

**Artemis Financial Vulnerability Assessment Report**

**Brennan Haggett**

**11/06/2023**

Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comments** |
| **1.0** | **11/06/2023** | **Brennan Haggett** |  |

Client



Developer

Brennan Haggett

* Interpreting Client Needs
* **What is the value of secure communications to the company?**

The value of secure communications for Artemis Financial is a fundamental part of its business. Artemis will need to build and maintain customer trust, safeguard customer financial data, and maintain compliance with regulations. If the company is able to ensure these things are upheld, it will be able to maintain legitimacy and credibility within the customer base and throughout the market. As Artemis Financial deploys a RESTful web API, the company's modernization standards will need to be robust, and security measures will need to be secure to prevent and mitigate threats. As Artemis Financial ensures secure communications, the company will gain a competitive edge, creating long-term client relationships.

* **Does the company make any international transactions?**

As Artemis Financial is a web-based company, it will be operating across multiple different countries and transacting from different regions of the world. Artemis Financial should prioritize secure communications to comply with various governmental regulations and safeguard sensitive financial data during cross-border exchanges. Employing encryption protocols and secure communication channels will be needed to protect against external threats and ensure the confidentiality and integrity of financial information across international transactions.

* **Are there governmental restrictions about secure communications to consider?**

Governmental restrictions on secure communications vary across jurisdictions. It's vital for Artemis Financial to comply with the laws and regulations governing secure communications. Diffrent countries may have specific encryption requirements or restrictions on the use of certain cryptographic algorithms. The compnay can also need to fallow industry-specific standards and regulations for handling payment card data. Artemis Financial should conduct a thorough review of relevant laws and standards in the regions where it operates to ensure compliance to all laws.

* **What external threats might be present now and in the immediate future?**

External threats for Artemis Financial's web application include cybersecurity risks such as data breaches, phishing attacks, ransomware, and exploitation of software vulnerabilities. This can also include threats related to evolving technologies, such as artificial intelligence-driven The company can face social engineering attacks targeting users or employees, which can lead to data breaches.

**What are the modernization requirements that you must consider?**

The modernization requirements for Artemis Financial should be to ensure that they are up-to-date on all technological advancements and evolving security measures. This includes updating software and infrastructure to ensure the latest technologies for enhanced efficiency and customer experience. The company should adopt cybersecurity practices and tools to protect against evolving threats. The company should be following trends such as cloud adoption, mobile banking, and advanced analytics, which can help the company stay competitive in the financial sector

* Areas of Security

Artemis Financial's web application will need to have a comprehensive security strategy from all perspectives and across various areas. Firstly, it is important that we implement secure coding practices so that we create a safe and secure software foundation, ensuring security against vulnerabilities and unauthorized access from outside users. The application's reliance on RESTful APIs shows the importance of secure API interactions, safeguarding data exchanges from one client to another, and preventing potential compromises in the system. Encryption within the code and communication lines is crucial to maintaining integrity, creating high confidence in the company's system, and ensuring that sensitive information transmitted over the internet is secure. Cryptography plays a major role in the security of the code and customer data. If we are able to employ encryption techniques for data, we can prevent unauthorized decoding and enhance the overall data protection process. The client/server step in the vulnerability assessment process flow shows that communication is a vital part of web applications, and Artemis Financial's software should use these security measures to do so, which will include physical security and change control to safeguard hardware, networks, and data. Input validation is vital to prevent injection attacks and ensure the integrity of user inputs and their data. By going through each of these security steps one by one Artemis Financial can establish a robust defense against potential vulnerabilities, unauthorized access, and data breaches.

Manual Review

**Inadequate Encapsulation in customer.java:**

**Vulnerability-** Lack of proper encapsulation for account\_balance.

**Recommendation**: Make account\_balance private and provide correct getter and setter methods for controlled access.

**Input Validation and XSS Concern in GreetingController.java:**

**Vulnerability:** Lack of input validation for the name parameter, this can potentially lead to injection attacks. Constructing a response directly from user input may introduce XSS vulnerabilities.

**Recommendation:** Implement input validation for user inputs and sanitize output to prevent injection and XSS attacks.

**Database Connection Handling in DocData.java:**

**Vulnerability:** Opening a database connection without proper exception handling or resource management (closing the connection), which can lead to resource leaks.

**Recommendation**: Implement proper exception handling and ensure the database connection is closed after use to prevent resource leaks.

* Static Testing



* **Document Results**

Vulnerable Dependencies: 13

Vulnerabilities Found: 124

**Dependency**

**1- bcprov-jdk15on-1.46.jar**

CVE count - 18

Severity - HIGH

Evidence Count - 38

[**https://nvd.nist.gov/vuln/search/results?form\_type=Advanced&results\_type=overview&search\_type=all&cpe\_vendor=cpe%3A%2F%3Abouncycastle&cpe\_product=cpe%3A%2F%3Abouncycastle%3Alegion-of-the-bouncy-castle-java-crytography-api&cpe\_version=cpe%3A%2F%3Abouncycastle%3Alegion-of-the-bouncy-castle-java-crytography-api%3A1.46**](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Abouncycastle&cpe_product=cpe%3A%2F%3Abouncycastle%3Alegion-of-the-bouncy-castle-java-crytography-api&cpe_version=cpe%3A%2F%3Abouncycastle%3Alegion-of-the-bouncy-castle-java-crytography-api%3A1.46)

In the Bouncy Castle JCE Provider version 1.55 and earlier the ECIES implementation allowed the use of ECB mode. This mode is regarded as unsafe and support for it has been removed from the provider.

**2- hibernate-validator-6.0.18.Final.jar**

CVE count - 1

Severity - MEDIUM

Evidence Count - 32

<https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aredhat&cpe_product=cpe%3A%2F%3Aredhat%3Ahibernate_validator&cpe_version=cpe%3A%2F%3Aredhat%3Ahibernate_validator%3A6.0.18>

A flaw was found in Hibernate Validator version 6.1.2.Final. 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.

**3- jackson-databind-2.10.2.jar**

CVE count - 1

Severity - MEDIUM

Evidence Count - 32

<https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Afasterxml&cpe_product=cpe%3A%2F%3Afasterxml%3Ajackson-databind&cpe_version=cpe%3A%2F%3Afasterxml%3Ajackson-databind%3A2.10.2>

jackson-databind through 2.15.2 allows attackers to cause a denial of service or other unspecified impact via a crafted object that uses cyclic dependencies. NOTE: the vendor's perspective is that this is not a valid vulnerability report, because the steps of constructing a cyclic data structure and trying to serialize it cannot be achieved by an external attacker.

**4- log4j-api-2.12.1.jar**

CVE count - 6

Severity - HIGH

Evidence Count - 39

<https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aapache&cpe_product=cpe%3A%2F%3Aapache%3Alog4j&cpe_version=cpe%3A%2F%3Aapache%3Alog4j%3A2.12.1>

Apache Log4j2 versions 2.0-beta7 through 2.17.0 (excluding security fix releases 2.3.2 and 2.12.4) are vulnerable to a remote code execution (RCE) attack when a configuration uses a JDBC Appender with a JNDI LDAP data source URI when an attacker has control of the target LDAP server. This issue is fixed by limiting JNDI data source names to the java protocol in Log4j2 versions 2.17.1, 2.12.4, and 2.3.2.

**5- logback-core-1.2.3.jar**

CVE count - 1

Severity - LOW

Evidence Count - 42

<https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aqos&cpe_product=cpe%3A%2F%3Aqos%3Alogback&cpe_version=cpe%3A%2F%3Aqos%3Alogback%3A1.2.3>

In logback version 1.2.7 and prior versions, an attacker with the required privileges to edit configurations files could craft a malicious configuration allowing to execute arbitrary code loaded from LDAP servers.

**6- snakeyaml-1.25.jar**

CVE count - 1

Severity - MEDIUM

Evidence Count - 31

<https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Avmware&cpe_product=cpe%3A%2F%3Avmware%3Aspring_boot&cpe_version=cpe%3A%2F%3Avmware%3Aspring_boot%3A2.2.4>

In Spring Boot versions 3.0.0 - 3.0.6, 2.7.0 - 2.7.11, 2.6.0 - 2.6.14, 2.5.0 - 2.5.14 and older unsupported versions, there is potential for a denial-of-service (DoS) attack if Spring MVC is used together with a reverse proxy cache.

**7- spring-boot-2.2.4.RELEASE.jar**

CVE count - 8

Severity - CRITICAL

Evidence Count - 44

<https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Avmware&cpe_product=cpe%3A%2F%3Avmware%3Aspring_boot&cpe_version=cpe%3A%2F%3Avmware%3Aspring_boot%3A2.2.4>

In Spring Boot versions 3.0.0 - 3.0.5, 2.7.0 - 2.7.10, and older unsupported versions, an application that is deployed to Cloud Foundry could be susceptible to a security bypass. Users of affected versions should apply the following mitigation: 3.0.x users should upgrade to 3.0.6+. 2.7.x users should upgrade to 2.7.11+. Users of older, unsupported versions should upgrade to 3.0.6+ or 2.7.11+.

**8- spring-boot-starter-web-2.2.4.RELEASE.jar**

CVE count - 3

Severity - CRITICAL

Evidence Count - 39

<https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Avmware&cpe_product=cpe%3A%2F%3Avmware%3Aspring_framework&cpe_version=cpe%3A%2F%3Avmware%3Aspring_framework%3A5.2.3>

A Spring MVC or Spring WebFlux application running on JDK 9+ may be vulnerable to remote code execution (RCE) via data binding. The specific exploit requires the application to run on Tomcat as a WAR deployment. If the application is deployed as a Spring Boot executable jar, i.e. the default, it is not vulnerable to the exploit. However, the nature of the vulnerability is more general, and there may be other ways to exploit it.

**9- spring-core-5.2.3.RELEASE.jar**

CVE count - 3

Severity - CRITICAL

Evidence Count - 35

<https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Avmware&cpe_product=cpe%3A%2F%3Avmware%3Aspring_framework&cpe_version=cpe%3A%2F%3Avmware%3Aspring_framework%3A5.2.3>

A Spring MVC or Spring WebFlux application running on JDK 9+ may be vulnerable to remote code execution (RCE) via data binding. The specific exploit requires the application to run on Tomcat as a WAR deployment. If the application is deployed as a Spring Boot executable jar, i.e. the default, it is not vulnerable to the exploit. However, the nature of the vulnerability is more general, and there may be other ways to exploit it.

**10- spring-web-5.2.3.RELEASE.jar**

CVE count - 11

Severity - CRITICAL

Evidence Count - 34

<https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Avmware&cpe_product=cpe%3A%2F%3Avmware%3Aspring_framework&cpe_version=cpe%3A%2F%3Avmware%3Aspring_framework%3A5.2.3>

A Spring MVC or Spring WebFlux application running on JDK 9+ may be vulnerable to remote code execution (RCE) via data binding. The specific exploit requires the application to run on Tomcat as a WAR deployment. If the application is deployed as a Spring Boot executable jar, i.e. the default, it is not vulnerable to the exploit. However, the nature of the vulnerability is more general, and there may be other ways to exploit it.

**11- spring-webmvc-5.2.3.RELEASE.jar**

CVE count - 11

Severity - CRITICAL

Evidence Count - 36

<https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Avmware&cpe_product=cpe%3A%2F%3Avmware%3Aspring_framework&cpe_version=cpe%3A%2F%3Avmware%3Aspring_framework%3A5.2.3>

A Spring MVC or Spring WebFlux application running on JDK 9+ may be vulnerable to remote code execution (RCE) via data binding. The specific exploit requires the application to run on Tomcat as a WAR deployment. If the application is deployed as a Spring Boot executable jar, i.e. the default, it is not vulnerable to the exploit. However, the nature of the vulnerability is more general, and there may be other ways to exploit it.

**12- tomcat-embed-core-9.0.30.jar**

CVE count - 24

Severity - CRITICAL

Evidence Count - 30

<https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aapache&cpe_product=cpe%3A%2F%3Aapache%3Atomcat&cpe_version=cpe%3A%2F%3Aapache%3Atomcat%3A9.0.30>

Improper Input Validation vulnerability in Apache Tomcat.Tomcat from 11.0.0-M1 through 11.0.0-M11, from 10.1.0-M1 through 10.1.13, from 9.0.0-M1 through 9.0.81 and from 8.5.0 through 8.5.93 did not correctly parse HTTP trailer headers. A specially crafted, invalid trailer header could cause Tomcat to treat a single request as multiple requests leading to the possibility of request smuggling when behind a reverse proxy. Users are recommended to upgrade to version 11.0.0-M12 onwards, 10.1.14 onwards, 9.0.81 onwards or 8.5.94 onwards, which fix the issue.

**13- tomcat-embed-websocket-9.0.30.jar**

CVE count - 25

Severity - CRITICAL

Evidence Count - 30

<https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aapache&cpe_product=cpe%3A%2F%3Aapache%3Atomcat&cpe_version=cpe%3A%2F%3Aapache%3Atomcat%3A9.0.30>

Improper Input Validation vulnerability in Apache Tomcat.Tomcat from 11.0.0-M1 through 11.0.0-M11, from 10.1.0-M1 through 10.1.13, from 9.0.0-M1 through 9.0.81 and from 8.5.0 through 8.5.93 did not correctly parse HTTP trailer headers. A specially crafted, invalid trailer header could cause Tomcat to treat a single request as multiple requests leading to the possibility of request smuggling when behind a reverse proxy. Users are recommended to upgrade to version 11.0.0-M12 onwards, 10.1.14 onwards, 9.0.81 onwards or 8.5.94 onwards, which fix the issue.

* Mitigation Plan

To address the security vulnerabilities in Artemis Financial's software, we will need to create a comprehensive mitigation plan that will be able to understand each vulnerability and come up with an action plan. These actions will be to make sure that the authentication mechanisms are secure and secure, implement safe and secure input validation, and enhance data encryption for both storage and communication. Then we will need to secure the API and update the software version so that we are always up-to-date on the latest software. These actions will be tailored to specific vulnerabilities that have been identified and will provide a strategic plan to fortify the software's resilience against external threats.

To ensure that Artemis Financial's web-based software is safe against the identified vulnerabilities, we will start to upgrade critical dependencies like Bouncy Castle, Hibernate Validator, and Jackson Databind to their latest versions, ensuring the implementation of secure coding practices and logging libraries such as Log4j and Logback will be updated, and configurations will be refined to reduce these exposures. Vital components such as Spring Boot, Spring Core, Spring Web, and Tomcat will undergo version updates, accompanied by comprehensive security audits. This approach aims to enhance the software's resilience, minimize attacks, and uphold security practices in alignment with industry standards.