#l4_0528de95f198afafbcfb0c09d2e43b6e0ea663ec) - jackson-databind before 2.13.0 allows a Java StackOverflow exception and denial of service via a large depth of nested objects.
* [log4j-api-2.12.1.jar](file:///C:\Users\johno\eclipse-workspace\CS%20305%20Module%20Two%20Static%20Testing%20Code%20Base.zip_expanded\Module2.2\target\dependency-check-report.html#l9_a55e6d987f50a515c9260b0451b4fa217dc539cb) - Apache Log4j2 versions 2.0-beta7 through 2.17.0 (excluding security fix releases 2.3.2 and 2.12.4) are vulnerable to a remote code execution (RCE) attack when a configuration uses a JDBC Appender with a JNDI LDAP data source URI when an attacker has control of the target LDAP server.

* [logback-core-1.2.3.jar](file:///C:\Users\johno\eclipse-workspace\CS%20305%20Module%20Two%20Static%20Testing%20Code%20Base.zip_expanded\Module2.2\target\dependency-check-report.html#l11_864344400c3d4d92dfeb0a305dc87d953677c03c) -In logback version 1.2.7 and prior versions, an attacker with the required privileges to edit configurations files could craft a malicious configuration allowing to execute arbitrary code loaded from LDAP servers.
* [mongo-java-driver-2.4.jar](file:///C:\Users\johno\eclipse-workspace\CS%20305%20Module%20Two%20Static%20Testing%20Code%20Base.zip_expanded\Module2.2\target\dependency-check-report.html#l12_f9bbd594f981d60f6a5b2d1ec3463b772355b64f) – No vulnerability found
* [snakeyaml-1.25.jar](file:///C:\Users\johno\eclipse-workspace\CS%20305%20Module%20Two%20Static%20Testing%20Code%20Base.zip_expanded\Module2.2\target\dependency-check-report.html#l14_8b6e01ef661d8378ae6dd7b511a7f2a33fae1421) - The Alias feature in SnakeYAML 1.18 allows entity expansion during a load operation, a related issue to CVE-2003-1564.
* [spring-boot-2.2.4.RELEASE.jar](file:///C:\Users\johno\eclipse-workspace\CS%20305%20Module%20Two%20Static%20Testing%20Code%20Base.zip_expanded\Module2.2\target\dependency-check-report.html#l15_225a4fd31156c254e3bb92adb42ee8c6de812714) - Spring-boot versions prior to version v2.2.11.RELEASE was vulnerable to temporary directory hijacking. This vulnerability impacted the org.springframework.boot.web.server.AbstractConfigurableWebServerFactory.createTempDir method. NOTE: This vulnerability only affects products and/or versions that are no longer supported by the maintainer.
* [spring-core-5.2.3.RELEASE.jar](file:///C:\Users\johno\eclipse-workspace\CS%20305%20Module%20Two%20Static%20Testing%20Code%20Base.zip_expanded\Module2.2\target\dependency-check-report.html#l16_3734223040040e8c3fecd5faa3ae8a1ed6da146b) - In spring framework versions prior to 5.3.20+ , 5.2.22+ and old unsupported versions, application with a STOMP over WebSocket endpoint is vulnerable to a denial of service attack by an authenticated user.

## 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.

### To remedy identified security vulnerabilities:

* Nested objects in Jackson-databind-2.10-2.jar shouldn’t have large depths. Their depth should be reduced significantly.
* Vulnerability in log4j-api-2.12.1.jar can be fixed by limiting JNDI data source names to the java protocol in Log4j2 versions 2.17.1, 2.12.4, and 2.3.2.
* LDAP servers execution arbitrary code needs to be updated so attacker couldn’t craft malicious configuration. Update it so they can’t create any configuration for people who have required privileges to edit configuration files.
* Remove entity expansion during load operation feature in snakeyaml-1.25.jar
* Spring-boot-2.2.4.RELEASE.jar – Vulnerability was fixed on next version, solution unnecessary.
* Spring-core-5.2.3.RELEASE.jar – This vulnerability was fixed on versions after 5.3.20+,5.2.22+.