****

# 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** | **5/22/2022** | **Nicholas Boodoo** |  |

## 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

Nicholas Boodoo

## 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?

Artemis Financial is a finance group that has access to the financial information of all of its customers and clients. Because of this, the data maintained by this organization is very sensitive in nature and is highly sought by attacks since it will give them access to the financial data to many customers and clients. Financial institutions also have their guidelines and are under strict government regulations which help to not only protect the financial institutions themselves, but also their clients. Also if these transactions are happening across borders, Artemis Financial would also need to adhere to other countries’ regulations as well regarding financial transactions. Because of all these factors, security is the number one priority. Many attackers will be targeting these types of institutions because of the data that they have access to. The entire system should be monitored extremely well as well as unknown login attempts, two factor authentication, HTTPS certificates and other security tools to ensure data integrity and security.

## 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.

Regarding Artemis Financial’s software application, I think that API, cryptography, and the client server are the most important aspects of the Vulnerability Assessment Process Flow Diagram. The API will be the main line of defense against attackers. The API must be secure and allow login methods such as two factor authentication. If there are breaches in the API, then the entire system will be compromised. Cryptography allows the system to be encrypted and prevent unauthorized access. Cryptography will prevent data from being accessed or even prevent data from being decoded by the wrong individuals. And lastly the client server relationship is very important because the sever will need to create certificates and authenticate data from the client.

## 3. Manual Review

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

The code does not use HTTPS, which allows for secure communication over a network. This may leave room for attackers to find breaches and find their way into the system. Anytime a login in involved, HTTPS at minimum should be used to ensure data security. There are no types of authentication in place. Authentication is important for verification purposes. Requests are also not validated in this code. This means that anybody can find their way into the system. Also I think that it is a big security risk to include names in the request parameters. This can leak potentially sensitive data about clients to unauthorized users.

## 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
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

Published vulnerabilites include: **CVE-2018-1000613, CVE-2015-6644,** [**CVE-2016-1000338,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000338)[**CVE-2016-1000342,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000342)[**CVE-2016-1000343,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000343)[**CVE-2016-1000344,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000344)[**CVE-2016-1000352,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000352)[**CVE-2016-1000341,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000341)[**CVE-2016-1000345,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000345)[**CVE-2017-13098,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2017-13098)[**CVE-2020-15522,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-15522)[**CVE-2016-1000339,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000339) **CVE-2020-26939,** [**CVE-2015-7940,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2015-7940)[**CVE-2018-5382,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2018-5382)[**CVE-2013-1624,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2013-1624) **CVE-2016-1000346.**

These vulnerabilities appear to be exploitable and will allow access to code and sensitive data. These vulnerabilities appear to have been fixed in newer versions of the file, so updates will be required. These issues have been fixed by updating to 1.60 and later.

### hibernate-validator-6.0.18.Final.jar

Published vulnerabilities include: [**CVE-2020-10693.**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-10693)

This vulnerability allows unauthorized access because it shows invalid certificates as valid. We can update to the latest version here and it should fix the issue.

### jackson-databind-2.10.2.jar

Published vulnerabilities include: [**CVE-2020-25649,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-25649)[**CVE-2020-36518**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-36518)

This vulnerability allows XML attacks. This is a very high threat level since it compromises data integrity. We can update to the latest version here and it should fix the issue.

### log4j-api-2.12.1.jar

Published vulnerabilities include: **CVE-2020-9488**

This vulnerability causes improper validation of certificate. This can allow for unauthorized access to the system. Fixed in Apache Log4j 2.12.3 and 2.13.1.

### logback-core-1.2.3.jar

Published vulnerabilities include: **CVE-2021-42550**

This vulnerability gives attackers the required privileges to edit configuration files. We can update to the latest version here and it should fix the issue.

### snakeyaml-1.25.jar

Published vulnerabilities include: [**CVE-2017-18640**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2017-18640)

This vulnerability allows expansion during a load operation. We can update to the latest version here and it should fix the issue.

### spring-aop-5.2.3.RELEASE.jar

Published vulnerabilities include: [**CVE-2016-1000027,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000027)[**CVE-2022-22965,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22965)[**CVE-2021-22118,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22118)[**CVE-2020-5421.**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-5421)[**CVE-2022-22950,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22950)[**CVE-2022-22968,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22968)[**CVE-2021-22060,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22060)[**CVE-2021-22096**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22096)

This vulnerability causes code execution errors along with a multitude of Spring Framework issues. We will need to update to the latest Spring AOP.

### spring-boot-2.2.4.RELEASE.jar

Published vulnerabilities include: **CVE-2022-27772**

This vulnerability is vulnerable to temporary directory hijacking. This vulnerability only affects products and/or versions that are no longer supported by the maintainer.

### spring-core-5.2.3.RELEASE.jar

Published vulnerabilities include: [**CVE-2016-1000027,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000027)[**CVE-2022-22965,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22965)[**CVE-2021-22118,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22118)[**CVE-2020-5421,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-5421)[**CVE-2022-22950,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22950) **CVE-2022-22968,** [**CVE-2021-22060,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22060) **CVE-2021-22096**

This vulnerability is causes code execution errors that can leave the system vulnerable. Updating to the latest Spring Core should resolve the vulnerability.

### tomcat-embed-core-9.0.30.jar

Published vulnerabilities include: [**CVE-2020-1938,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-1938)[**CVE-2020-11996,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-11996)[**CVE-2020-13934,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13934)[**CVE-2020-13935,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13935)[**CVE-2020-17527,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-17527)[**CVE-2021-25122,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-25122)[**CVE-2021-41079,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-41079) **CVE-2021-42340,** [**CVE-2022-29885,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-29885)[**CVE-2020-9484,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-9484)[**CVE-2021-25329,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-25329)[**CVE-2021-30640,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-30640)[**CVE-2021-24122,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-24122)[**CVE-2021-33037,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-33037)[**CVE-2019-17569,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2019-17569)[**CVE-2020-1935,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-1935) **CVE-2020-13943**

These vulnerabilities cause improper privilege management which can allow unauthorized access into the system. Updating Tomcat Core should solve the vulnerability issues.

### tomcat-embed-websocket-9.0.30.jar

Published vulnerabilities include: [**CVE-2020-1938,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-1938)[**CVE-2020-8022,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-8022)[**CVE-2020-11996,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-11996)[**CVE-2020-13934,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13934)[**CVE-2020-13935,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13935)[**CVE-2021-25122,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-25122)[**CVE-2021-41079,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-41079)[**CVE-2022-29885,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-29885)[**CVE-2020-9484,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-9484)[**CVE-2021-25329,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-25329)[**CVE-2021-30640,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-30640)[**CVE-2021-24122,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-24122)[**CVE-2021-33037,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-33037)[**CVE-2019-17569,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2019-17569)[**CVE-2020-1935,**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-1935)[**CVE-2020-13943**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13943)

These vulnerabilities cause improper privilege management which can allow unauthorized access into the system and can be compromised. Updating Tomcat Core should solve the vulnerability issues.

## 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.

First and foremost, we need to ensure that the system is secure and communication is happening only over a secure network, therefore implementing HTTPS would be the logical action. Regarding the request parameters, those should be switched to headers and scrubbed of all client names to prevent data leaks. We will also need to implement a minimum of a two factor authentication system for every single login, which must be verified with the appropriate certificates before a user is allowed access the system. We will also need to perform regular updates so that we have all of our files up to date to reduce the amount of vulnerabilities that are in our system.