GLOBALRAIN

Artemis Financial Vulnerability Assessment Report

Table of Contents

Document Revision History

Version	Date	Author	Comments
1.0	**11/7/2023**	**Tanner Meininger**	**Added information to 'Interpreting Client Needs', 'Areas of Security', and 'Manual Review' sections**
1.1	**11/8/2023**	**Tanner Meininger**	**Added information to 'Static Testing' and 'Mitigation Plan' sections**

Client

ARTEMIS
FINANCIAL

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
Tanner Meininger

1. Interpreting Client Needs

Value of Secure Communications:
Secure communications are of utmost importance to Artemis Financial. Given that they handle sensitive financial information, maintaining the confidentiality and integrity of data is crucial. Secure communications help protect client data from eavesdropping, tampering, or interception during transmission. This is vital for maintaining the trust of their clients and ensuring compliance with data protection regulations.

International Transactions:
Artemis Financial does engage in international transactions, as part of their financial planning services. This involves the transfer of financial data across borders. Therefore, the company should consider international data protection laws and regulations, such as GDPR in Europe, to ensure compliance in these transactions.

Governmental Restrictions:
There may be governmental restrictions on secure communications, especially when dealing with international data transfers. Artemis Financial should be aware of and adhere to any regulatory requirements or restrictions imposed by governments in the regions where they operate.

External Threats:
The following external threats are present now and could pose a risk in the immediate future:

- **Phishing Attacks**: Hackers might attempt to impersonate Artemis Financial to gain access to sensitive client information.
- **Data Interception**: Attackers may attempt to intercept data during transit, especially in international transactions.
- **Distributed Denial of Service (DDoS) Attacks**: These attacks could disrupt the availability of Artemis Financials' services, causing financial losses and reputation damage.
- **Malware and Ransomware**: These pose a risk of data breaches, financial loss, and system disruption.

Modernization Requirements:
To address the modernization needs of Artemis Financial, we should consider the following factors:

- **Open-Source Libraries**: The use of open-source libraries should be carefully managed and monitored. They can introduce security vulnerabilities if not kept up to date. Regular audits and patch management are necessary.
- **Evolving Web Application Technologies**: Keeping up with evolving web application technologies is essential to ensure that the software remains secure and performs efficiently. This includes using the latest security protocols, frameworks, and best practices.
- **Security Awareness Training**: Employees should receive security awareness training to recognize and report security threats like phishing attempts. This is vital to mitigate the human element of security risks.

- **Incident Response Plan**: Artemis Financial should establish an incident response plan to address security incidents promptly and effectively.

2. Areas of Security

In the context of Artemis Financials' software application, a few areas of the Vulnerability Assessment Process Flow are applicable:

- **Cryptography**: Given that financial data is sensitive, encryption plays a critical role in protecting data at rest and in transit. Evaluating encryption use and potential vulnerabilities is crucial for ensuring data confidentiality.
- **APIs**: Artemis Financial relies on APIs for its web-based application. Secure API interactions are essential to safeguard the data and communications between different components of the application.
- **Input Validation**: Secure input validation is relevant to ensure that data from users and external sources are properly validated, preventing common security vulnerabilities like SQL injection and cross-site scripting (XSS).
- **Code Error**: Secure code handling is relevant because it addresses how errors are handled within the code. Proper error handling can prevent information leakage and enhance the overall security of the application.

3. Manual Review

Vulnerabilities found through manual review:

- **GreetingController**: There seems to be a lack of input validation
- **CRUDController**: Business names are sent as request parameters
- **DocData**: The database connection parameters are written in a way that disallows for it to be reused
- No authentication found and overall lack of cryptography
- No HTTPS is used within the service

4. Static Testing

- **Name**: bcprov-jdk15on-1.46.jar
    - **Vulnerability IDs**:
        - CVE-2016-1000352
        - CVE-2016-1000346
        - CVE-2016-1000345
        - CVE-2016-1000344
        - CVE-2016-1000343
        - CVE-2016-1000342
        - CVE-2016-1000341
        - CVE-2016-1000339

        - CVE-2016-1000338
        - CVE-2018-5382
        - CVE-2017-13098
        - CVE-2013-1624
    - **Fix**: Upgrade to org.bouncycastle:bcprov-jdk15on 1.66+

- **Name**: hibernate-validator-6.0.18.Final.jar
    - **Vulnerability ID**:
        - CVE-2020-10693
    - **Fix**: Upgrade to org.springframework.boot:spring-boot-starter-web 2.2.7.RELEASE+

- **Name**: jackson-databind-2.10.2.jar
    - **Vulnerability IDs**:
        - CVE-2023-35116
        - CVE-2021-46877
        - CVE-2022-42004
        - CVE-2022-42003
        - CVE-2020-36518
        - CVE-2020-25649
    - **Fix**: Upgrade to org.springframework.boot:spring-boot-starter-web 2.6.15+

- **Name**: log4j-api-2.12.1.jar
    - **Vulnerability ID**:
        - CVE-2020-9488
    - **Fix**: Upgrade to Apache Log4j 2.12.3 and 2.13.1+

- **Name**: logback-core-1.2.3.jar
    - **Vulnerability ID**:
        - CVE-2021-42550
    - Fix: Upgrade to logback version 1.2.8+

- **Name**: snakeyaml-1.25.jar
    - **Vulnerability IDs**:
        - CVE-2022-1471
        - CVE-2017-18640
        - CVE-2022-25857
        - CVE-2022-38749
        - CVE-2022-38751
        - CVE-2022-38752
        - CVE-2022-41854
        - CVE-2022-38750
    - **Fix**: Upgrade to version 2.0 and beyond

- **Name**: spring-boot-2.2.4.RELEASE.jar and spring-boot-starter-web-2.2.4.RELEASE.jar
    - **Vulnerability IDs**:
        - CVE-2023-20873
        - CVE-2022-27772

    - CVE-2023-20883
  - **Fix**: Upgrade to 3.0.6+ or 2.7.11+

- **Name**: spring-core-5.2.3.RELEASE.jar, spring-web-5.2.3.RELEASE.jar, and spring-webmvc-5.2.3.RELEASE.jar
  - **Vulnerability IDs**:
    - CVE-2023-20863
    - CVE-2023-20861
    - CVE-2022-22971
    - CVE-2022-22970
    - CVE-2022-22968
    - CVE-2022-22965
    - CVE-2022-22950
    - CVE-2021-22060
    - CVE-2021-22096
    - CVE-2021-22118
    - CVE-2020-5421
    - CVE-2016-1000027
  - **Fix**: Upgrade to org.springframework.boot:spring-boot-starter-web 2.5.15+

- **Name**: tomcat-embed-core-9.0.30.jar and tomcat-embed-websocket-9.0.30.jar
  - **Vulnerability ID**:
    - CVE-2023-45648
    - CVE-2023-42795
    - CVE-2023-44487
    - CVE-2023-41080
    - CVE-2023-28708
    - CVE-2022-42252
    - CVE-2021-43980
    - CVE-2022-34305
    - CVE-2022-29885
    - CVE-2021-41079
    - CVE-2021-33037
    - CVE-2021-30640
    - CVE-2021-25329
    - CVE-2021-25122
    - CVE-2021-24122
    - CVE-2020-17527
    - CVE-2020-13943
    - CVE-2020-13935

        - CVE-2020-13934
        - CVE-2020-8022
    - **Fix**: Upgrade to org.springframework.boot:spring-boot-starter-web 2.7.17+

5. Mitigation Plan

- Add input validation within the GreetingController.
- Move request parameters within the CRUDController to either the header or the body, as compared to the URI.
- Change the database connection parameters within DocData so it can either be reused or modified.
- Put in place authentication and banking industry standard cryptography systems.
- Implement HTTPS communication protocol.
- Update all dependencies that are listed above to their most current versions.