****

# 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** | **1/18/2023** | **Michael Barbuzano** |  |

## 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

Michael Barbuzano

## Interpreting Client Needs

1. Secure communications are essential for a company such as Artemis Financial, because they deal with very sensitive client information, such as their financial information.
2. Because Artemis Financial deals with a global client base, they most certainly make international transactions.
3. The Federal Trade Commission Act requires businesses to provide reasonable security for sensitive information. This would apply to a company like Artemis Financial, as the financial information of their client’s is considered sensitive information.
4. Possible external threats to the company would be anyone looking to steal the financial information of their clients, for malicious purposes.
5. Although open-source libraries are a very useful time-saving tool for programmers. Open-source libraries have common vulnerabilities, which could leave a program exposed to hackers who know how to exploit these vulnerabilities. Also, open-source libraries containing malware can completely compromise a web application.

## Areas of Security

Because this web application is utilizing REST APIs, secure API interactions are going to be a crucial area of security. API security can prevent attacks such as cross-site scripting, SQL injections, and protect sensitive data from being compromised.

The use of cryptography seems essential for any web application that deals with sensitive information, such as the Artemis Financials’ client financial information. Cryptography is the process of protecting sensitive information from unauthorized users. Assuring secure communication and encrypting data to protect company and client information would certainly be essential for such a web application.

## Manual Review

I do not have any experience working with APIs, so after searching for a few hours to find any API implementation. I finally found a folder called “API.class” inside “org.apiguardian.api”. I am not sure what exactly secure API implementation looks like, but I did not notice any form of encryption, so I’m assuming more security measures could be taken to assure secure API interactions.

After doing some research on implementing cryptography in Java, I found that Java has a cryptography API provided with the Java cryptography extension, JCE. I did not notice this in the program and think this would be a great addition in working towards protecting client data.

## Static Testing

After running the Maven dependency check, 66 vulnerabilities we found and 13 vulnerable dependencies. The first of which was dependency bcprov-jdk15on-1.46.jar, with vulnerabilities CVE-2016-1000352, CVE-2016-1000346, CVE-2016-1000344, CVE-2016-1000344, CVE-2016-1000342, CVE-2016-1000341, CVE-2016-1000339, CVE-2018-5382, CVE-2017-13098, and CVE-2013-1624. Most of these vulnerabilities refer to Bouncy Castle JCE version 1.55 and earlier being very vulnerable to attacks. It seems like these early versions of the software are vulnerable due to weak encryption.

The next dependency was spring-boot-2.2.4.RELEASE.jar, with vulnerability CVE-2022-27772. This vulnerability was first published March 30th, 2022, so it is recent. This vulnerability states that spring versions prior to “v2.2.11.RELEASE” are vulnerable to temporary directory hijacking.

Then next dependency was spring-boot-2.2.4.RELEASE.jar with vulnerability CVE-2021-42550. This vulnerability was first published on December 16th, 2021. It states that logback version 1.2.7 and prior versions allow attackers with required privileges to edit configurations files could execute code loaded from the LDAP servers.

The next dependency was log4j-api-2.12.1.jar, with critical vulnerability CVE-2021-44228. This vulnerability was first published on December 10th, 2021 and states that versions of Apache Log4j prior to 2.15.0 do not protect against attacker controlled LDAP and other JNDI related endpoints, allowing an attacker to execute code loaded from LDAP servers.

The next dependency detected was snakeyaml-1.25.jar, with vulnerability code CVE-2022-25857. This vulnerability was first published August 30th, 2022 and states that package ord.yaml:snakeyaml versions prior to version 1.3.1 are vulnerable to Denial of Service (DOS) attacks, due to missing nested depth limitation for collection.

The next dependency was jackson-databind-2.10.2.jar, with vulnerability CVE-2020-25649. This vulnerability was first published on December 03, 2020 and it states that a flaw was found in FasterXML Jackson Databind, where it does not have entity expansion secured properly. This causes a vulnerability to XML external entity attacks, which could compromise data integrity.

The next dependency found was tomcat-embed-core-9.0.30.jar, with 20 vulnerabilities. One of which being CVE-2020-1938, a critical vulnerability. This vulnerability was first published on February 24th, 2020, and it states that when using Apache JServ Protocol, care must be taken when trusting connections to Apache Tomcat. This is because Tomcat treats AJP connections as having higher trust than similar HTTP connections. This could be exploited by hackers in numerous ways.

The next dependency detected was hibernate-validator-6.0.18.Final.jar, with vulnerability hibernate-validator-6.0.18.Final.jar. This vulnerability was first published May 6th, 2020, and it states that a flaw was found in Hibernate Validator version 6.1.2Final, that enables invalid EL expressions to be evaluated as if they were valid.

Although there were a few more dependencies found, the rest did not have any published vulnerabilities in the National Vulnerability Database.

## Mitigation Plan

To deal with the vulnerabilities caused by the bcprov-jdk15on-1.46.jar, it seems like updating to any version after 1.55 would solve these vulnerabilities, as versions before 1.55 are considered unsafe and are not supported.

Looking back at these vulnerabilities it appears that using an outdated package can lead to many vulnerabilities in a program, as many of the vulnerabilities found in the dependency check seem to be caused by an outdated version of the software. Moving forward to reduce the number of vulnerabilities in a program, it would be best to do extensive research on each package, before implementing them into the project, as well as keeping up to date as new vulnerabilities are constantly being discovered.

Other than running updates and doing research on any newly published vulnerabilities, I also think it would be important to make sure that there is secure API implementation is secure. As unsecure APIs can allow hackers to exploit their vulnerabilities to compromise sensitive data. Protecting data should be the highest priority at a company like Artemis Financial.

Another step towards protecting sensitive client data would be to implement cryptography, such as the Java cryptography API provided by the Java Cryptography extension. The implementation of cryptography ensures that the correct information is going to the correct client and has not been tampered with, essential technology for a company like Artemis Financial.

Works Cited

NortonOnline. (n.d.). *What are some of the laws regarding internet and data security?* What Are Some of the Laws Regarding Internet and Data Security? Retrieved January 21, 2023, from <https://us.norton.com/blog/privacy/laws-regarding-internet-data-security>

Staff, the P. N. O., & Staff, D. P. I. P. and C. T. O. (2022, April 26). *Protecting personal information: A guide for business*. Federal Trade Commission. Retrieved January 21, 2023, from <https://www.ftc.gov/business-guidance/resources/protecting-personal-information-guide-business>

*The dangers of open source risk*. Veracode. (n.d.). Retrieved January 21, 2023, from <https://www.veracode.com/security/dangers-open-source-risk#:~:text=Open%20source%20libraries%20containing%20malware,frequently%20used%20in%20ransomware%20attacks>.

Help Net Security. (2020, May 19). *How secure are open source libraries?* Help Net Security. Retrieved January 21, 2023, from <https://www.helpnetsecurity.com/2020/05/21/secure-open-source-libraries/>

Contributor, T. T. (2019, August 16). *What is API security? definition from whatis.com.* App Architecture. Retrieved January 21, 2023, from <https://www.techtarget.com/searchapparchitecture/definition/API-security#:~:text=Implementing%20API%20security%20is%20important,and%20the%20programs%20they%20support>.

RSI Security. (2022, December 23). *What is cryptography in cyber security: Types, examples & more*. RSI Security. Retrieved January 21, 2023, from <https://blog.rsisecurity.com/what-is-cryptography-in-cyber-security/#:~:text=Cryptography%20in%20computer%20network%20security,it%20unreadable%20without%20a%20key>.

Morgan, N. (n.d.). *The role of cryptography in information security*. Triskele Labs. Retrieved January 22, 2023, from https://www.triskelelabs.com/blog/the-role-of-cryptography-in-information-security#:~:text=Cryptography%20ensures%20the%20integrity%20of,been%20tampered%20with%20during%20transmission.

Team, C. (2021, November 3). *7 best programming languages for cryptography*. Codecademy Blog. Retrieved January 22, 2023, from https://www.codecademy.com/resources/blog/programming-languages-for-cryptography/

Bertram, A. (2021, March 29). *5 bad practices that lead to insecure apis in cloud computing: TechTarget*. Cloud Computing. Retrieved January 22, 2023, from https://www.techtarget.com/searchcloudcomputing/tip/5-bad-practices-that-lead-to-insecure-APIs-in-cloud-computing