****

# 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** | **03/15/2022** | **Douglas New** |  |

## 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

Douglas New

## 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?

The value of secure communication is it will help the customer trust the company and give them confidence in the company. The need for secure communication would be critical as they are dealing with people money whether it is financial future or peace of mind that property is covered there needs to be trust and confidence.

There is no specification of international transactions but the fact that it is web based would say there would be transactions occurring internationally.

There are restrictions set by the government to ensure a secure communication since the company would be dealing with insurance, stocks, retirement, and other financials.

As with any web-based program there is always a chance for a threat especially hackers that want to cause harm to various structures in the financial industry. Then we have threats of other competitions that want to receive the most business at any cost.

The main requirement with the modernization would be the need to increase security in the program as it will be web based. With the open-source library, they could utilize the feature to have more security protocols that would counter threats a lot easier and more effectively.

## 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.

API: Since the system will be web-based it will need a secure configuration so there are not threats to the system and it will help ensure a secure interface.

Cryptography: There would be a need for encryption since there would be data transfer across the internet, so the information is not easily compromised while it is not in a secure server or personal computer.

Client/Server: Since the communication between the company and the customer is across the web this is an area that also needs to be checked.

Secure Coding: The secure coding applies since the logistics of the system needs to be consistent and uniform with the code being secure.

## 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 are two things I have found during my manual review the first is an issue in the CRUDControllee that has Direct Object Referencing on line 13 value=”business\_name” this action could cause a vulnerability to DocData and show its internal objects. The second is in the DocData file were the username and password are set root which is not a good idea as it can be easily figured out and thus hacked by brute force attack.

## 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
4. The Bouncy Castle Crypto package

* bcprov-jdk15on-1.46.jar
* CVE Code: CVE-2013-1624
* CVE Code: CVE-2015-6644
* CVE Code: CVE-2015-7940
* CVE Code: CVE-2016-1000338
* CVE Code: CVE-2016-1000339
* CVE Code: CVE-2016-1000341
* CVE Code: CVE-2016-1000342
* CVE Code: CVE-2016-1000343
* CVE Code: CVE-2016-1000344
* CVE Code: CVE-2016-1000345
* CVE Code: CVE-2016-1000346
* CVE Code: CVE-2016-1000352
* CVE Code: CVE-2017-13098
* CVE Code: CVE-2018-1000613
* CVE Code: CVE-2018-5382
* CVE Code: CVE-2020-15522
* CVE Code: CVE-2020-26939
* CVE count: 17
* Severity: Unknown
* Evidence: 38

1. Hibernate's Bean Validation (JSR-380) reference implementation.

* hibernate-validator-6.0.18.Final.jar
* CVE Code: CVE-2020-10693
* CVE count: 1
* Severity: medium
* Evidence: 34

1. General data-binding functionality for Jackson: works on core streaming API

* jackson-databind-2.10.2.jar
* CVE Code: CVE-2020-25649
* CVE count: 1
* Severity: high
* Evidence: 41

1. The Apache Log4j API

* log4j-api-2.12.1.jar
* CVE Code: CVE-2020-9488
* CVE count: 1
* Severity: low
* Evidence: 44

1. logback-core module

* logback-core-1.2.3.jar
* CVE Code: CVE-2021-42550
* CVE count: 1
* Severity: medium
* Evidence: 33

1. YAML 1.1 parser and emitter for Java

* snakeyaml-1.25.jar
* CVE Code: CVE-2017-18640
* CVE count: 1
* Severity: high
* Evidence: 46

1. Spring AOP

* spring-aop-5.2.3.RELEASE.jar
* CVE Code: CVE-2020-5421
* CVE Code: CVE-2021-22060
* CVE Code: CVE-2021-22096
* CVE Code: CVE-2021-22118
* CVE count: 4
* Severity: high
* Evidence: 33

1. Spring Core

* spring-core-5.2.3.RELEASE.jar
* CVE Code: CVE-2020-5421
* CVE Code: CVE-2021-22060
* CVE Code: CVE-2021-22096
* CVE Code: CVE-2021-22118
* CVE count: 4
* Severity: high
* Evidence: 36

1. Core Tomcat implementation

* tomcat-embed-core-9.0.30.jar
* CVE Code: CVE-2019-17569
* CVE Code: CVE-2020-11996
* CVE Code: CVE-2020-13934
* CVE Code: CVE-2020-13935
* CVE Code: CVE-2020-13943
* CVE Code: CVE-2020-17527
* CVE Code: CVE-2020-1935
* CVE Code: CVE-2020-1938
* CVE Code: CVE-2020-9484
* CVE Code: CVE-2021-24122
* CVE Code: CVE-2021-15122
* CVE Code: CVE-2021-25329
* CVE Code: CVE-2021-30640
* CVE Code: CVE-2021-33037
* CVE Code: CVE-2021-41079
* CVE Code: CVE-2021-42340
* CVE count: 16
* Severity: critical
* Evidence: 33

1. Core Tomcat implementation

* tomcat-embed-websocket-9.0.30.jar
* CVE Code: CVE-2019-17569
* CVE Code: CVE-2020-11996
* CVE Code: CVE-2020-13934
* CVE Code: CVE-2020-13935
* CVE Code: CVE-2020-13943
* CVE Code: CVE-2020-17527
* CVE Code: CVE-2020-1935
* CVE Code: CVE-2020-1938
* CVE Code: CVE-2020-8022
* CVE Code: CVE-2020-9484
* CVE Code: CVE-2021-24122
* CVE Code: CVE-2021-25122
* CVE Code: CVE-2021-25329
* CVE Code: CVE-2021-30640
* CVE Code: CVE-2021-33037
* CVE Code: CVE-2021-41079
* CVE Code: CVE-2021-42340
* CVE count: 17
* Severity: critical
* Evidence: 32

## 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 first is the username/password the user’s data access credentials need to be set up to store separately from the root, so it is not easily accessible with in the database itself. The password also needs to be unique and not contain anything from the username or personal known information.

Next is the code needs to be revised with secure coding by utilizing the use of authentication as well as being able to show and report any errors in the system.

Next the server version needs updating as this would solve majority of the dependency issues that were found by the test. Since most of the results show a newer version that has less security issues.

Next the cryptographic issue the certificate should be validated this would ensure the site has minimum vulnerabilities that could be exploited.

Last a TLS certification needs to be implemented to protect the client and the server by preventing the client API request from being compromised.