CS 305 Project One

Document Revision History

Version	Date	Author	Comments
1.0	05/25/2025	David Droege	

Client

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
David Droege

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?
- Are there any international transactions that the company produces?
- Are there governmental restrictions on secure communications to consider?
- What external threats might be present now and in the immediate future?
- What modernization requirements must be considered, such as the role of open-source libraries and evolving web application technologies?

- Secure communication is crucial to Artemis Financial. The company works with clients' personal financial data. If the information about a client's banking information, account balance, or financial plans were to be leaked it could cause serious harm to the client.
- Considering that Artemis Financial uses an API for their services, it should be assumed that they would have international customers. Therefore, the assumption should be made that there are international transactions occurring.
- The company works with international clients to help them invest, save, retire, and buy insurance. Most of the services that the company offers are taxable events. It is important for Artemis Financial and their clients to be aware of taxes and tax laws surrounding the business and transactions.
- External threats can be individuals or groups that would try to intercept customer banking information or attempt to gain access to the Artemis Financial database.
- Using an API means that the program could constantly be updated and evolved. It is crucial for Artemis Investments to have continual upkeep and maintenance done on the program. It is important that all features keep up as any lagging in software versions can be turned into an exploit.

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.

- Input Validation
    - Artemis Financial should have rigorous checks that verify client input. Not verifying client input can lead to injection, which can cause access to sensitive information or compromise the program.
- API's
    - Artemis Financial uses an API to run their program, it is crucial that they tend to it and keep up with any updates that may come out. Not doing routine upkeep and maintenance leaves the program vulnerable.
- Cryptography

    - It is especially important when working with client banking information to keep it encrypted as it is transferred from client to server. Without encryption, client information is left unsecure and vulnerable to interception.
- Client/Server
    - Hand in hand with encryption, the interaction between client and server needs to be secure. The client needs to be protected, and the server needs to be protected, each in their own secure way.
- Code Error
    - Errors may seem harmless at times, however even a small error can lead to serious damage if taken advantage of.

3. Manual Review

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

- In DocData.java there are hardcoded credentials used for a database connection. A highly risky approach as it poses the risk of unauthorized access.
- In DocData.java key and value parameters are accepted as inputs which leaves the risk for injection.
- In CRUDController.java there is a lack of input validation which can allow for malicious inputs.
- In myDateTime.java there is also a lack of input validation which again can allow malicious inputs.
- In DocData.java stack trace logs can be exposed when the print command is called, leaving risk for a leak
- In GreetingController.java there is a lack of protection from Cross-Origin resource Sharing and Cross-Site Request Forgery, both lead to a false injection masked as a client.
- In CRUDController.java there is also a lack of protection from CORS and CSRF, abbreviation of the above-mentioned risks.

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

- Bcprov-jdk15on-1.46.jar
    a. Description: vulnerable to attacks, leaks, exposure, improper verification, poor generation, injection, and excessive computing consumption
    b. Solution: An update is required of the software that will patch the vulnerabilities
- Spring Framework
    a. Description: risks of injection, remote access, unauthorized access, and malicious file insertion

    b. Solution: An update of the Spring Framework is required, and input validation needs to be fortified
- Loback-core-1.2.3.jar
    a. Description: Risk of log manipulation
    b. Solution: An update of the logback software is required
- Log4j-api-2.12.1.jar
    a. Description: potential allowance of malicious log messages
    b. Solution: Update log4j and secure logging patterns
- Snakeyaml-1.25.jar
    a. Description: Risk of remote code execution
    b. Solution: An update of the SnakeYAML software is required and fortified input validation
- Jackson-databind-2.10.2.jar
    a. Description: Risk of remote code execution
    b. Solution: An update is required of Jackson Databind software
- Tomcat-embed-core-9.0.30.jar
    a. Description: risk of remote code execution
    b. Solution: An update is required of Tomcat software
- Hibernate-validator-6.0.18.Final.jar
    a. Description: lack of input validation and data manipulation
    b. Solution: An update of Hibernate Validator is required and fortified input validation

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.

- Manual Review Mitigation
    - The major risks that have been found when doing a manual review are that there is a lack of input validation and sensitive information is insecurely stored. The best solution is to thoroughly review the code and add input validation wherever it can be added. The other solution is to ensure that sensitive information is stored in separate files from where it is being called and ensure that there is a secure and impenetrable connection when being called.
- Static Review
    - The major risks that have been found through the static review is out-of-date software and lack of input validation. The solution is to ensure that all software being used in the service is held up to date. The other solution is to review the code and add input validation wherever it is possible to.
- Overview
    - Reviewing the program and finding all risks has enlightened that there is a serious need for input validation and routine upkeep and maintenance. Without the listed processes the system is vulnerable to injections and exploits, leaving client information at serious risk.