GLOBALRAIN

Artemis Financial Vulnerability Assessment Report

Table of Contents

Document Revision History

Version	Date	Author	Comments
1.0	7/12/23	Brian Modzelewski	

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
Brian Modzelewski

1. Interpreting Client Needs

Secure communication describes the mechanism the business uses to send and receive data between the client and the server. Using a secure communication technique eliminates the possibility of data compromise. Since Artemis Financial manages customers' finances and develops financial plans for them, AF has access to sensitive data that outside attackers may search for. When it comes to the customer contact operations, the organization should place a strong focus on secure communications. SinceArtemis Financial is a financial institution, all of its transactions and communications must abide by all applicable laws and regulations. Any dangers, real or imagined, would come from someone attempting to obtain client financial information or client personal information. Attacks are a possibility if the API isn't sufficiently secure. Information may then leak as a result of this. All communication should be done over HTTPS since sensitive information will be passing back and forth between the firm and the client. Two-factor authentication should be enabled to assist prevent any false login attempts.

2. Areas of Security

In order to achieve a level of security to thwart any outside intervention and also to supply software to protect business and client privilege, secure coding is essential. Code mistakes are important because the organization has to manage problems securely. Since web services employ RESTful APIs, which require safe communication, APIs are important. Last but not least, input validation is important because when user input is received by the RESTful API, it needs to be cleared and verified.

3. Manual Review

When exchanging sensitive information, HTTPS is advised; the service does not utilize it. There is also no authentication mechanism in place for use in verification. Business names are given as request parameters within the CRUD Controller class; requests are not verified, leaving the system open to intrusion. This demonstrates its vulnerability by making information accessible to others.

4. Static Testing

bcprov-jdk15on-1.46.jar – several vulnerabilities on 1.46, update to latest version.
CVE-2013-1624
CVE-2015-6644
CVE-2015-7940
CVE-2016-1000338
CVE-2016-1000339
CVE-2016-1000341
CVE-2016-1000342
CVE-2016-1000343
CVE-2016-1000344
CVE-2016-1000345
CVE-2016-1000346
CVE-2016-1000352

CVE-2017-13098
CVE-2018-1000613
CVE-2018-5382
Log4j-api-2.12.1.jar – one vulnerability, update to latest version.
CVE-2020-9488
Snakeyalm-1.25.jar – one vulnerability, update to latest version.
CVE-2017-18640
Jackson-databind-2.10.2.jar – one vulnerability, update to latest version.
CVE-2020-25649
Tomcat-embed-core-9.0.30.jar – several vulnerabilities, update to the latest tomcat version.
CVE-2019-17569
CVE-2020-11996
CVE-2020-13934
CVE-2020-13935
CVE-2020-13943
CVE-2020-17527
CVE-2020-1935
CVE-2020-1938
CVE-2020-8022
CVE-2020-9484
CVE-2021-24122
Hibernate-validator-6.0.18.Final.jar – one vulnerability, update to latest version.
CVE-202-10693Spring-core-5.2.3.RELEASE.jar – one vulnerability, update to latest version.
CVE-2020-5421

5. Mitigation Plan

It is advised that we first establish the security of corporate and customer information before addressing any current or potential problems. We must migrate to HTTPS for all communication in order to ensure the security of the data, which will keep prying eyes at bay. Second, request parameters would be sent to headers. Third, we would eliminate any mention of company names from database credentials that were hard-coded. In order to secure user information, we wish to build and activate two-factor authentication solutions. Finally, we will update each dependent found during the last dependency check.