GLOBALRAIN

Artemis Financial Vulnerability Assessment Report

Table of Contents

Document Revision History

Version	Date	Author	Comments
1.0	11/15/2023	Bryce Jensen	

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

Bryce Jensen

Interpreting Client Needs

What is the value of secure communications to the company?

Secure communications are of the utmost importance due to the sensitive financial information involved in their customers' individual financial plans. To ensure client's trust, it is important to maintain confidentiality of their data. Secure communications help with this by ensuring that the data is protected during transit. This reduces the risk of security risks and tampering with the data.

Does the company make any international transactions?

Although not explicitly expressed in the rubric, it is safe to assume that Artemis Financial makes international transactions. This means that additional protections are needed to ensure client's data is safe.

Are there governmental restrictions about secure communications to consider?

Absolutely. In fact, according to UNCTAD (United Nations Conference on Trade and Development), 71% of all countries have legislation regarding data protection and privacy. While only 15% have no legislation on the topic at all (UNCTAD, 2021). In Artemis Financials' case, because they participate in international restrictions, adherence to governmental restrictions would be very important.

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

Just like any company with that uses the internet for data transmission, potential threats may include:

Data interception: While in transmission, attackers may attempt to intercept data. This could include sensitive financial data. While it should be encrypted, nothing is 100% secure, especially after interception.

Brute force attacks: Unauthorized attempts to access the clients' accounts could give access to financials. This can also tie into brute force attacking the – above mentioned – encryption on the intercepted data.

What are the modernization requirements that you must consider?

Artemis Financial would like to modernize their operations. Some of the modernization requirements might include:

Use of open-source libraries: Assessing the security of open-source used in Artemis Financials' software is necessary to ensure there are no vulnerabilities in the library that would compromise data security. Updating to address these vulnerabilities is a must.

The adoption of evolving web application technologies: Web application technologies are constantly evolving. This means that Artemis Financial needs to stay on top of the latest security practices and include them in their own software to keep new threats from breaking through their security. To go along with this, ensuring that best security practices are being practiced, holding back emerging threats.

Cloud services: As Artemis Financial grows, they may choose to expand their business. It would be recommended that they implement cloud services now. This will make this potential transition easier when that choice is made. Cloud services will allow their data to be accessed from other locations.

UX Improvements: While not as important as some improvements, implementing a better user experience will help ensure that their clients have better experiences working with Artemis Financials' tools. Making this work across devices ensures that the experience is future proof as less people use desktop web browsers to access their information. Also, making this user-friendly and modern look helps the users have a better experience as well.

Areas of Security

Using the Vulnerability Assessment Process Flow Diagram (VAPFD), the areas of security that best relate to this software are:

Input Validation: Input validation is a great place to start when it comes to software security. All input should be properly verified and validated. This means that any input should be sanitized to prevent injection attacks like SQL Injection and XSS (Cross site scripting).

APIs: This can also benefit from Input Validation. Artemis Financial already has a RESTful API, it just needs to implement measures like auth tokens and rate limiting to help prevent unauthorized access.

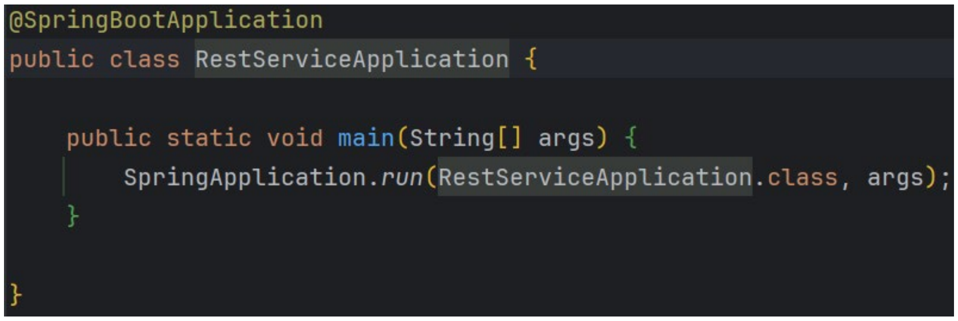
Cryptography: The sensitive nature of Artemis Financials' data requires things like encryption and other methods of cryptography. Specifically, E2E (end-to-end) encryption will ensure that the data being transferred will be more protected if intercepted.

Code Quality: Specifically, when it comes to secure coding practices, code quality is important for protection against security vulnerabilities.

Manual Review

Following along with the VAPFD, it starts with *Input Validation:*

The `RestServiceApplication.java` file (shown below) is the main driver of the code. It accepts a String of `args` as a parameter, which is fine, so long as those arguments are validated and sterilized of unwelcome or unapproved input.

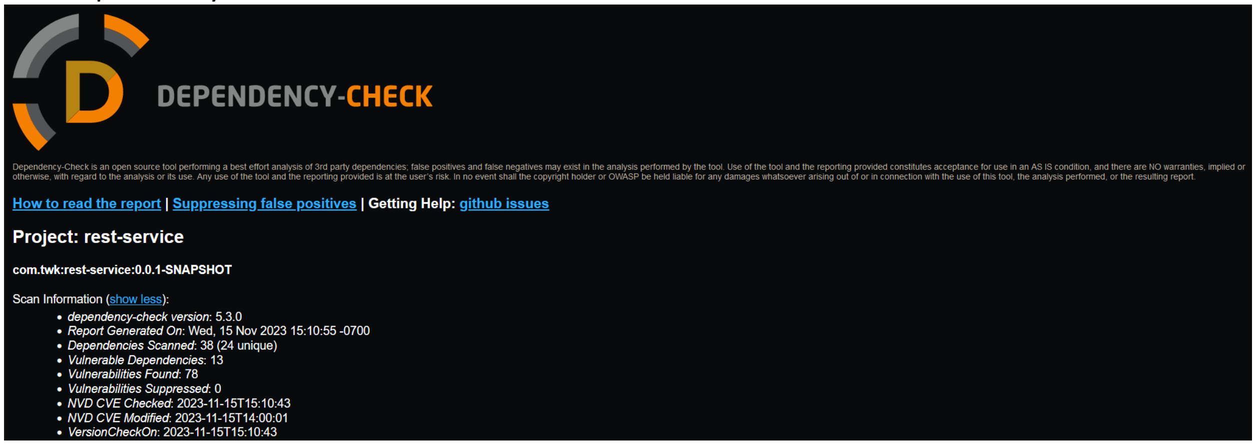


```
@SpringBootApplication
public class RestServiceApplication {

    public static void main(String[] args) {
        SpringApplication.run(RestServiceApplication.class, args);
    }

}
```

Another thing I noticed in the DocData.java file was that the credentials are hardcoded into the code. This is done for convenience to simplify workflow. After the initial testing of the software, this should be taken care of and not make it into production.

Static Testing

The Dependency Check screenshot is included below:



Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependencies; false positives and false negatives may exist in the analysis performed by the tool. Use of the tool and the reporting provided constitutes acceptance for use in an AS IS condition, and there are NO warranties, implied or otherwise, with regard to the analysis or its use. Any use of the tool and the reporting provided is at the user's risk. In no event shall the copyright holder or OWASP be held liable for any damages whatsoever arising out of or in connection with the use of this tool, the analysis performed, or the resulting report.

How to read the report | Suppressing false positives | **Getting Help:** github issues

Project: rest-service

com.twk:rest-service:0.0.1-SNAPSHOT

Scan Information (show less):

- *dependency-check version*: 5.3.0
- *Report Generated On*: Wed, 15 Nov 2023 15:10:55 -0700
- *Dependencies Scanned*: 38 (24 unique)
- *Vulnerable Dependencies*: 13
- *Vulnerabilities Found*: 78
- *Vulnerabilities Suppressed*: 0
- *NVD CVE Checked*: 2023-11-15T15:10:43
- *NVD CVE Modified*: 2023-11-15T14:00:01
- *VersionCheckOn*: 2023-11-15T15:10:43

A detailed vulnerability and solution list is provided below:

Dependency	Vulnerability	Description	Solution
bcprov-jdk15on-1.46.jar	**CVE-2013-1624**	The TLS implementation in the Bouncy Castle Java library before 1.48 and C# library before 1.8 does not properly consider timing side-channel attacks on a noncompliant MAC check operation during the processing of malformed CBC padding, which allows remote attackers to conduct distinguishing attacks and plaintext-recovery attacks via statistical analysis of timing data for crafted packets, a related issue to CVE-2013-0169.	?
spring-boot-2.2.4.RELEASE.jar	**CVE-2023-20873**	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+.	Upgrade to the required version as listed in the description.
logback-core-1.2.3.jar	**CVE-2021-42550**	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.	Upgrade to version: 1.3.0-alpha11, 1.2.9
log4j-api-2.12.1.jar	**CVE-2020-9488**	Improper validation of certificate with host mismatch in Apache Log4j SMTP appender. This could allow an SMTPS connection to be intercepted by a man-in-the-middle attack which could leak any log messages sent through that appender. Fixed in Apache Log4j 2.12.3 and 2.13.1	Upgrade to version: 2.12.3 or 2.13.1
snakeyaml-1.25.jar	**CVE-2017-18640**	The Alias feature in SnakeYAML before 1.26 allows entity expansion during a load operation, a related issue to CVE-2003-1564.	Don't have permissions to view the fix, I assume it is "update to the latest version"

jackson-databind-2.10.2.jar	**CVE-2022-42003**	In FasterXML jackson-databind before 2.14.0-rc1, resource exhaustion can occur because of a lack of a check in primitive value deserializers to avoid deep wrapper array nesting, when the UNWRAP_SINGLE_VALUE_ARRAYS feature is enabled. Additional fix version in 2.13.4.1 and 2.12.17.1	Upgrade to version: 2.13.4.1 or 2.12.17.1
tomcat-embed-core-9.0.30.jar	**CVE-2020-1938**	When using the Apache JServ Protocol (AJP), care must be taken when trusting incoming connections to Apache Tomcat. Tomcat treats AJP connections as having higher trust than, for example, a similar HTTP connection. If such connections are available to an attacker, they can be exploited in ways that may be surprising. In Apache Tomcat 9.0.0.M1 to 9.0.0.30, 8.5.0 to 8.5.50 and 7.0.0 to 7.0.99, Tomcat shipped with an AJP Connector enabled by default that listened on all configured IP addresses. It was expected (and recommended in the security guide) that this Connector would be disabled if not required. This vulnerability report identified a mechanism that allowed: - returning arbitrary files from anywhere in the web application - processing any file in the web application as a JSP Further, if the web application allowed file upload and stored those files within the web application (or the attacker was able to control the content of the web application by some other means) then this, along with the ability to process a file as a JSP, made remote code execution possible. It is important to note that mitigation is only required if an AJP port is accessible to untrusted users. Users wishing to take a defence-in-depth approach and block the vector that permits returning arbitrary files and execution as JSP may upgrade to Apache Tomcat 9.0.31, 8.5.51 or 7.0.100 or later. A number of changes were made to the default AJP Connector configuration in 9.0.31 to harden the default	Upgrade to the required version as listed in the description.

		configuration. It is likely that users upgrading to 9.0.31, 8.5.51 or 7.0.100 or later will need to make small changes to their configurations.	
hibernate-validator-6.0.18.Final.jar	**CVE-2020-10693**	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.	Apply the Critical Patch Update, ASAP.
spring-web-5.2.3.RELEASE.jar	**CVE-2016-1000027** (OSSINDEX) [CVE-2016-1000027] CWE-502: Deserialization of Untrusted Data	Pivotal Spring Framework through 5.3.16 suffers from a potential remote code execution (RCE) issue if used for Java deserialization of untrusted data. Depending on how the library is implemented within a product, this issue may or not occur, and authentication may be required. NOTE: the vendor's position is that untrusted data is not an intended use case. The product's behavior will not be changed because some users rely on deserialization of trusted data.	Upgrade to version: 6.0.13
spring-beans-5.2.3.RELEASE.jar	**CVE-2022-22965** (OSSINDEX) [CVE-2022-22965] CWE-94: Improper Control of Generation of Code ('Code Injection')	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.	Upgrade to version: 6.0.13
spring-webmvc-5.2.3.RELEASE.jar	**CVE-2021-22060** (OSSINDEX) [CVE-2021-22060] CWE-117: Improper Output Neutralization for Logs	In Spring Framework versions 5.3.0 - 5.3.13, 5.2.0 - 5.2.18, and older unsupported versions, it is possible for a user to provide malicious input to cause the insertion of additional log entries. This is a follow-up to CVE-2021-22096 that protects against additional types of input	Upgrade to version: 6.0.13

		and in more places of the Spring Framework codebase.	
spring-context-5.2.3.RELEASE.jar	**CVE-2022-22968** (OSSINDEX) [CVE-2022-22968] CWE-178: Improper Handling of Case Sensitivity	In Spring Framework versions 5.3.0 - 5.3.18, 5.2.0 - 5.2.20, and older unsupported versions, the patterns for disallowedFields on a DataBinder are case sensitive which means a field is not effectively protected unless it is listed with both upper and lower case for the first character of the field, including upper and lower case for the first character of all nested fields within the property path.	Upgrade to version: 6.0.13
spring-expression-5.2.3.RELEASE.jar	**CVE-2022-22950** (OSSINDEX)	n Spring Framework versions 5.3.0 - 5.3.16 and older unsupported versions, it is possible for a user to provide a specially crafted SpEL expression that may cause a denial of service condition.	Upgrade to version: 6.0.13

Mitigation Plan

Input validation needs to be done with any transfer of data between files. This prevents manual input by a user and arising security breaches.

Update dependencies to the latest version. This will solve most – if not all – of the vulnerabilities listed in the dependency report.

References

UNCTAD. (2021, Dec 12). *Data Protection and Privacy Legislation Worldwide*. Retrieved from unctad.org: https://unctad.org/page/data-protection-and-privacy-legislation-worldwide