****

# 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 | 01-23-2021 | Amber Foster | Provided information for the required areas. |

## Client



## Instructions

Deliver this completed vulnerability assessment report, identifying your findings of security vulnerabilities and articulating recommendations for next steps to remedy the issues you have found.

Respond to the five steps outlined below and include your findings. Replace the bracketed text on all pages with your own words. If you choose to include images or supporting materials, be sure to insert them throughout.

## Developer

Amber Foster

## 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?**

The Value of secure communication to the company is to increase the confidence of the customer. Secure communications can affect the privacy of customers and not just their emails.

* **Are there any international transactions that the company produces?**

Artemis Financials’ is a web-based company and that means that the company has internal transactions. So, the internal transactions entail different stocks from the clints buying and selling in order to make money.

* **Are there governmental restrictions about secure communications to consider?**

The governmental restrictions about the secure communications needs to be considered. As they play an important role in the securing the communications. But there are many securing communications when dealing with money, stocks, and retirements. It can allow hackers to gain digital access to all types of communications from emails to social media messages and to public cloud databases. Although there is one way to help secure communications from hackers. There is a bill that will not only delete privacy but adding a back door into every American communication. So, when an unwanted hacker wants to try and stealing private information and money, it will open a software to trap every unwanted hacker from commending crimes and reducing the risk of crimes and terrorism being made on every American.

* **What external threats might be present now and in the immediate future?**

The current threats that we are facing now and, in the future, include increasing hackers and emerging new techniques of data stealing. Also, another externalthreat is an increase in new viruses.

* **What are the “modernization” requirements that must be considered, such as the role of open-source libraries and evolving web application technologies?**

The modernization requirements contain the company to be fully web-based. This can benefit the company from an open-source library in order for the company to add more layers of security and functionality to the program.

## 2. Areas of Security

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

**API**:

Area of security is used for the web application that is needed for APIs. We need to require 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. OAuth is an internet protocol that is publicly available which is commonly used as a mode at which uses are given access to a website or application for their information on another different website where they are not given the passwords.

**Cryptography:**

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 have to 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:**

Client/Server is the communication between the client and server and takes on the tasks upon web applications. The Clients 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 are able to 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 are able contain high systems which are able to control themselves from vulnerabilities that may occur. Measures that are to be taken to make all systems secure.

**Secure Coding:**

Secure Coding applies to the application for many situations where code sanitization is needed to maintain constancy in business logic application. Also, secure coding indicates an exercise by which the computer software is developed in ways that protects against vulnerabilities that are to be introduced accidentally. 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.

## 

## 3. Manual Review

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

**Data Access:**

I was able to find 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 of the root user. The root is used for the username and password and the root password could be simply deduced from just guesstimating. Also, unauthorized users can easily gain access the system by using a brute force attack.

**Direct Object Reference:**

I was able to find a 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. 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 Financial’s 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.**

* Bouncy Castle, CVE-2015-6644.
* Apache Log4j SMTP appender, CVE-2020-9488.
* Core Tomcat, CVE-2020-13935.

1. **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.

1. **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 (Source, 2016).
* 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 Financial’s software application.

**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 Certificatemutual 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.

**MY REFERENCES**

Manico, Jim. (2014). Oracle Press. *“Iron-Clad Java”*.

Rahalkar, Sagar. (August 2018). Packt Publishing. *“Network Vulnerability Assessment”.*

Lawrence, Shari & Pfleeger, Charles P; (August 2011). Pearson. *“Analyzing Computer Security: A Threat/ Vulnerability/ Countermeasure Approach”*.

Vacca, John R. (May 2017). Morgan Kaufmann. “*Computer and Information Security Handbook,* *Third Edition”*.

Viega, John; Gary McGraw (2001). *"Building Secure Software: How to Avoid Security Problems the Right Way"*. (Secure Coding).

Whitson, Gordon. (2016). *"Understanding OAuth: What Happens When You Log into a Site with Google, Twitter, or Facebook"*. (API).

Christenson, P. (July 2013). *"Authentication and Cryptography"*. (Cryptography).

Dustdar, S.; Schreiner, W. (2005). *"A survey on web services composition"*. (Client/Server).

NIST. (May 2020). NATIONAL VULNERABILITY DATABASE From https://nvd.nist.gov/vuln