****

# 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 5](#_Toc32574612)

[3. Manual Review 5](#_Toc32574613)

[4. Static Testing 5](#_Toc32574614)

[5. Mitigation Plan 7](#_Toc32574615)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **1/19/2021** | **Michael Linsenbigler** | **Initial Assessment and Findings** |

## 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

Michael Linsenbigler

## 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?

Secure communications is most likely the most valuable need of Artemis Financial. Being a firm that performs international financial consulting, the information being handled is critically important. Data such as account and social security numbers, as well as income and personal client information would be a large part of daily communications. If this data were to be compromised, it could cost the company irreversible financial damage and also damage their professional reputation as a trusted institution.

Artemis Financial does produce international transactions and therefore need to be aware of threats and attacks that target those specific areas. Connections originating from and passing through countries around the world can be more difficult to regulate due to variances in laws and transparency that is provided by different countries. This is another reason that the software and security protocols can manage the connections and data transfers properly.

There are absolutely governmental restrictions in place regarding secure communications both locally and globally. There are legal and business requirements that dictate what information can be viewed, shared and also what can be stored and in what capacity. There are also regulations that apply levels of responsibility to companies that utilize sensitive customer data that place the burden of keeping the information secure on the company. An example of this is the General Data Protection Regulation (GDPR) which is a regulation that requires businesses to protect the personal data and privacy of EU citizens for transactions that occur within EU member states. (Nadeau, 2020)

External threats that may exist now and in the immediate future for Artemis Financial are based around unauthorized access to sensitive data. Tactics such as “man-in-the-middle” are common with international sessions having an unauthorized node connected and recording information that passes through. Denial of service (DoS) attacks can also present opportunities for another party to exploit a security weakness by taking down the software system and gain entry via an alternate communication path or during the system’s restoral process. Since the company uses an API for communications and transactions, all threats and risks that target those web interfaces are applicable here.

In order to keep the company’s systems secure and as up-to-date as possible with emerging technology trends, consideration must be made to how the software is developed now. Using open source libraries is a great way to utilize pre-coded packages to carry out standard system functions but these libraries need to be sourced properly and updated as any potential threats are identified and patched. Good coding practices such as good class references and a modular structure can allow for updated API technologies to be swapped out for more efficient modules without requiring a complete re-coding of the project. Basically, ensuring the project is developed with a secure, modular framework that allows for future updates and improvements should be a guiding principle.

## 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.

One major area of focus for Artemis Financial should be to provide secure API interactions. The API is the main source of interaction with their clients and it is what gives them access to the company’s resources. Making sure all communication within the API framework is private and secure is critical.

The use of cryptography in the development of software is paramount for this project. Not only to ensure the security of highly sensitive data but also realizing that cryptography is regulated as an export item in this case as Artemis Financial does business internationally.

The functionality of the system is communication between a server and client, therefore server/client vulnerabilities need to be taken into consideration. These may overlap with the some of the same practices involved with API security such as authentication and access control.

Proper code quality impacts all areas of this project and needs to be of the highest quality to ensure all code functions to improve the customer experience, drive revenue and not cause any functional or security hurdles for the company. Utilizing the highest quality code practices will allow for the most secure transactions as well as a logical and efficiently operating business core.

## 3. Manual Review

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

Upon manual inspection, I have identified a few major security vulnerabilities related to access control within the existing code base. The first one is located in the CRUDController.java class line 13, where “business name” is requested from the client. This leaves the DocData object open to injection threats where an unauthorized source can substitute another value into this field and potentially compromise the database. The same situation exists for GreetingController.java line 16 where a request is made for “name”.

Another security risk I found was in the DocData.java class line 27 having the username and password set to “root” and “root”. This is an extremely dangerous data combination and can be very easily guessed by anyone looking to gain access by simply trying common default values.

## 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 : CVE-2018-5382**

Bouncy Castle BKS version 1 keystore (BKS-V1) files use an HMAC that is only 16 bits long, which can allow an attacker to compromise the integrity of a BKS-V1 keystore. All BKS-V1 keystores are vulnerable. Bouncy Castle release 1.47 introduces BKS version 2, which uses a 160-bit MAC.

**Recommendations:** Issue can be mitigated with a software patch / update

**Attribution:** [RHSA-2014:0371 - Security Advisory](https://access.redhat.com/errata/RHSA-2014:0371)

**jackson-databind-2.10.2.jar : CVE-2020-25649**

A flaw was found in FasterXML Jackson Databind, where it did not have entity expansion secured properly. This flaw allows vulnerability to XML external entity (XXE) attacks. The highest threat from this vulnerability is data integrity.

**Recommendations:** Issue can be mitigated with a software patch / update

**Attribution:** [CVE-2020-25649 FasterXML Jackson Databind Vulnerability](https://security.netapp.com/advisory/ntap-20210108-0007/)

**log4j-api-2.12.1.jar : CVE-2020-9488**

Improper validation of certificate with host mismatch in Apache Log4j SMTP appender. This could allow an SMTPS connection to be intercepted by a man-in-the-middle attack which could leak any log messages sent through that appender.

**Recommendations:** Issue can be mitigated with a software patch / update

**Attribution:** [[LOG4J2-2819] Apache.org](https://issues.apache.org/jira/browse/LOG4J2-2819)

**snakeyaml-1.25.jar : CVE-2017-18640**

The Alias feature in SnakeYAML 1.18 allows entity expansion during a load operation

**Recommendations:** Issue can be mitigated with a software patch / update

**Attribution:** [[SECURITY] Update: snakeyaml-1.26-1.fc31 fedoraproject.org](https://lists.fedoraproject.org/archives/list/package-announce@lists.fedoraproject.org/message/PTVJC54XGX26UJVVYCXZ7D25X3R5T2G6/)

**spring-core-5.2.3.RELEASE.jar : CVE-2020-5421**

In Spring Framework versions 5.2.0 - 5.2.8, 5.1.0 - 5.1.17, 5.0.0 - 5.0.18, 4.3.0 - 4.3.28, and older unsupported versions, the protections against RFD attacks from CVE-2015-5211 may be bypassed depending on the browser used through the use of a jsessionid path parameter.

**Recommendations:** Issue can be mitigated with a software patch / update

**Attribution:** [CVE-2020-5421: RFD Protection Bypass via jsessionid](https://tanzu.vmware.com/security/cve-2020-5421)

**tomcat-embed-core-9.0.30.jar : CVE-2019-17569**

The refactoring present in Apache Tomcat 9.0.28 to 9.0.30, 8.5.48 to 8.5.50 and 7.0.98 to 7.0.99 introduced a regression. The result of the regression was that invalid Transfer-Encoding headers were incorrectly processed leading to a possibility of HTTP Request Smuggling if Tomcat was located behind a reverse proxy that incorrectly handled the invalid Transfer-Encoding header in a particular manner. Such a reverse proxy is considered unlikely

**Recommendations:** Issue can be mitigated with a software patch / update

**Attribution:** [Oracle Critical Patch Update Advisory - January 2021](https://www.oracle.com/security-alerts/cpujan2021.html)

**tomcat-embed-websocket-9.0.30.jar : CVE-2020-9484**

When using Apache Tomcat versions 10.0.0-M1 to 10.0.0-M4, 9.0.0.M1 to 9.0.34, 8.5.0 to 8.5.54 and 7.0.0 to 7.0.103 if a) an attacker is able to control the contents and name of a file on the server; and b) the server is configured to use the PersistenceManager with a FileStore; and c) the PersistenceManager is configured with sessionAttributeValueClassNameFilter="null" (the default unless a SecurityManager is used) or a sufficiently lax filter to allow the attacker provided object to be deserialized; and d) the attacker knows the relative file path from the storage location used by FileStore to the file the attacker has control over; then, using a specifically crafted request, the attacker will be able to trigger remote code execution via deserialization of the file under their control. Note that all of conditions a) to d) must be true for the attack to succeed.

**Recommendations:** Issue can be mitigated with a software patch / update

**Attribution:** [McAfee Security Bulletin - (CVE-9484)](https://kc.mcafee.com/corporate/index?page=content&id=SB10332)

## 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.

**CRUDController.java class line 13**

The value should be provided by a server side access control database and not retrieved from a client side request.

**GreetingController.java line 16**

The value should be provided by a server side access control database and not retrieved from a client side request.

**DocData.java class line 27**

The username and password should be created using a strong combination of letters/numbers/special characters and be managed by an access control policy module.

**bcprov-jdk15on-1.46.jar : CVE-2018-5382**

update to Bouncy Castle release 1.47 or higher

**jackson-databind-2.10.2.jar : CVE-2020-25649**

update the jackson-databind package to version 2.10.5 or 2.11 and higher

**log4j-api-2.12.1.jar : CVE-2020-9488**

* upgrade to 2.13.2 which supports this feature
* previous versions can set the system property mail.smtp.ssl.checkserveridentity to true to globally enable hostname verification for SMTPS connections.

**snakeyaml-1.25.jar : CVE-2017-18640**

* update SnakeYAML 1.18 with a patch provided via open source
* upgrade to any version beyond 1.18

**spring-core-5.2.3.RELEASE.jar : CVE-2020-5421**

affected version should be updated or upgraded to 5.2.9, 5.1.18, 5.0.19, 4.3.29

**tomcat-embed-core-9.0.30.jar : CVE-2019-17569**

update to tomcat version 9.0.31

**tomcat-embed-websocket-9.0.30.jar : CVE-2020-9484**

* upgrade to Apache Tomcat 10.0.0-M5, 9.0.35, 8.5.55, 7.0.104 or later
* configure the PersistenceManager with an appropriate value for sessionAttributeValueClassNameFilter to ensure that only application provided attributes are serialized and deserialized.

**Citations:**

Nadeau, M. (2020, June 12). What is the GDPR, its requirements and facts? Retrieved January 19, 2021, from https://www.csoonline.com/article/3202771/general-data-protection-regulation-gdpr-requirements-deadlines-and-facts.html