****

# 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** | **7/16/2023** | **Crystal Berkhan** |  |

## 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

Crystal Berkhan

## Interpreting Client Needs

Artemis Financial is a consulting company that develops individualized plans for their customers, including savings, retirement, investments, and insurance. The company wants to modernize their operations, protect their organization from external threats, and utilize the most current and effective software security.

Secure communications are extremely critical to this company. Artemis Financial handles very private and sensitive data pertaining to their clients, such as tax information and social security numbers; therefore, it is important to ensure the client’s information is secure and protected. It Is currently unknown whether the company makes any international transactions; however, it is safe to assume the company does make international transactions.

The Gramm-Leach Bliley Act rule and the Safeguards Rule are two governmental restrictions the company needs to abide by. The Gramm-Leach Bliley Act, “requires financial institutions to inform customers about their information-sharing practices and allow customers to opt out of having their information shared with certain third parties” (Newman & Richie, 2022). The Safeguards Rule includes, “more specific criteria for what safeguards financial institutions must implement as part of their information security program such as limiting who can access consumer data and using encryption to secure the data” (Newman & Richie, 2022).

Due to the company harboring sensitive financial information for their clients, the information of the clients is most at risk. Cyber security attacks targeting private information could cause significant harm to customers, such as identity theft or monetary loss.

The company should maintain a regular maintenance schedule to resolve any bugs or issues with the software, update the software as needed, and conduct security protocols.

## Areas of Security

The areas of security that apply to Artemis Financial’s web application include input validation, APIs, code quality, code error, and cryptography. Input validation protects users by requesting validation of the account owner and aids in identifying malicious attempts to enter invalid data. The quality of the code needs to be up to date with standard practices to ensure user’s data is not at risk and the application’s performance is not weakened by bugs. APIs are useful in securing the data as it is transferred, reduces the risk of outside access, and implements encryption, authentication, and access controls. Cryptography reduces the risk of unintended access to information and allows for the hiding/coding of information.

## Manual Review

After reviewing the current code, I noticed the Crud file, Greeting file, Greeting Controller file, DocData file and the Customer file does not require input validation and it lacks any error handling. There is not any sort of authentication system in place to verify users and prevent unwanted access to information. The customer file also does not have the “account\_balance” field as private. In the Crud Controller file, the RequestMapping method does not perform input validation on the “business\_name”.

## Static Testing

A screenshot of a computer

Description automatically generated

[Dependency-Check Report](file:///C:\Users\Cryst\eclipse-workspace\rest-service\target\dependency-check-report.html)

* Bouncy Castle Crypto Package
  + a Java implementation of cryptographic algorithm
  + CVE-2013-1624
    - does not properly consider timing side-channel attacks on a noncompliant MAC check operation during the processing
  + **CVE-2015-6644**
  + **CVE-2015-7940**
  + CVE-2016-1000338
    - the DSA does not fully validate ASN.1 encoding of signature on verification.
  + CVE-2016-1000339
  + CVE-2016-1000341
  + CVE-2016-1000342
  + CVE-2016-1000343
    - the DSA key pair generator generates a weak private key if used with default values
  + CVE-2016-1000344
  + CVE-2016-1000345
    - vulnerable to padding oracle attack
  + CVE-2016-1000346
    - public key is not fully validated
  + CVE-2016-1000352
    - allowed the use of ECB mode. This mode is regarded as unsafe and support for it has been removed from the provider.
  + CVE-2017-13098
  + CVE-2018-5382
  + CVE-2020-0187
    - possible incorrect cryptographic algorithm chosen due to an incomplete comparison
  + CVE-2020-26939
    - attackers can obtain sensitive information about a private exponent because of Observable Differences in Behavior to Error Inputs
  + Update version to 1.60
* Spring Boot
  + CVE-2022-27772
    - vulnerable to temporary directory hijacking
  + CVE-2023-20883
    - there is potential for a denial-of-service
  + Update to the current version
* Logback Core
  + CVE-2021-42550
    - an attacker with the required privileges to edit configurations files could craft a malicious configuration allowing to execute arbitrary code loaded from LDAP servers
  + Update to current version
* Log4j
  + CVE-2020-9488
    - Improper validation of certificate with host mismatch
  + CVE-2021-44228
  + CVE-2021-44832
    - vulnerable to a remote code execution (RCE) attack
  + CVE-2021-45046
  + CVE-2021-45105
    - did not protect from uncontrolled recursion from self-referential lookups
  + Update to current version
* SnakeYAML
  + parser and emitter for Java
  + CVE-2017-18640
  + CVE-2021-4235
    - a maliciously crafted YAML file can cause the system to consume significant system resources
  + CVE-2022-1471
    - SnakeYaml's Constructor() class does not restrict types which can be instantiated during deserialization
  + CVE-2022-25857
  + CVE-2022-3064
  + CVE-2022-38749
  + CVE-2022-38750
  + CVE-2022-38751
  + CVE-2022-38752
  + CVE-2022-41854
    - vulnerable to Denial of Service attacks
  + CVE-2023-2251
    - Uncaught Exception in GitHub repository
  + Upgrade to the latest version
* Jackson Databind
  + General data-binding functionality for Jackson: works on core streaming API
  + CVE-2020-25649
  + CVE-2020-36518
  + CVE-2021-46877
  + CVE-2022-42003
  + CVE-2022-42004
  + CVE-2023-35116
  + Upgrade to the latest version
* Tomcat Embedded Core
  + Core Tomcat implementation
  + CVE-2019-17569
    - The result of the regression was that invalid Transfer-Encoding headers were incorrectly processed leading to a possibility of HTTP Request Smuggling
  + CVE-2020-11996
  + CVE-2020-13934
  + CVE-2020-13935
    - The payload length in a WebSocket frame was not correctly validated in Apache Tomcat
  + CVE-2020-13943
    - exceeded the agreed maximum number of concurrent streams for a connection
  + CVE-2020-17527
  + CVE-2020-1935
    - the HTTP header parsing code used an approach to end-of-line parsing that allowed some invalid HTTP headers to be parsed as valid
  + CVE-2020-1938
  + CVE-2020-8022
  + CVE-2020-9484
    - an attacker is able to control the contents and name of a file on the server
  + CVE-2021-24122
  + CVE-2021-25122
    - could duplicate request headers and a limited amount of request body from one request to another meaning user A and user B could both see the results of user A's request
  + CVE-2021-25329
  + CVE-2021-30640
    - A vulnerability in the JNDI Realm of Apache Tomcat allows an attacker to authenticate using variations of a valid user name and/or to bypass some of the protection provided by the LockOut Realm
  + CVE-2021-33037
    - did not correctly parse the HTTP transfer-encoding request header in some circumstances leading to the possibility to request smuggling when used with a reverse proxy
  + CVE-2021-41079
    - did not properly validate incoming TLS packets
  + CVE-2021-43980
  + CVE-2022-29885
    - the EncryptInterceptor incorrectly stated it enabled Tomcat clustering to run over an untrusted network
  + CVE-2022-34305
    - the Form authentication example in the examples web application displayed user provided data without filtering, exposing a XSS vulnerability.
  + CVE-2022-42252
    - Tomcat did not reject a request containing an invalid Content-Length header making a request smuggling attack possible
  + CVE-2023-28708
    - did not include the secure attribute. This could result in the user agent transmitting the session cookie over an insecure channel.
  + Upgrade to the latest version
* Hibernate Validator
  + CVE-2020-10693
    - A bug in the message interpolation processor enables invalid EL expressions to be evaluated as if they were valid. This flaw allows attackers to bypass input sanitation (escaping, stripping) controls that developers may have put in place when handling user-controlled data in error messages.
  + Upgrade to the latest version
* Spring web
  + CVE-2016-1000027
    - potential remote code execution
  + CVE-2020-5421
    - older unsupported versions, the protections against RFD attacks may be bypassed depending on the browser used
  + CVE-2021-22096
    - it is possible for a user to provide malicious input to cause the insertion of additional log entries
  + CVE-2021-22118
    - a WebFlux application is vulnerable to a privilege escalation
  + Upgrade to the latest version
* Spring Beans
  + CVE-2022-22965
    - application running on JDK 9+ may be vulnerable to remote code execution (RCE) via data binding
  + Upgrade to the latest version
* Spring WebMVC
  + CVE-2021-22060
    - In older unsupported versions, it is possible for a user to provide malicious input to cause the insertion of additional log entries
  + Upgrade to the latest version
* Spring Context
  + CVE-2022-22968
    - In older unsupported versions, the patterns for disallowedFields on a DataBinder are case sensitive which means a field is not effectively protected unless it is listed with both upper and lower case for the first character of the field, including upper and lower case for the first character of all nested fields within the property path.
  + Upgrade to the latest version
* Spring Expression
  + CVE-2022-22950
  + CVE-2023-20861
  + CVE-2023-20863
    - it is possible for a user to provide a specially crafted SpEL expression that may cause a denial-of-service (DoS) condition.
  + Upgrade to the latest version

## Mitigation Plan

To resolve the identified security vulnerabilities in Artemis Financial’s software, all systems need to be updated to their current versions to resolve any bugs and reduce the security risks from earlier versions. It would also be beneficial to implement authentication to protect access to customer’s information.

**Works Cited**

Newman, J., & Richie, A. (2022, August 8). *FTC strengthens security safeguards for consumer financial information following widespread data breaches*. Federal Trade Commission. <https://www.ftc.gov/news-events/news/press-releases/2021/10/ftc-strengthens-security-> safeguards-consumer-financial-information-following-widespread-data