****

# 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** | **[Date]** | **[Your name]** |  |

## 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

Shayla Vincent

## Interpreting Client Needs

1. Since Artemis Financial is in the financial industry and they deal with savings, retirements, investments, and insurance it is paramount to have tight security. Any communications throughout the firm, either internal or external to clients, need to be secure to prevent any private information from being exposed.
2. The firm that I am a developer for, Global Rain, specializes in business around the world. Based on that and the fact that Artemis Financial is in the investment industry it is safe to say that they will have dealings internationally.
3. As someone that also has a background in finance, I know that governmental restrictions involving people’s personal information, especially money, are held to the highest standards. For example, in America, the company would have to comply with the standards set by the securities and exchange commission in order to be involved in securities. When dealing with a consulting company, there are many governmental rules to adhere to.
4. When dealing with financials there are many external threats to be aware of. This can include malware, phishing, ransomware, and many more.
5. The company must be aware of the ever-evolving nature of the internet. New threats and vulnerabilities can appear at any time. Open-source libraries are liable to negative changes and web applications must follow guidelines to avoid risk.

## Areas of Security

1. Input validation

Artemis Financial is an online consulting company that services clients. Clients will have a login page for their profile, which means that the company must have input validation to confirm the person’s identity. Having a multiple set login will help protect against malicious intruders.

1. APIs

We have been told that Artemis Financial has a RESTful web application programming interface so we must look out for secure interactions. There are many posable threats to an API. OWASP published a top 10 API security risk in 2023, that outlined threats such as third-party vendors, broken authentication mechanisms, improper inventory management, and more.

1. Cryptography/Encryption

To protect client and employee confidentiality the finance firm will need to encrypt its information. This can be done by converting information into a ciphertext and using an algorithm to transmit information back and forth. Online transactions will always need to be encrypted due to regulations.

1. Client/Server

When dealing with customers through a server, communication must be protected. Prevention and detection systems should be in place to protect both the client and the server.

1. Encapsulation

Keeping data secured from other processes in the system further protects information from leaking. Protecting systems from each other helps prevent a system-wide break.

## Manual Review

1. CRUD
   1. There needs to be input validation on the parameters for the two constructors.
2. CRUDController
   1. Input validation needed on the value.
3. Customer
   1. No protection or input validation to showInfo().
   2. Account\_balance should be private to avoid accidentally changing from other methods.
   3. Deposit needs an input validation for parameter.
   4. Since it is named deposit there should be a catch for a negative input
4. DocData
   1. The read\_document has a backend vulnerability. The MySql database connection includes a user’s username and password. Even an employee should never be able to access another user’s password.
   2. No input validation for read\_document parameters.
5. Greeting
   1. No input validation for constructors’ parameters
6. GreetingController
   1. No input validation for value
7. myDateTime
   1. No input validation for parameters which includes making sure that they enter int’s and it is nonnegative.
8. RestServiceApplication
   1. Input validation for args

## Static Testing

A screenshot of a computer

Description automatically generated

## Mitigation Plan

A lot of the dependencies and vulnerabilities that can be seen through the manual and static check are due to the lack of input validation procedures. To avoid those vulnerabilities a developer needs to make sure that any time an input is implemented they need to create a validation to determine if it is a valid input. This process is one in a mass list of general guidelines developers should follow when creating safe code. Here is a rough breakdown of a mitigation action plan:

1. Identify input sources.
2. Classify by sensitivity.
3. Create a list for valid inputs to check against.
4. Create an error page for invalid inputs.
5. Regularly update your dependency checker tool.

**References**

O. A. S. P. (2023). Owasp top 10 API security risks – 2023. OWASP Top 10 API Security Risks – 2023 - OWASP API Security Top 10. https://owasp.org/API-Security/editions/2023/en/0x11-t10/

Owasp Secure Coding Practices Quick Reference Guide. (n.d.). https://owasp.org/www-pdf-archive/OWASP\_SCP\_Quick\_Reference\_Guide\_v2.pdf