

**Artemis Financial Vulnerability Assessment Report**

Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comments** |
| **1.0** | **5-27-23** | **Jason Decampo** |  |

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

Jason Decampo

* Interpreting Client Needs

What is the value of secure communications to the company?

Financial consulting firms offering their clients individualized services would require working with sensitive financial data. Client information is kept secure from unauthorized access due to an isolated communication line. The company's reputation as a dependable and trustworthy service provider would depend on this. Secure communication is crucial for internal discussions within the financial consulting firm as well as for interactions with clients. Consultants frequently need to securely collaborate on projects and communicate sensitive information. Using encrypted communication solutions increases operational security by ensuring that internal discussions, document sharing, and data exchanges are shielded from unauthorized access.

Does the company make any international transactions?

The scenario mentions "entrepreneurs, businesses, and government agencies around the world" so I would assume international transactions do occur.

Are there governmental restrictions about secure communications to consider?

Strict data protection laws, such as the Gramm-Leach-Bliley Act (GLBA) and the General Data Protection Regulation (GDPR), frequently apply to financial consulting firms. I am unsure if they must follow certain financial regulations as well. Secure communications protect customer data and avert data breaches, which aid the organization in adhering to these standards.

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

The financial sector's most common threat would be cyberattacks due to the valuable information it holds. Secure chat systems, encryption and multiple forms of authentication would help protect sensitive information from being intercepted or accessed by unauthorized parties.

What are the modernization requirements that you must consider?

The software used by financial consulting organizations may need to be integrated with external financial tools, databases, and APIs. To enable easy data interchange, integration features such obtaining financial data from market databases or establishing a connection with accounting software for data analysis should be taken into consideration as well. A remote access feature to allow consultants to work from anywhere, across a range of devices and platforms, could be beneficial to the firm's modernization.

* Areas of Security

Cryptography - Cyberattacks including data breaches, unauthorized access, and man-in-the-middle attacks are all prevented by cryptography. The company can reduce the probability of a successful attack by putting strong encryption algorithms, secure key management procedures, and other cryptographic protections into place.

Input validation - This ensures that any data entered into the system is accurate and complies with the necessary constraints for structure, format, and limitations.

Code Quality - This enables the program to run consistently, delivering correct results and reducing errors. Code that has been carefully written and tested lowers the possibility of data corruption, calculation errors, and system malfunctions.

Code Error - Security flaws are created by coding mistakes and could be used by attackers. Input validation flaws or injection vulnerabilities are examples of security-related faults that might result in data breaches or allow unauthorized access to sensitive financial data.

APIs - These offer defined methods and protocols for data retrieval and processing, enabling easy data flow and integration. For analysis and decision-making, it allows the software to access current market data, client account information, or other relevant data sources.

* Manual Review

I found the following vulnerabilities inside of the CRUDController:

A lack of input validation - the name parameter received via the @RequestParam annotation. Potential security flaws like input validation bypass, SQL injection, or cross-site scripting (XSS) attacks could result from this.

Lack of Authentication and Authorization - The provided code does not include any authentication or authorization mechanisms.

Insecure Data Transmission - When working with sensitive material, it's crucial to use secure protocols like HTTPS to encrypt the data during transmission and shield it from possible hacking or eavesdropping.

* Static Testing

tomcat-embed-core-9.0.30.jar - The refactoring present in Apache Tomcat 9.0.28 to 9.0.30, 8.5.48 to 8.5.50 and 7.0.98 to 7.0.99 introduced a regression. The result of the regression was that invalid Transfer-Encoding headers were incorrectly processed leading to a possibility of HTTP Request Smuggling if Tomcat was located behind a reverse proxy that incorrectly handled the invalid Transfer-Encoding header in a particular manner. Such a reverse proxy is considered unlikely. Update to 10.1.0

spring-webmvc-5.2.3.RELEASE.jar - 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 and in more places of the Spring Framework codebase. Update to current version.

spring-web-5.2.3.RELEASE.jar - 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. Update to current version.

spring-expression-5.2.3.RELEASE.jar - In 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. Update to current version.

spring-context-5.2.3.RELEASE.jar - 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. Update to current version.

spring-boot-autoconfigure-2.2.4.RELEASE.jar - \*\* UNSUPPORTED WHEN ASSIGNED \*\* spring-boot versions prior to version v2.2.11.RELEASE was vulnerable to temporary directory hijacking. This vulnerability impacted the org.springframework.boot.web.server.AbstractConfigurableWebServerFactory.createTempDir method. NOTE: This vulnerability only affects products and/or versions that are no longer supported by the maintainer. Update to current version.

spring-boot-2.2.4.RELEASE.jar - \*\* UNSUPPORTED WHEN ASSIGNED \*\* spring-boot versions prior to version v2.2.11.RELEASE was vulnerable to temporary directory hijacking. This vulnerability impacted the org.springframework.boot.web.server.AbstractConfigurableWebServerFactory.createTempDir method. NOTE: This vulnerability only affects products and/or versions that are no longer supported by the maintainer. Update to current version.

spring-beans-5.2.3.RELEASE.jar - 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. Deploy as Spring Boot exe.

snakeyaml-1.25.jar - The Alias feature in SnakeYAML before 1.26 allows entity expansion during a load operation, a related issue to CVE-2003-1564. Update to current version.

logback-core-1.2.3.jar - 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. Update to current version.

log4j-api-2.12.1.jar - 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. Update to current version.

jackson-databind-2.10.2.jar - A flaw was found in FasterXML Jackson Databind, where it did not have entity expansion secured properly. This flaw allows vulnerability to XML external entity (XXE) attacks. The highest threat from this vulnerability is data integrity. Update to current version.

hibernate-validator-6.0.18.Final.jar - 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. Update to version 6.0.20.

bcprov-jdk15on-1.46.jar - 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. Update to current version.

* Mitigation Plan

Update all systems noted in static testing to current versions unless otherwise specified. Utilize HTTPS for encryption. Implement authentication and authorization mechanisms. Utilize input validation for user inputs.