****

# 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

Dillon Belanger

## Interpreting Client Needs

**Secure Communication:**

Secure communications are valuable to preventing violation of the securities put in place. The company could be heavily impacted being a financial company holding sensitive information and data. Any breaches of communication could lead to the leaking of account information, and personal information of clients. This in turn would ruin the credibility of the company and potential clients in the future.

**International Transactions:**

Artemis Financial has international transactions, these are the connection between customers and their accounts throughout the world. Transactions across borders can be tricky as trying to be trustful to their customers can also get the company in legal trouble. As well as potential fraud being put in place from different areas which is common among international transactions.

**Government Restrictions:**

There are restrictions on transactions across borders and internationality, especially with the European Union. This involves the General Data Protection Regulation; this regulates communication. This requires the data handles and providers to make sure the data is protected and secure. So, the security of the data being used needs to be heavily monitored given these regulations.

**External Threats:**

For external threats, there is authentication that can be manipulated which allows unauthorized access to the systems, giving access to account details. There can be issues between the client and application which can also cause threats to happen. One of the bigger threats is denial of service threats. These involve copious amounts of traffic sent to the web application to bloat memory and resources of the system. APIs can have these threats, there is vulnerability in the API methods and tokens.

**Modernization:**

Modernization has come in many forms. One form is incremental modernization, changing legacy systems to be more manageable to lower risks of breaking. The other is an open source of modern technologies. So, making changes small so maintaining proper function is important. There is much flexibility to modernization which can upgrade systems in the future. The goal is making these technologies more accessible through open source.

## Areas of Security

There are many areas of importance that should be monitored within the application. These include API, Cryptography, Client/Server, and Secure Coding. API is important for the web application to utilize API calls. The configuration of the API must be secure to limit the potential of them being compromised in security and connection. In Cryptography it also applies to Artemis Financial given they access and transmit their data across the web. Encryption retains the confidentiality of the data for them and their customers. With Client/Server the communication between clients and the server is represented. The clients must communicate with the server backend and UI, being a considerable importance to security. Lastly, there is Secure Coding, with Artemis Financial having multiple instances of code sanitization being key. This maintains the logic of the business, if there are errors in the code and checks, then the application will fail.

## Manual Review

With the manual review, I was able to identify a vulnerability with CRUDController.java section. This can potentially expose the internal objects; these can be passed and accessed with code if injected. There was also a vulnerability with data access. For example, in the DOCData.java section, the method of access involves definition of a location. This could cause username and password vulnerability. The username and password are root, this allows them to be easily guessed by unauthorized users.

## Static Testing

Within this there were five dependencies noted. The first one noted was Bouncy Castle Crypto Package. There were various vulnerabilities that could disclose information, allowing malicious applications access to confidential data. A solution for this is to regularly update the application and operating system. The second dependency was Apache Log4j API. This involved a vulnerability with certificate validations, this could allow logs of messages to be revealed. The solution includes upgrading to a newer version, the newer version having built in verification. The third dependency was YAML 1.1. The vulnerability allows the entity to expand during operation. Limiting and failing the system of memory if not allocated correctly, a fix could be to migrate to SnakeYAML, which should solve the issue. The fourth dependency was Starter for reading and writing json. The vulnerability in this is the possibility of injecting commands. The solution to this is restrictions put into place with syntax of the bracketed parts of the loop strings. The final dependency was Core Tomcat, this had multiple vulnerabilities. Most were too many requests at once which kicks in denial of service. The fix for this dependency seems to be upgrading to a newer version to mitigate the issues. This is a crucial step as the high consumption of the memory could crash the system. So, keeping the data up to date will be of utmost importance.

## Mitigation Plan

For an effective mitigation to happen I would begin by addressing the data access and username vulnerabilities. This can be fixed with alphanumeric characters for usernames and passwords. This mitigates the risk of brute forces into the system. Next, to prevent other exploitation, updating the Apache Server to a newer version would be beneficial. Next is reviewing the code and modifying it to have more secure coding usage. The quality of the code should be enhanced for proper authentication. Furthermore, the issue of certificate validation should be checked. The code should be sanitized for proper validation and verification. This helps prevent exploitation and vulnerability more so. Lastly, implementing mutual checks on the client and server-side should be done using pinned certificates, this helps prevent vulnerability in such of the Bouncy Castle dependency etc.

**References:**

Apache. (2020). Add support for specifying an SSL configuration for SmtpAppender.

Bhalla, N., Sethi, R. K., Sivaranjan, R., Foroughi, E., & Whittington, G. C. (2020).

U.S. *Patent No. 10,706,156* Washington, DC: U.S. Patent and Trademark Office.

Hu, Q., Asghar, M. R., & Brownlee, N. (2019). Checking certificate revocation efficiently using certificate revocation guard. *Journal of Information Security and Applications*

48, 102356.

Mondschein, C. F., & Monda, C. (2019). The EU's General Data Protection Regulation (GDPR) in a research context. In *Fundamentals of Clinical Data Science* (pp. 55-71). Springer, Cham.

Mooij, A. J., Ketema, J., Klusener, S., & Schuts, M. (2020, February). Reducing Code Complexity through Code Refactoring and Model-Based Rejuvenation. In *2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER)*

(pp. 617-621). IEEE.

Rahman, R. U., Wadhwa, D., Bali, A., & Tomar, D. S. (2020). The Emerging Threats of Web Scrapping to Web Applications Security and Their Defense Mechanism. *In Encyclopedia of Criminal Activities and the Deep Web* (pp. 788-809). IGI Global.