****

# 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** | **January 25, 2022** | **Joaquin Esguerra Jr** |  |

## 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

Joaquin Esguerra Jr

## 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?

With Artemis Financial primarily handling funds for its clients along with putting financial plans in place on their behalf, the information they hold is highly sensitive and is considered valuable to outside attackers or threats. Based on this information, secure communications within the company and with its clients is of great importance. Being a financial institution, Artemis Financial is required to adhere to government regulations and restrictions. All communications and transactions must fall under the restrictions and regulations or as a result they too can become a risk to securities for the organization and its cliental. Potential threats or attackers would primarily be seeking to gain information from both the client side as well as the organization. Ensuring the API is secure is a starting point of defense, without a secure API, information can be leaked through the security structure. (Ekanayake 2020) Another line of defense would be “Two-Step Verification”. This verification works to prevent wrongful login attempts and requires all communication to be conducted through HTTPS with sensitive information alternating back and forth through response headers.

## 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.

To best defend against potential attacks, secure coding is considered a best practice and relevant for Artemis Financial to be able to hold to their organization on client privilege. The structure set up with the code builds upon the level of security present. (Gonzalez 2019) The organization must handle all code errors that are presented and if multiple logins in attempts occur a flag must be created on the organization’s end to prevent further action. As a result of the web services use of RESTful API, APIs would be relevant along with the need for a secure communication level. RESTful API also takes in user input which in turn makes input validation relevant because user input, on the client side, must be validated on the organizations side in a secure manner.

## 

## 3. Manual Review

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

* Non validated service requests create vulnerabilities to outside threats
* HTTPS is not in use when sharing of sensitive information
* CRUDController class holds the business names which are sent as request parameters which can lead to further vulnerabilities as this information can be leaked to threats
* Verification lacks requirements for authentication due to systems not in place

## 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

* bcprov-jdk15on-1.46.jar – several vulnerabilities on 1.46, update to latest version.

CVE-2013-1624

CVE-2015-6644

CVE-2015-7940

CVE-2016-1000338

CVE-2016-1000339

CVE-2016-1000341

CVE-2016-1000342

CVE-2016-1000343

CVE-2016-1000344

CVE-2016-1000345

CVE-2016-1000346

CVE-2016-1000352

CVE-2017-13098

CVE-2018-1000613

CVE-2018-5382

CVE-2020-15522

CVE-2020-26939

* Hibernate-validator-6.0.18.Final.Jar - one vulnerability, update to latest version.

CVE-202-10693

* Jackson-databind-2.10.2.jar - one vulnerability, update to latest version.

CVE-2020-25649

* Log4j-api-2.12.1.jar – one vulnerability, update to latest version.

CVE-2020-9488

* Logback-core-1.2.3.jar

CVE-2021-42550

* Snakeyalm-1.25.jar – one vulnerability, update to latest version.

CVE-2017-18640

* Spring-aop5.2.3.RELEASE.jar

CVE-2020-5421

CVE-2021-22060

CVE-2021-22118

* Spring-core-5.2.3.RELEASE.jar – one vulnerability, update to latest version.

CVE-2020-5421

CVE-2021-22060

CVE-2021-22096

CVE-2021-22118

* Tomcat-embed-core-9.0.30.jar – several vulnerabilities, update to latest tomcat version.

CVE-2019-17569

CVE-2020-11996

CVE-2020-13934

CVE-2020-13935

CVE-2020-13943

CVE-2020-17527

CVE-2020-1935

CVE-2020-1938

CVE-2020-9484

CVE-2021-24122

CVE-2021-25122

CVE-2021-25329

CVE-2021-30640

CVE-2021-33037

CVE-2021-41079

CVE-2021-42340

* Tomcat-embed-websocket-9.0.30.jar - several vulnerabilities, update to latest tomcat version.

CVE-2019-17569

CVE-2020-11996

CVE-2020-13934

CVE-2020-13935

CVE-2020-13943

CVE-2020-17527

CVE-2020-1935

CVE-2020-1938

CVE-2020-8022

CVE-2020-9484

CVE-2021-24122

CVE-2021-25122

CVE-2021-25329

CVE-2021-30640

CVE-2021-33037

CVE-2021-41079

CVE-2021-42340

Graphical user interface, application

Description automatically generated

## 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.

Artemis Financials must convert to HTTPS to remedy current and future issues. This conversion will not only remedy issues but also keep the organization and client’s info secure as all direct communication will be blocked from snooping. Artemis Financial will also need to convert request parameters to headers. The parameters are less secure for their visibility and the encryption in the headers will provide more security. Along with the previous two conversions, Artemis Financials will need to dig into its hard coded database to remove the credentials which contain all business names. Implementing and enabling two-step verification to support general authentication systems will protect client’s information along with blocking potential outside threats from gaining access to the organization and client’s sensitive information. As a final move to protect against threats, Artemis Financial will need to utilize the provide list from the dependency check and process all necessary updates.

References

Ekanayake, C. (2020, December 28). *An overview on API security*. Medium. Retrieved from https://medium.com/sa-team-blog/an-overview-on-api-security-1e7bf69f7877

Gonzalez, K., Kenneth Gonzalez Security Intelligence Analyst, Gonzalez, K., Analyst, S. I., & Kenneth Gonzalez is a contributor for SecurityIntelligence. (2019, February 24). *A step-by-step guide to vulnerability assessment*. Security Intelligence. Retrieved from https://securityintelligence.com/a-step-by-step-guide-to-vulnerability-assessment/

*Introduction to HTTPS*. The HTTPS-Only Standard. (n.d.). Retrieved from https://https.cio.gov/faq/

*Software testing: Static testing*. GeeksforGeeks. (2019, May 13). Retrieved from https://www.geeksforgeeks.org/software-testing-static-testing/