****

# CS 305 Project One

**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** | **26 March 2022** | **Shingirai Chiremba** |  |

## Client



## Developer

Shingirai Chiremba

## 1. Interpreting Client Needs

**Determine your client’s needs and potential threats and attacks associated with their application and software security requirements. Consider the following 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 about secure communications to consider?**
* **What external threats might be present now and in the immediate future?**
* **What are the “modernization” requirements that must be considered, such as the role of open source libraries and evolving web application technologies?**

Every organization strives to achieve the best security for their systems and products, Artemis Financial is one of those many organizations that is trying to achieve this. Having a secure system means the organization is using secure protocols and secure data communication techniques to send and receive the data between the server and client. Enabling secure communications ensures that there is no loss of data for example, the data is not being tampered with by anyone in the middle of the communication and the trusted user gets it. The company may produce international transaction if someone access the data from their end. Currently, there are no government restriction on secure communication for an organization’s internal communications. The government promotes companies to have secure communications to protect the data and privacy of the user. However, the **Electronic Communications Privacy Act (ECPA)** [1986] - protects certain wire, oral, and electronic communications from unauthorized interception, access, use, and disclosure. These laws are primarily directed at data breeches, protecting organizations from cyber-attacks, entrapment and blackmail. External threats that are present now and in the future include but are not limited to the advancement of technology. Hackers only get better because the security gets better, offering these hackers an adrenaline rushing irresistible challenge to overcome. So, implementing controls against known threats like SANS 25 and OWASP Top 10 vulnerabilities for web-based applications should be a basic security requirement. Advancements made in software security have seen the introduction of web applications that work in sand-boxed mode to ensure great security and in addition, the company can also make use of open source library to add a layer of functionality to their program .

**2. Areas of Security**. **Referring to the Vulnerability Assessment Process Flow Diagram, identify which areas of security are applicable to Artemis Financials’ software application. Justify your reasoning for why each area is relevant to the software application.**

Essentially, all areas of the VAPFD are important and necessary to assess but for this company four are critical. These four are API, Cryptography, Client/Server and Encapsulation. **APIs** - If the command input function needs to be accessible remotely, the RESTful API that Artemis Financial has in place will need to be implemented to allow access to the functionality. The API needs to be integrated in a way that does not allow unexpected access to our systems. Input validation at the API level can also be assessed as a contingency. In this case, the areas of security are used for the web application that is needed for Artemis Financial RESTful API. A requirement must be met that everything is safe and secure by ensuring that the relationship is protected for the APIs limit to the potential of compromised security among the system and the connection of the API. Also, Application Programming Interface (API) indicates software negotiator that can enable two or more applications to communicate together. API contains open authorization (OAuth) that basically enables the users to provide the right to enter various web resources without necessarily the need to share passwords in third party state. OAuth is an internet protocol that is publicly available which is commonly used as a mode at which users are given access to a website or application for their information on another different website where they are not given the passwords. which users are given access to a website or application for their information on another different website where they are not given the passwords? **Cryptography** – the use of a RESTful API warrants securing data on both the client side and the server side. Proper cryptography needs to be implemented to secure data. Cryptography is used for the capability to access and transmit information through the internet. Without the proper encryption, it could result in a compromise of the integrity and confidentiality of their data or their customer’s data. Also, cryptography is the science for data protection by transferring it into a secure format through a process known as encryption. It helps prevent data from being decoded by any unintended persons. One attribute of cryptography is when the involved parties do not trust each other but they must cooperate to share the data where both enter some private inputs. Here, a secure implementation of cryptography is done after the computation of both parties where they know that each party has not changed or cheated. They include commitment schemes and various computational securities which enables both parties to trust the schemes used even if they do not trust each other. **Client/Server** – The implementation of RESTful API access means that Artemis Financial must ensure the proper certificates are used to ensure data is safe during transfer via https requests. The Client’s communicating has the backend of the system through the frontend UI. Also, client/server indicates the relationships between the various programs in an application that are comprised of clients or users that order requests that mainly offer the service to be provided. Client servers contain security features that can secure data and they include Physical security which means the protection of hardware, networks and data from actions that may be done physically in an organization across all components in the computer system. Change control security, which refers to the mode by which components can contain high systems which can control themselves from vulnerabilities that may occur. Artemis Financial ultimately will have a web-based platform with multiple users that all have valid access to the system. **Encapsulation** – command functions need to be able to access and retrieve data without conflicting with the existing data structure in other words, code sanitization is needed to maintain constancy in business logic application. Encapsulating indicates an exercise by which the computer software is developed in ways that protects against vulnerabilities that are to be introduced accidentally (secure coding). The commonly known defects brought about by the code include bugs and logical flows during the coding process. Analyzing of all cases that may have happened in vulnerabilities is done and professional can come upon vulnerabilities brought about by error of programming and with this they must identify code that has been done insecurely and here is where they find alternatives to the code. Basically, they rewrite the code but securely hence make it more secure in the software. Focusing on these 4 areas of security will ultimately ensure the highest of code quality is produced

## 3. Manual Review

**Continue working through the Vulnerability Assessment Process Flow Diagram. Identify all vulnerabilities in the code base by manually inspecting the code.**

There is a vulnerability within the DocData.java file, where the data access method is to access the data that involves the description of the location for the data in which is the username and password that has numerous possible vulnerabilities. The root users and the password are not recommendable for use by the root user. The root is used for the username and password and the root password could be simply deduced from just guessing and estimating. Unauthorized users can easily gain access into the system by using a brute force attack. There was another vulnerability within the CRUDController.java file, where the application could be exposing the inside objects and then being retrieved and passed through a sequence of code injection. This occurs while the “business name” value is passed through the CRUD method exposing the DocData object data access vulnerability.

## 4. Static Testing

**Run a dependency check on Artemis Financials’ software application to identify all security vulnerabilities in the code. Record the output from dependency check report. Include the following:**

1. **The names or vulnerability codes of the known vulnerabilities**
2. **A brief description and recommended solutions provided by the dependency check report**
3. **Attribution (if any) that documents how this vulnerability has been identified or documented previously**

**a. The names or vulnerability codes of the known vulnerabilities**.

• Bouncy Castle, CVE-2015-6644.

• Apache Log4j SMTP appender, CVE-2020-9488.

• Core Tomcat, CVE-2020-13935.

**b. A brief description and recommended solutions provided by the dependency check report.**

The Bouncy Castle allows for malicious application to access the private data stored on the database. The recommended solution for vulnerability is to frequently update the application among the operating system to prevent a compromise of the security of the application. The Apache Log4j SMTP appender allows for log messages to be revealed if they are sent through the appended. The recommended solution for vulnerability is to upgrade to the newer version being 2.13.2 and having a built-in feature for verification. The Core Tomcat allows for high consumption of memory that can crash the system. The recommended solution for vulnerability is to use an updated version to address the dependency and the vulnerabilities.

**c. Attribution (if any) that documents how this vulnerability has been identified or documented previously**.

Apache SMTP Log4j appender: Updating the server to 2.13.2 will help improve the support of the security measures. Also, it can prevent the setting of true value and make the application vulnerable. The Bouncy Castle: Updating the application among the operating system when avoiding such malicious applications for compromising security of the application. Core Tomcat: Updating to the correct validation found in the Apache Tomcat which will help support WebSocket Framework when triggering an infinite loop for DoS attacks.

## 5. Mitigation Plan

**After interpreting your results from the manual review and static testing, identify the steps to remedy the identified security vulnerabilities for Artemis Financials’ software application.**

The Mitigation plan will be broken down as follows:

**Data access username and password:**

The problem for data access username and password is creating strong combination of alphanumeric characters for both the username and password. This should help improve the risk from brute force attacks into the system.

**Code review and modification:**

The code review and modification, by integrate secure coding practices should help the quality of the code for having proper authentication and error catching when occurring. This should help mitigation of authentication vulnerabilities.

**Updating Apache server:**

The use of vulnerability CVE-2020-9488 by updating the Apache Server to the newer version. By updating the Apache Server, it will fix several of the vulnerabilities within the previous versions of the Apache Server.

**Certificate Validation:**

The problems of certificate validation are when code is being properly sanitized when allowing for correct validation and verification of all digital certificates for the application and webserver. This should help avoid the exploitation of vulnerability CVE-2020-9488 and CVE-2020-13935.

**TLS Certificate mutual checking:**

The problem of TLS Certificate mutual checking would apply to both the client and server-side through pinned certificates when mutual checking. This should help avoid compromising the requests of client API and help mitigate the potential vulnerabilities found in the Bouncy Castle dependency