CS 305 Project One

Document Revision History

Version	Date	Author	Comments
1.0	**07-20-25**	**Elizabeth Marticello**	**Revisions to the Code Base have not started yet. However, a code review and mitigation plan is attached below.**

Client

ARTEMIS FINANCIAL

Instructions

Submit this completed vulnerability assessment report. Replace the bracketed text with the relevant information. In this report, identify your security vulnerability findings and recommend the next steps to remedy the issues you have found.

- Respond to the five steps outlined below and include your findings.
- Respond using your own words. You may also include images or supporting materials. If you include them, make certain to insert them in the relevant locations in the document.
- Refer to the Project One Guidelines and Rubric for more detailed instructions about each section of the template.

Developer
Elizabeth Marticello

1. Interpreting Client Needs
Determine your client's needs and potential threats and attacks associated with the company's application and software security requirements. Consider the following questions regarding how companies protect against external threats based on the scenario information:

- What is the value of secure communications to the company?

Secure communications are integral to Artemis Financial. As a consulting company that manages clients' financial plans, they must have the latest and most efficient software security to keep client's savings, retirements, investments, insurance, and personal information safe.

- Are there any international transactions that the company produces?

It's safe to assume that as a financial consulting company Artemis Financial handles international transactions.

- Are there governmental restrictions on secure communications to consider?

There are numerous countries that follow restrictions on secure communications. Consider the EU Cybersecurity Act. It is not explicitly said that Artemis Financial does business in the European member states.

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

There are many different threats that could be present regarding Artemis Financial. Some include phishing attempts made to customers or employees, social engineering, and account takeovers.

- What modernization requirements must be considered, such as the role of open-source libraries and evolving web application technologies?

After assessing the current architecture updates can be made to better modernize the open-source libraries and web applications. Some goals would be to increase performance, ensure efficient scalability, and enforce security updates.

2. Areas of Security
Refer to the vulnerability assessment process flow diagram. Identify which areas of security apply to Artemis Financial's software application. Justify your reasoning for why each area is relevant to the software application.

Each of the following will apply to Artemis Financial's software application…
- *Input Validation: processes user input, which most likely is sensitive personal information.*
- *Cryptography: necessary for encrypting and protecting sensitive customer information.*
- *APIs will be necessary to interface between the client and servers.*

- *Client/Server: As a Financial Consulting company, customers will most likely be able to access their account information on an electronic device that communicates with Artemis' server.*
- *Code Error: Proper code handling will be necessary to ensure top-tier performance and security.*
- *Code Quality: High end code quality is mandatory to guarantee a functional and secure application.*
- *Encapsulation: Using encapsulation is paramount to providing a secure application by only allowing mandatory information to be shared with its components and users.*

3. Manual Review

Continue working through the vulnerability assessment process flow diagram. Identify all vulnerabilities in the code base by manually inspecting the code.

Below is a list of both vulnerabilities and strong points:

- *Models: Both Greetig.java and CRUD.java use FINAL variables without setters. This is good. myDateTime.java and DocData.java appear to be unfinished.*
- *Controllers: Both CRUDController.java and GreetingController.java use public methods. Could these be made private for better encapsulation?*
- *APIs- more on these vulnerabilities in the next section.*

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 the dependency-check report. Include the following items:

- The names or vulnerability codes of the known vulnerabilities
- A brief description and recommended solutions provided by the dependency-check report
- Any attribution that documents how this vulnerability has been identified or documented previously

Below are all the highest ranked severity vulnerabilities. All nine appear to be CVE-2025-8247. A vulnerability that affects an unknown part of the file /admin.php where manipulation can lead to sql injection and can be attacked remotely.

Dependency	Vulnerability IDs	Package	Highest Severity	CVE Count	Confidence	Evidence Count
log4j-api-2.12.1.jar	cpe:2.3:a:apache:log4j:2.12.1:*:*:*:*:*:*:*	pkg:maven/org.apache.logging.log4j/log4j-api@2.12.1	CRITICAL	5	Highest	46
spring-core-5.2.3.RELEASE.jar	cpe:2.3:a:pivotal_software:spring_framework:5.2.3:release:*:*:*:*:*:*:* cpe:2.3:a:vmware:spring_framework:5.2.3:release:*:*:*:*:*:*:*	pkg:maven/org.springframework/spring-core@5.2.3.RELEASE	CRITICAL	13	Highest	30
spring-expression-5.2.3.RELEASE.jar	cpe:2.3:a:pivotal_software:spring_framework:5.2.3:release:*:*:*:*:*:*:* cpe:2.3:a:vmware:spring_framework:5.2.3:release:*:*:*:*:*:*:*	pkg:maven/org.springframework/spring-expression@5.2.3.RELEASE	CRITICAL	14	Highest	30
spring-context-5.2.3.RELEASE.jar	cpe:2.3:a:pivotal_software:spring_framework:5.2.3:release:*:*:*:*:*:*:* cpe:2.3:a:vmware:spring_framework:5.2.3:release:*:*:*:*:*:*:*	pkg:maven/org.springframework/spring-context@5.2.3.RELEASE	CRITICAL	14	Highest	28
spring-webmvc-5.2.3.RELEASE.jar	cpe:2.3:a:pivotal_software:spring_framework:5.2.3:release:*:*:*:*:*:*:* cpe:2.3:a:vmware:spring_framework:5.2.3:release:*:*:*:*:*:*:*	pkg:maven/org.springframework/spring-webmvc@5.2.3.RELEASE	CRITICAL	14	Highest	30
spring-web-5.2.3.RELEASE.jar	cpe:2.3:a:pivotal_software:spring_framework:5.2.3:release:*:*:*:*:*:*:* cpe:2.3:a:vmware:spring_framework:5.2.3:release:*:*:*:*:*:*:*	pkg:maven/org.springframework/spring-web@5.2.3.RELEASE	CRITICAL	18	Highest	28
tomcat-embed-core-9.0.30.jar	cpe:2.3:a:apache:tomcat:9.0.30:*:*:*:*:*:*:*	pkg:maven/org.apache.tomcat.embed/tomcat-embed-core@9.0.30	CRITICAL	41	Highest	39
snakeyaml-1.25.jar	cpe:2.3:a:snakeyaml_project:snakeyaml:1.25:*:*:*:*:*:*:* cpe:2.3:a:yaml_project:yaml:1.25:*:*:*:*:*:*:*	pkg:maven/org.yaml/snakeyaml@1.25	CRITICAL	10	Highest	28
spring-boot-2.2.4.RELEASE.jar	cpe:2.3:a:vmware:spring_boot:2.2.4:release:*:*:*:*:*:*:* cpe:2.3:a:vmware:spring_framework:2.2.4:release:*:*:*:*:*:*:*	pkg:maven/org.springframework.boot/spring-boot@2.2.4.RELEASE	CRITICAL	14	Highest	32

5. Mitigation Plan

Interpret the results from the manual review and static testing report. Then identify the steps to mitigate the identified security vulnerabilities for Artemis Financial's software application.

- *Input validation could be implemented in the GreetingController.java by ensuring that "name" is formatted properly.*
- *Denial of Service could be implemented in the GreetingConteoller.java to limit check the entry.*
- *Some areas in the code base need to be completed.*
- *Other areas, like in both java controllers, have opportunities for methods to have more encapsulation.*
- *Due to the unknown origin of the critical risk vulnerability CVE-2025-8247. I would recommend a suspension of access to the Artemis servers until the issue can be properly investigated and resolved.*