****

# 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 5](#_Toc32574613)

[4. Static Testing 6](#_Toc32574614)

[5. Mitigation Plan 9](#_Toc32574615)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **7/10/2023** | **Christopher Anderson** |  |

## Client



## Instructions

Submit this completed vulnerability assessment report. Replace the bracketed text with the relevant information. In the report, identify your findings of security vulnerabilities and provide recommendations for the next steps to remedy the issues you have found.

* Respond to the five steps outlined below and include your findings.
* Respond using your own words. You may also choose to include images or supporting materials. If you include them, make certain to insert them in all the relevant locations in the document.
* Refer to the Project One Guidelines and Rubric for more detailed instructions about each section of the template.

## Developer

Christopher Anderson at Global Rain

## Interpreting Client Needs

After reading about Artemis Financial’s urge to modernize their RESTful web application

programming interface (API) I understand that heightening system security is the top priority for this firm. It may be assumed that Artemis Financial is a large international firm that must maintain tight security around all communications relevant to their various projects. It is paramount to understand that savings, retirement, investments, and insurance planning involves a great amount of government regulation involving tax codes, confidentiality agreements regarding certain business deals and given the scale of Artemis Financial’s operation it is likely that there are hundreds of millions of dollars of liability to be accounted for. Given the value of Artemis Financial’s services there will be hackers that attempt to steal or extort confidential client data and financial plans from Artemis Financial. Therefore, Artemis Financial’s RESTful web API must have its security measures modernized to ensure that there are ample defenses for their products and services.

## Areas of Security

The most important areas of security for Artemis Financial’s RESTful web API are secure API interactions, strong Cryptography, and proper Encapsulation. Since this application will be hosted online, it is vital that all user interactions with the API are completely secure and that these interactions cannot be intercepted or altered by various hacking threats. Cryptography also plays an essential role in the defense of this system since this company will be in constant communication with international clients that may be prone to hackers and other criminal entities attempting to intercept their system’s messages. For this system, Encapsulation will be paramount to the defense of data within the system as this will prevent external functions from accessing sensitive class information. Although other areas of security are apparent within Artemis Financial’s RESTful web API, the essential areas for defense are the API, Encapsulation and Cryptography.

## Manual Review

* In myDateTime.java’s myDateTime class, the integer variables named mySecond, myMinute and myHour are not marked as private and could be altered by a hacker accessing the system.
* In customer.java’s customer class, the integer variable named account\_balance is not marked as private that could result in a hacker intercepting an active user and easily accessing their account balance.
* In pom.xml, the version of the Spring Framework being utilized for the boot starter parent is the 2.2.4 release, but the newest version is the 3.1.1 release that includes a variety of updates including REST and Java EE 6 support.
* In pom.xml, the version of the Maven Dependency Check utilized is 5.3.0 yet the newest is version 8.3.1 this could cause discrepancies since some vulnerabilities may not be detected or may be inaccurately rated as lesser threats than they are by the older version of the Maven Dependency Check.
* In pom.xml, version 1.46 of bouncy castle is being utilized when the latest version is 1.75 that includes updates to the encryption packages along with other increases to the systems efficiency.

## Static Testing

A screenshot of a computer

Description automatically generated

* Dependency Name: bcprov-jdk15on-1.46.jar, Vulnerability ID:   
  cpe:2.3:a:bouncycastle:bouncy-castle-crypto-package:1.46:\*:\*:\*:\*:\*:\*:\*,  
  cpe:2.3:a:bouncycastle:bouncy\_castle\_crypto\_package:1.46:\*:\*:\*:\*:\*:\*:\*,  
  cpe:2.3:a:bouncycastle:legion-of-the-bouncy-castle-java-crytography-api:1.46:\*:\*:\*:\*:\*:\*:\*, and  
  cpe:2.3:a:bouncycastle:the\_bouncy\_castle\_crypto\_package\_for\_java:1.46:\*:\*:\*:\*:\*:\*:\* is a package of cryptographic algorithms implemented in Java that includes vulnerabilities that will allow invisible data to be introduced and provides opportunity for an attacker to recover private encryption keys.
* Dependency Name: hibernate-validator-6.0.18.Final.jar, Vulnerability ID: cpe:2.3:a:redhat:hibernate\_validator:6.0.18:\*:\*:\*:\*:\*:\*:\* is a Hibernate Bean Validator file that describes system constraints in when modeling in the Java EE6 platform, this file introduces a vulnerability where attackers can bypass input validator controls and this vulnerability is corrected in version 6.0.20 of the hibernate\_validator file.
* Dependency Name: jackson-databind-2.10.2.jar, Vulnerability ID: cpe:2.3:a:fasterxml:jackson-databind:2.10.2:\*:\*:\*:\*:\*:\*:\*, and

cpe:2.3:a:fasterxml:jackson-modules-java8:2.10.2:\*:\*:\*:\*:\*:\*:\* is a file relating to data-binding in the core streaming API that allows for a Java StackOverflow error and denial of service attacks in pre-version 2.13.4 releases.

* Dependency Name: log4j-api-2.12.1.jar, logback-core-1.2.3.jar, and snakeyaml-1.25.jar, Vulnerability ID: cpe:2.3:a:apache:log4j:2.12.1:\*:\*:\*:\*:\*:\*:\*, cpe:2.3:a:qos:logback:1.2.3:\*:\*:\*:\*:\*:\*:\*, and cpe:2.3:a:snakeyaml\_project:snakeyaml:1.25:\*:\*:\*:\*:\*:\*:\* are files relating to parsing data and emitting data for Java, the Apache API, and the logback core module that present vulnerabilities for man-in-the-middle attacks, and DoS attacks. It is recommended that Log4j version 2.12.3, logback version 1.2.8, and SnakeYaml version 2.0 or the latest versions be utilized.
* Dependency Name: spring-boot-2.2.4.RELEASE.jar, Vulnerability ID: cpe:2.3:a:vmware:spring\_boot:2.2.4:release:\*:\*:\*:\*:\*:\* is the primary boot file for Spring version 2.2.4 that introduces a temporary directory hijacking vulnerability that is mitigated by implementing version 2.2.11 or later of the spring boot file.
* Dependency Name: spring-boot-starter-web-2.2.4.RELEASE.jar, Vulnerability ID: cpe:2.3:a:vmware:spring\_boot:2.2.4:release:\*:\*:\*:\*:\*:\*, and

cpe:2.3:a:web\_project:web:2.2.4:release:\*:\*:\*:\*:\*:\* is the Spring Framework’s boot file for building web applications that introduces a temporary directory hijacking vulnerability that is mitigated by implementing version 2.2.11 or later of the spring boot file.

* Dependency Name: spring-core-5.2.3.RELEASE.jar, Vulnerability ID: cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*,

cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*, and

cpe:2.3:a:vmware:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\* is the core operating file for the Spring Framework version 5.2.3 that introduces vulnerabilities for remote code execution, privilege escalation, reflected file download attacks and denial of service attacks that can only be remedied through careful version control for spring core and it’s referenced files.

* Dependency Name: spring-data-rest-webmvc-3.2.4.RELEASE.jar, Vulnerability ID: cpe:2.3:a:pivotal\_software:spring\_data\_rest:3.2.4:release:\*:\*:\*:\*:\*:\*, and

cpe:2.3:a:vmware:spring\_data\_rest:3.2.4:release:\*:\*:\*:\*:\*:\* is a data file relating to RESTful web application architectures that presents vulnerabilities for attacks to uncover hidden entity attributes should they know the structure of the domain model, this vulnerability is dealt with by implementing version 3.7.3 or later.

* Dependency Name: spring-web-5.2.3.RELEASE.jar, Vulnerability ID: cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*,

cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*,

cpe:2.3:a:vmware:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*, and

cpe:2.3:a:web\_project:web:5.2.3:release:\*:\*:\*:\*:\*:\* is a data file specific to the Spring Framework’s design of web applications that allows for potential remote code execution, privilege escalation, reflected file download attacks, and denial of service attacks that are mitigated by utilizing release 6.0.8 of spring web.

* Dependency Name: spring-webmvc-5.2.3.RELEASE.jar, Vulnerability ID: cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*,

cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*,

cpe:2.3:a:vmware:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*, and

cpe:2.3:a:web\_project:web:5.2.3:release:\*:\*:\*:\*:\*:\* is another Spring web application file that is susceptible to remote code execution and privilege escalation that is correct by version 6.0.8.

* Dependency Name: tomcat-embed-core-9.0.30.jar, Vulnerability ID: cpe:2.3:a:apache:tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*, and

cpe:2.3:a:apache\_tomcat:apache\_tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\* is the Apache Tomcat system’s core operating file that is vulnerable to DDoS style attacks, HTTP Request Smuggling, Jakarta Server Pages source code disclosure, and an exploit that allows attackers to bypass protections. These vulnerabilities can be mitigated through version control with implementation of the latest release of software packages including version 8.5.91 of Apache Tomcat.

* Dependency Name: tomcat-embed-websocket-9.0.30.jar, Vulnerability ID: cpe:2.3:a:apache:tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*, and

cpe:2.3:a:apache\_tomcat:apache\_tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\* is another core operating file for the Tomcat system that is subject to memory leaks, denial of service exploits, an incorrect default permissions vulnerability, HTTP Request/Response Smuggling Attacks, and a vulnerability that allows an attacker to be authenticated allowing protections to be bypassed.

## Mitigation Plan

There are several options to mitigate the vulnerabilities discovered during the Manual Review of the

Artemis Financial system. Within myDateTime.java’s myDateTime class all data variables should be set to private to ensure that the myDateTime class’ data is secure and similarly in customer.java’s customer class the account\_balance variable must be designated as private to promise that a customer’s account balance cannot be accessed by external classes and functions. From the pom.xml file, the system will strongly benefit from referencing the latest versions of bouncy castle encryption, the Spring Framework boot starter parent file, and the Maven Dependency Check. A basic step toward a secure software package is to ensure that the principle of Encapsulation is utilized and that all software packages referenced are up to date.

Although there are many vulnerabilities that were detected during the Static Testing phase, the solution to all these complications is to employ consistent version control. The concept of version control is vital to the security of the Artemis Financial software package since all the vulnerabilities that have been uncovered are only found in older, or isolated, versions of the software being referenced. While conducting version control for the software packages referenced it is also vital to check compatibility between the different versions of Apache Tomcat, the Spring Framework and Bouncy Castle to ensure that new weaknesses are not being introduced to the software’s architecture. The solution to all complications uncovered during the Static Testing phase for the Artemis Financial software package is consistent version control.