GLOBALRAIN

Artemis Financial Vulnerability Assessment Report

Table of Contents

Document Revision History

Version	Date	Author	Comments
1.0	3/24/2024	Richard Seabridge	

Client

Developer
Richard Seabridge

1. Interpreting Client Needs

Firstly, I'll shall address the obvious because this is a financial company security is of the utmost importance due to the nature of the business. As stated, "Security is everyone's business". All financial institutions deal with security in their own way, but they have mush to protect from outside threats. The institutions have a lot of information on their individual clients, government agencies, and everyone around the world. So, the value of security is at the top for Artemis Financial. When a government agency is being dealt with in any manner then it is safe to say that most if not all transactions will be encrypted in some way. One major restriction is in sharing classified information or trade secrets from other nations agencies.

Most records or information are kept on computer now a days, so the threats are always present or could come about soon for example cyber-attacks, data breaches, or exploitations of vulnerabilities. They could be hacked remotely from someone attaching a keylogging application to the server or if someone were able to use the vulnerability within the dependencies of the code. Open-source libraries are just that "Open". It is a very useful tool for us developers but if we are not careful then those libraries could introduce vulnerabilities if they are not properly used or maintained as intended. Web applications are always evolving as is everything else within the tech field. New technologies are always improving on the old features but can present issues later if they are not kept up to date or implemented in the way was intended.

2. Areas of Security

- APIs – as a financial institution deals in all manner of information, it is important that these interactions are secure.
- Cryptography – Not only will Artemis being dealing with individuals, but they will also have government agencies. This part is very important as they will have to use encryption to keep the information as secure as possible.

- Code Quality – The first step in security should be with the developers. Using standard practices with creating the code will also help with preventing unwanted access.
- Code Error – Implementing the proper code will ensure the program or services work as intended and secure user input errors.

3. Manual Review

- Outdated Maven Plugin – Current version was 5.3.0 so this needs to be updated to the newest version which is 9.0.10. This will cause faulty, or no dependencies checks which will cause inaccurate test results.
- Outdated third-Party libraries – Open-source libraries must be maintained and updated to ensure that no vulnerabilities can be exploited.

4. Static Testing

- Bouncy-castle crypto package: This would allow and attacker to inject extra elements in the sequence making up the signature and still have it validate.
- Hibernate validator: This would allow attackers to bypass input sanitation controls the developers may be put in place.
- Jackson-databind: SMTPS connections could be intercepted by a "man in the middle" attack
- Snakeyaml: remote code execution by deserializing yaml content.
- Sprinboot: unsupported versions and cloud foundry could lead to a security bypass.
- Springweb: Third party vulnerability and must be authenticated
- Tomcat-embedded-core/websocket: trigger high CPU usage which could render an unresponsive server.

5. Mitigation Plan

After reviewing the code manually and performing the static test it would be essential to simply update dependencies to their newest or current versions.