Mohamed Babiker
CS-305

CS 305 Module Three Project one
Artemis Financial Vulnerability Assessment Report

Document Revision History

Version	Date	Author	Comments
1.0	09/14/2023	Mohamed Babiker	Good facts

Client

Instructions

Submit this completed vulnerability assessment report. Replace the bracketed text with the relevant information. In the report, identify your findings of security vulnerabilities and provide recommendations for 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 choose to include images or supporting materials. If you include them, make certain to insert them in all 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
Mohamed Babiker

1. Interpreting Client Needs

The effectiveness of Artemis Financial's communications is the cornerstone of its success. By safeguarding sensitive financial information, such as investment portfolios, retirement plans, and insurance data, they not only preserve their clients' confidence but also their reputation in a fiercely competitive market. In addition to preventing data breaches, Artemis Financial's commitment to security positions them in the eyes of their clients as a reliable and responsible partner, ensuring long-term success and development.
Like my parents in 2012 when they learnt about the Bank of America DDoS assault, customers nowadays are well aware of the need of secure connections. I can vividly recall how my parents made a big deal out of it because they were afraid they would lose everything.
In order to ensure safe communications, Artemis Financial must manage regulatory constraints and compliance within the financial industry. Additionally, dealing with region-specific rules adds complexity. Compliance protects client data as well as the business against fines and penalties.

Along with legislative difficulties, Artemis Financial also has to deal with external dangers including DDoS assaults, data breaches, phishing, and ransomware. The business must maintain a proactive security posture to combat these chronic and growing threats, strengthening the protection of sensitive financial data through robust defenses and ongoing monitoring.
Artemis Financial should proactively make use of cutting-edge web application technologies and open-source frameworks to modernize. These modernization initiatives have to include up-to-date software inventory maintenance, safe coding standards adherence, and ongoing security risk awareness training.

2. Areas of Security

The RESTful API acts as a gateway to the financial information and services provided by Artemis Financial. It is essential to ensure safe API interactions since it guards against risks including endpoint manipulation, data injection, and unauthorized access to private financial data. The confidentiality and integrity of financial data may be at danger due to an API security lapse. It's crucial to evaluate the API's authentication procedures, authorisation restrictions, and use of secure communication protocols in order to properly prevent external threats. Any flaws in these areas immediately affect the privacy and accessibility of financial data.
Data security in the financial sector requires cryptography. It guards against hacking, data theft, and eavesdropping and safeguards communications, customer data, and financial transactions. Failure of encryption may also result in financial fraud and unauthorized access to private financial information. A complete examination of cryptographic implementations, including key management, encryption algorithms, and certificate validation, is necessary to identify any potential weaknesses. Neglecting cryptography-related vulnerabilities might result in significant financial losses and reputational damage for the company.

3. Manual Review

Security flaws were discovered during my investigation of Project One's Code Base, particularly in the "DataProcessor" module's inadequate input validation. We want reliable input validation techniques, which should include limit analysis and data type checks. For API interactions and cryptography, the code also need industry-standard authentication and encryption. Although the small breadth of the code prevents a thorough architectural study, it provides insights. For increased production robustness, we should additionally include thorough error handling and logging. This strategy improves application security since it is based on industry best practices.

4. Static Testing

Prioritizing upgrading the packages linked to each of these vulnerabilities is advised in order to address them effectively. The list of packages and their corresponding severity levels are shown below:

- tomcat-embed-core-9.0.30.jar, Critical severity (22 CVEs)
- log4j-api-2.12.1.jar, Critical severity (5 CVEs)
- snakeyaml-1.25.jar, Critical severity (10 CVEs)
- spring-boot-2.2.4.RELEASE.jar, Critical severity (3 CVEs)
- jackson-databind-2.10.2.jar, High severity (6 CVEs)
- bcprov-jdk15on-1.46.jar, High severity (17 CVEs)
- spring-web-5.2.3.RELEASE.jar, High severity (4 CVEs)
- spring-beans-5.2.3.RELEASE.jar, High severity (1 CVE)
- logback-core-1.2.3.jar, Medium severity (1 CVE)
- hibernate-validator-6.0.18.Final.jar, Medium severity (1 CVE)
- spring-webmvc-5.2.3.RELEASE.jar, Medium severity (1 CVE)
- spring-context-5.2.3.RELEASE.jar, Medium severity (1 CVE)
- spring-expression-5.2.3.RELEASE.jar, Medium severity (3 CVEs)
-

5. Mitigation Plan

It is advised to simply upgrade them to the most recent versions for each of the detected dependencies mentioned below in order to resolve the security vulnerabilities:
the following jars: spring-boot-2.2.4.RELEASE.jar, log4j-api-2.12.1.jar, logback-core-1.2.3.jar, jackson-databind-2.10.2.jar, and hibernate-validator-6.0.18.Winter Beans for the Spring Web 5.2.3 Release: Jars For Spring Web MVC, publish Jars. 5.2.3 5.2.3.1 Spring Expression Release Jars 5.2.3.1
You may efficiently fix the known security flaws related to each of these dependencies and increase the security of your application by updating them to the most recent versions. due to the lack of more information.