****

# 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** | **3/18/2024** | **Nicholle Caudy** | **This document is geared toward assessing threats to Artemis Financial with a mitigation plan to help prevent vulnerabilities within their system. A code review and dependency assessment are also available in this document.** |

## 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

Nicholle Caudy

## Interpreting Client Needs

Artemis Financial is a consulting company that individualizes financial plans for their customers that include plans for investments, insurance, retirement, and savings. This company is interested in modernizing their operations and using the most effective and current software security. They need advice on how to protect their organization from external threats. The value of secure communications with this company is vital because of the personal information they work with for their customers. The data being sent back and forth between the customer and the client contains private/personal information, setting up the proper safeguards to ensure attacks and hackers can gain this information is invaluable to both the organization and its customers. There is no indication that Artemis Financial will be working with international transactions, but the code and system should be set up to allow these transactions to occur securely. Setting up the system to connect to the International Transaction Log may also be added. There are currently no government restrictions on secure communication that I could find although there are many plans in the works and bills are passed all the time. We should set up the code now to ensure that all communication is secured to the highest degree to avoid having to make any changes to stay within any new government guidelines and to better protect the client and their customers. Artemis Financial will be working with personally sensitive information to help their clients with financial planning, this will put them at a much higher risk of attacks. This will more likely come from a single entity or smaller group trying to gain this personal information for financial gain. These attacks can include hacking, phishing, malware, ransomware, and brute force attacks. We also need to ensure that the principle of least privilege is provided for this system to prevent anyone from having access to the administrative functions. Some modernized requirements to consider are active maintenance checks for bugs or weakened security codes, and open-source libraries for reusable code instances for security updates. Web applications have come a long way over the years and are a great way to conduct business, but it also opens more vulnerabilities and security risks so extra steps need to be taken to ensure all areas of vulnerability are covered and safeguards in place if something does slip through.

## Areas of Security

The areas of security from the Vulnerability Assessment Process Flow include:

* Input Validation/Secure Input and Representation- Input validation for any system that requires input is crucial. This program has the potential for user input and this needs to be validated because they are entering something into the system and if doing so maliciously it could cause extreme damage or allow unauthorized access to personal data.
* APIs/Secure API Interactions- A secure API will be an important process for this application because it is designed for an environment like a web browser. Authorization and authentication will be vital for this application to protect the client and its customers. I recommend a two-step verification for users and administrators to further secure the information within the system. Using the principle of least privilege for this application is also essential.
* Cryptography/Encryption Use and Vulnerabilities- Cryptography is used to help protect confidential information and check for the integrity and authenticity of data. When used/coded properly an encryption algorithm data can be transferred securely by transforming in a way that makes it impossible to read for anyone who does not know the secret key. This will be vital because private/personal data will be transferred from one place to another and needs to be encrypted to avoid this information being stolen.
* Code Error/Secure Error Handling- Improper error handling can lead to several security problems for a website such as stack traces, database dump, and error codes that are displayed to the user that could potentially be a hacker. These error code messages could reveal implementation details that should not be made public. The error handling mechanism that should be used should handle any set of inputs while enforcing proper security and focus on any errors that can be generated by internal components.
* Code Quality/Secure Coding Practice/Patterns- Code quality is an area for concern because an API and input validation are present. Ensuring that we have the best code quality will ensure that no data is accidentally exposed as well as ensure that methods that are restricted from end users stay that way, meaning that the authorized users will get the data and methods that are available by their tier of authorization (principle of least privilege).

## Manual Review

Using the Vulnerability Assessment Process Flow Diagram specifically the areas I chose for this application to manually review the code provided there were some areas for improvement. While looking for an Apache Validator I checked the pom.xml file and greeting controller class file. I found that the input in the greeting controller file had no validators built in. I could not get any output. I could not find any API within the files of the application. This needs to be addressed as this will be an outside environment application and the input that this accepts needs to be handled in a secure way and without risk of being leaked to another http. There was no encryption present in this application. This may be because it is not specified whether there will be international communications but if in the future this is the plan then it needs to be developed to comply with any regulations and laws. I also looked for error handling, in the Doc.Data.java class there is a try/catch block, but this does not have any error handling application. There were no other classes that had any error handling that I could find. This needs to be corrected to ensure that error code messages do not reveal any implementation details and enforce proper security for any errors that may occur internally. The code quality for what is present is okay but needs to be expanded for functionality. The API makes nothing available to the end user, input validation needs to be implemented, error handling needs to be added, and for the future growth of the company and added security we need to implement some cryptography/encryption.

## Static Testing

|  |  |  |
| --- | --- | --- |
| **Dependency and Vuln IDs** | **Description** | **Mitigation** |
| Bcprov-jdk15on-1.46.jar  [cpe:2.3:a:bouncycastle:legion-of-the-bouncy-castle-java-crytography-api:1.46:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Abouncycastle&cpe_product=cpe%3A%2F%3Abouncycastle%3Alegion-of-the-bouncy-castle-java-crytography-api&cpe_version=cpe%3A%2F%3Abouncycastle%3Alegion-of-the-bouncy-castle-java-crytography-api%3A1.46) | Version 1.55 and earlier of Bouncy Castle JCE Provider causes a leak in information on the AES key and AESEngine, does not fully validate ASN.1, has possibly vulnerability to timing attacks, possible vulnerability of injected extra elements in the sequence making up the signature, generate a weak private key if used with default values, the DH public key is not fully validated, and the ECIES implementation allows the use of a mode that is regarded as unsafe. | Use the newest version of Bouncy Castle JCE Provider – version 1.8 |
| Spring-boot-2.2.4.RELEASE.jar  [cpe:2.3:a:vmware:spring\_boot:2.2.4:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Avmware&cpe_product=cpe%3A%2F%3Avmware%3Aspring_boot&cpe_version=cpe%3A%2F%3Avmware%3Aspring_boot%3A2.2.4) | Spring boot versions earlier than v2.2.11.RELEASE is vulnerable to temporary directory hijacking, versions 3.0.0-3.0.5 and 2.7.0-2.7.10 or older are potentially susceptible to security bypasses, and versions 3.0.0-3.0.6, 2.7.0-2.7.11, 2.6.0-2.6.14, 2.5.0-2.5.14 or older have a risk for denial-of-service attacks if Spring MVC is used together with a reverse proxy cache. | Update the version of Spring Boot to 3.0.6+ or 2.7.11+ |
| Logback-core-1.2.3.jar  [cpe:2.3:a:qos:logback:1.2.3:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aqos&cpe_product=cpe%3A%2F%3Aqos%3Alogback&cpe_version=cpe%3A%2F%3Aqos%3Alogback%3A1.2.3) | Version 1.2.7 and earlier in logback allows an attacker with the proper privileges to edit configuration files allowing them to maliciously configure and execute arbitrary code loaded from LDAP servers. Logback version 1.4.11 has a vulnerability of attackers to mount a Denial-of-Service attack by sending poisoned data. | Update the version of logback to 1.2.9 |
| Log4j-api-2.12.1.jar  [cpe:2.3:a:apache:log4j:2.12.1:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aapache&cpe_product=cpe%3A%2F%3Aapache%3Alog4j&cpe_version=cpe%3A%2F%3Aapache%3Alog4j%3A2.12.1) | Host mismatch with improper validation of certificate in Apache Log4j SMTP appender allowing an SMTPS connection to be intercepted by an attack called man-in-the-middle which could leak log message sent through that appender. JNDI features for configuration, parameters, and log messages do not protect against attacker-controlled LDAP. This version of Apache Log4j2 does not protect from uncontrolled recursion from self-referential lookups allowing attackers with control over TCM data to cause a DOS. Using this older version also vulnerable to RCE attacks when a configuration uses a JDBC Appender. | Use the newest version of Apache Log4j2- 2.17.1, 2.12.4, and 2.3.2 |
| Snakeyaml-1.25.jar  [cpe:2.3:a:snakeyaml\_project:snakeyaml:1.25:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Asnakeyaml_project&cpe_product=cpe%3A%2F%3Asnakeyaml_project%3Asnakeyaml&cpe_version=cpe%3A%2F%3Asnakeyaml_project%3Asnakeyaml%3A1.25)  [cpe:2.3:a:yaml\_project:yaml:1.25:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Ayaml_project&cpe_product=cpe%3A%2F%3Ayaml_project%3Ayaml&cpe_version=cpe%3A%2F%3Ayaml_project%3Ayaml%3A1.25) | Versions of SnakeYaml before 1.31 are vulnerable to DoS. Using SnakeYaml’s Constructor () class does not restrict types that can be instantiated during deserialization, this can allow the deserializing yaml content used by an attacker can lead to remote code execution. | Upgrade to version 2.0+ and use SnakeYaml’s SafeConstructor if parsing untrusted content. |
| Jackson-databind-2.10.2.jar  [cpe:2.3:a:fasterxml:jackson-databind:2.10.2:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Afasterxml&cpe_product=cpe%3A%2F%3Afasterxml%3Ajackson-databind&cpe_version=cpe%3A%2F%3Afasterxml%3Ajackson-databind%3A2.10.2) | Entity expansion is not secured properly due to a flaw found in FasterXML Jackson Databind, this allows vulnerability to XXE attacks and data vulnerability. Using this version of Jackson Databind can cause a Java StackOverflow exception and DoS. | Use the newest version of Jackson Databind- 2.17.0 |
| Tomcat-embed-core-9.0.30.jar  [cpe:2.3:a:apache:tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aapache&cpe_product=cpe%3A%2F%3Aapache%3Atomcat&cpe_version=cpe%3A%2F%3Aapache%3Atomcat%3A9.0.30)  [cpe:2.3:a:apache\_tomcat:apache\_tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aapache_tomcat&cpe_product=cpe%3A%2F%3Aapache_tomcat%3Aapache_tomcat&cpe_version=cpe%3A%2F%3Aapache_tomcat%3Aapache_tomcat%3A9.0.30) | Using this version of Apache Tomcat could allow for invalid payload lengths that could trigger an infinite loop which could ultimately lead to DoS. It could also allow for duplicate request headers, allowing a leak between headers. It is vulnerable to attacks where the attacker authenticates using variations of a valid username or to bypass some of the protection provided by the LockOut Realm. | Upgrade to version 9.0.44+ of Apache Tomcat |
| Hibernate-validator-6.0.18.Final.jar  [cpe:2.3:a:redhat:hibernate\_validator:6.0.18:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aredhat&cpe_product=cpe%3A%2F%3Aredhat%3Ahibernate_validator&cpe_version=cpe%3A%2F%3Aredhat%3Ahibernate_validator%3A6.0.18) | Using this version of Hibernate Validator allows attackers to bypass input sanitation controls. | Use the newest version of Hibernate Validator- 6.2.4.Final |
| Spring-web-5.2.3.RELEASE.jar  Spring-beans-5.2.3.RELEASE.jar  Spring-webmvc-5.2.3.RELEASE.jar  Spring-context-5.2.3.RELEASE.jar  Spring-expression-5.2.3.RELEASE.jar  [cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Apivotal_software&cpe_product=cpe%3A%2F%3Apivotal_software%3Aspring_framework&cpe_version=cpe%3A%2F%3Apivotal_software%3Aspring_framework%3A5.2.3)  [cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aspringsource&cpe_product=cpe%3A%2F%3Aspringsource%3Aspring_framework&cpe_version=cpe%3A%2F%3Aspringsource%3Aspring_framework%3A5.2.3) | 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 of Spring Framework may allow the protections against RFD attacks to be bypassed. These older versions may also make it possible for a user to input maliciously to cause the insertion of additional log entries. WebFlux applications are vulnerable to privilege escalations due to using an older version of Spring Framework. This older version also makes it possible for a user to provide a SpEL expression that has the potential to cause a DoS condition. | To avoid these issues use the version 5.3.20 of Spring Framework |

## Mitigation Plan

After interpreting the results from the manual review and the static testing some steps to remedy the identified security vulnerabilities for Artemis Financial’s application is to become security focused now as we continue to build this application to useability. This includes input validations, error code handling, APIs, and cryptography as discussed above. We need these areas to be completed in a very secure way as all the information this application will be handling will be personal/private and we need to protect their information using all our abilities. Using the updated and newest versions of Bouncy Castle, Spring Boot, logback, Apache log4j, Jackson Databind, Apache Tomcat, Hibernate Validator, and Spring Framework will allow us to take advantage of the patches that have been put in place with all the newest version and avoid vulnerabilities. We also need to use the newest version of Snakeyaml and use Snakeyaml’s SafeConstructor. The newest versions are all listed in the table above.

References:

* N. (2018, January 20). *The Kyoto Protocol*. Retrieved March 20, 2024, from <https://unfccc.int/process/the-kyoto-protocol/registry-systems/international-transaction-log>
* Graham, Cotton, & Blackburn (2020, June 23). *Graham, Cotton, Blackburn Introduce Balanced Solution to Bolster National Security, End Use of Warrant-Proof Encryption that Shields Criminal Activity*. Retrieved March 21, 2024, from <https://www.judiciary.senate.gov/press/rep/releases/graham-cotton-blackburn-introduce-balanced-solution-to-bolster-national-security-end-use-of-warrant-proof-encryption-that-shields-criminal-activity>
* WH.GOV (2021, June 9). *Executive Order on Protecting Americans' Sensitive Data from Foreign Adversaries*. Retrieved March 21, 2024, from <https://www.whitehouse.gov/briefing-room/presidential-actions/2021/06/09/executive-order-on-protecting-americans-sensitive-data-from-foreign-adversaries/>
* Detlefsen, J., & Kenan, K. (2014). *Iron-Clad Java: Building Secure Web Applications* (pp. chapters1, 3, 7, 10). McGraw Hill Computing. <https://learning.oreilly.com/library/view/iron-clad-java/9780071835886/ch01.html>
* (2024, March 18). *Dependency-Check*. Retrieved March 18, 2024, from <file:///C:/Users/nikki/eclipse-workspace/rest-service/target/dependency-check-report.html>