

**Artemis Financial Vulnerability Assessment Report**

Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comments** |
| **1.0** | **[3/18/23]** | **[Thomas Keller]** |  |

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

Thomas Keller

* Interpreting Client Needs

Security is of the utmost importance, considering that the application is dealing with the financial resources of clients. Hackers gaining access to these funds could be detrimental to both the business and the client. Connections must be secure for the clients safety. Transactions overseas must also adhere by the governments security measures. The use of open source libraries is allowed to make coding easier, however, these come with many security issues. These can be avoided or fixed, but caution must be used.

* Areas of Security

Secure API interactions are important because attackers can intercept these and possibly gain access to an account. Connections should be encrypted to ensure that customers can access their information securely and others cannot. Client/Server connections should be secure, as this is when many hacks can happen. Errors should be handled accordingly, quality is always of great importance, and encapsulation of code is a necessity.

* Manual Review

As in the CRUDController class, where the business logs in by entering their name, there could be more user validation, such as a password, so that their accounts can be secured. myDateTime could also be encapsulated, as its variables are not privatized. While this may not make a huge difference, it is still important to make secure data structures. The customers account balance is also not privatized, which could be a breach of security or potential leak. Another layer of security could be added by handling errors upon login. Enough invalid attempts could lock the users account for security purposes.

* Static Testing

Upon running the Maven dependency check, there were quite a few vulnerabilities within. here are the vulnerabilities with the corresponding codes.

bcprov-jdk15on-1.46.jar : CVE-2013-1624, CVE-2015-6644, CVE-2015-7940, CVE-2016-1000338, CVE-2016-1000339, CVE-2016-1000341, CVE-2016-1000342, CVE-2016-1000343, CVE-2016-1000344, CVE-2016-1000345, CVE-2016-1000346, CVE-2016-1000352, CVE-2017-13098, CVE-2018-5382, CVE-2020-0187, CVE-2020-26939

* This dependency is vulnerable to injection attacks, as well as attacks on the cryptography itself. Many of these vulnerabilities are fixed by updating to the newest version.

spring-boot-2.2.4.RELEASE.jar : CVE-2022-27772

* Vulnerable to temporary directory hijacking prior to version 2.2.11. Vulnerable method can be avoided or the dependency can be updated.

logback-core-1.2.3.jar : CVE-2021-42550

* A malicious configuration can allow attackers to execute arbitrary code. This is fixed in newer versions.

log4j-api-2.12.1.jar : CVE-2020-9488, CVE-2021-44228, CVE-2021-44832, CVE-2021-45046, CVE-2021-45105

* This dependency is susceptable to man in the middle attacks, attacker controlled LDAP and JNDI endpoints are a problem, and RCE attacks. Some of these vulnerabilities are for specefic methods, while updating can also improve security.

snakeyaml-1.25.jar : CVE-2017-18640, CVE-2021-4235, CVE-2022-1471, CVE-2022-25857, CVE-2022-3064, CVE-2022-38749, CVE-2022-38750, CVE-2022-38751, CVE-2022-38752, CVE-2022-41854

* This dependency allows entity expansion, and maliciously crafted YAML files can cause serious harm as well as data leaks. Validation of files is extremely important as well as an update.

jackson-databind-2.10.2.jar : CVE-2020-25649, CVE-2020-36518, CVE-2022-42003, CVE-2022-42004

* Entity expansion is not secured properly, so XXE attacks can threaten data security. Other issues are resolved in further updates.

tomcat-embed-core-9.0.30.jar : CVE-2019-17569, CVE-2020-11996, CVE-2020-13934, CVE-2020-13935, CVE-2020-13943, CVE-2020-17527, CVE-2020-1935, CVE-2020-1938, CVE-2020-8022, CVE-2020-9484, CVE-2021-24122, CVE-2021-25122, CVE-2021-25329, CVE-2021-30640, CVE-2021-33037, CVE-2021-41079, CVE-2021-43980, CVE-2022-29885, CVE-2022-34305, CVE-2022-42252

* This dependency leads to many denial of service vulnerabilities and data leaks, which obviously must be stopped. newer updates resolve some of these issues, but further action must be taken.

hibernate-validator-6.0.18.Final.jar : CVE-2020-10693

* can allow attackers to bypass input sanitation controls. This can be updated to fix the problem.

spring-web-5.2.3.RELEASE.jar

: CVE-2016-1000027, CVE-2020-5421, CVE-2021-22096, CVE-2021-22118

* Suffers an RCE issue if ued for Java deserialization of untrusted data. Input review and validation can solve this.

spring-beans-5.2.3.RELEASE.jar : CVE-2022-22965

* 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.

spring-webmvc-5.2.3.RELEASE.jar : CVE-2021-22060

* Vulnerable to insertion of additional log entries. Fixed in newer versions.

spring-context-5.2.3.RELEASE.jar : CVE-2022-22968

* improper handling of case sensitivity can lead to ineffective protection. Updating or proper handling can fix this.

spring-expression-5.2.3.RELEASE.jar : CVE-2022-22950

* ` It is possible for a user to provide a specially crafted SpEL expression that may cause a denial of service condition. Fixed in new updates.
* Mitigation Plan

A large part of the vulnerabilities in the code are simply fixed by updating the dependencies as stated above. Updating these dependencies will be a vital part of the mitigation plan. After updating to the latest security fix, the tomcat-embed-core will most likely still have vulnerabilities that must be dealt with or avoided individually.