

# Artemis Financial Vulnerability Assessment Report

Southern New Hampshire University

Module Three Project One

Emily Nagorski

Dr. Lyon

CS305-T1165 – Software Security

9/13/22

Table of Contents

[Document Revision History 3](#_Toc32574607)

[Client 3](#_Toc32574608)

[Instructions 3](#_Toc32574609)

[Developer 4](#_Toc32574610)

[1. Interpreting Client Needs 4](#_Toc32574611)

[2. Areas of Security 4](#_Toc32574612)

[3. Manual Review 5](#_Toc32574613)

[4. Static Testing 5](#_Toc32574614)

[5. Mitigation Plan 6](#_Toc32574615)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **9/13/22** | **Emily Nagorski** | **Updated security issues and mitigation plan** |

## 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

Emily Nagorski

## Interpreting Client Needs

Artemis financial is a consulting company, that has a duty to their customers to develop financial plans based on their customers needs and scenarios. Currently they are requesting to modernize their operation, which is where we come in. Due to the nature of their business, security is a top priority since personal, and financial information is involved. Artemis Financial does not specifically claim to do international business, however considering the nature of their accounts I think it would be best to keep the potential for international transactions on the table and treat this project as if they do. We do know that the nature of their business includes, savings, retirement, investments and insurance accounts and along with each of these things come restrictions and regulations in regard to communications. The U.S. does not have any specific data security laws, however there are federal restrictions and state laws that need to be followed. (Enterprise 2022) One such law is that companies have a duty to their customers to notify them of a data breach within 60 days of the attack. Additionally, it would be wise to consider HIPPA, Gramm-Leach-Biley Act and the Homeland Security Act. Each of these protect citizens and hold companies accountable for securing their systems and ensuring that customer information is safe and protected to an acceptable extent. (Enterprise 2022) Cyber security will need to be of the highest priority as Artemis Financial is handling extreme amounts of personal, medical, and financial information for each of their customers. Because of the amount of personal information they have on hand, they are at risk of ransomware attacks, and other techniques of data breaching in order to obtain a path into funds, and investor information. (Hoffman 2022) When considering modernization, its vital we frequently test our dependencies for threats, and either patch the hole, upgrade or downgrade where the threat hole does not exist or find another dependency altogether if no solution can be found.

## Areas of Security

APIs: In the event of an insecure API, this insecurity can be the target for hackers and put the personal information of clients at risk. APIs are at risk of man-in-the-middle, and distributed denial-of-service attacks when weaknesses exist, therefore it is in the best interest of our client to make sure this is one of our top priorities. (Fortinet 2022)

Cryptography: In the likely event that personal information needs to be shared and provided virtually it will be important to have documents and emails encrypted. This information can be kept secure by transforming it for unintended recipients, making it virtually useless for them. (Fruhlinger 2022) Having a focus on cryptography with help ensure the integrity of data has not been maliciously altered, and guarantee that the intended sender really was the one who sent it, as well as authenticate the receiver upon delivery. (Fruhlinger 2022)

Code Quality: Simply put quality code is secure code. This is a factor I believe should be the focus of every security and development team. Security starts with having clear, clean and concise code. Messy or untested code could lead to overflow incidents that compromise the security of the clients. (Gold 2020) We need to ensure that all our code is useful and fully tested before deployment to avoid serious security issues.

Encapsulation: Encapsulation is a method within object oriented programming. If used properly it can aid in the control of access that a user has to certain classes, however if used incorrectly it can open the product up to security risks. We want to make sure our data structures are as secure as possible and therefore it is in our best interest to consider encapsulation in that process while also doing rigorous testing to ensure its security.

## Manual Review

In RestServiceApplication.java we are importing springframework.boot on lines 3 and 4. A quick search of the CVE list database outline 15 potential security issues. These vulnerabilities allow for potential remote code execution, authentication bypass, and several other attacks. Most of these vulnerabilities can be avoided by specifying a version of the framework.

## Static Testing

**Bouncy Castle** HIGH CVE-2016-1000352, CVE-2016-1000346, CVE-2016-1000345, CVE-2016-100344, CVE-2016-1000343, CVE-2016-1000342, CVE-2016-1000341, CVE-2016-1000339, CVE-2016-1000338, CVE-2018-5382, CVE-2017-13098, CVE-2013-1624

* Some versions allow for invalid keys to be used to reveal details about a party’s private information, others are vulnerable to padding oracle attack.
* Many of the affected modes have been removed, those that have not been removed have been updated and patched in later versions. Some have not been removed or patched, and instead the authors of the library recommend to avoid its use unless necessary or safe to do so in a specific instance.

**Hibernate Validator** MEDIUM CVE-2020-10693

* A bug exists in the interpolation processor and it allows for invalid EL expressions to verify as valid. This enables attackers to bypass security restrictions.
* It is recommended to upgrade to a newer version and re-pushing your application to production. No additional mitigation plans exist at this time. (IBM 2020)

**Jackson Databind** HIGH CVE-2020-36518, CVE-2020-25649

* This vulnerability allows for a Java StackOverFLow exception, therefore allowing the attackers to deny service or external entity attacks putting data at risk.
* The code has since been updated to fix the vulnerability, in version 2.9.8

**Log4j** LOW CVE-2021-44832, CVE-2021-45105, CVE-2021-45046, CVE-2021-44228, CVE-2020-9488

* These vulnerabilities pertain to logging within an application or product where an attacker could perform remote code executions attacks or denial of service.
* Many of the logging issues have been solved in newer versions, as this was a widely reaching vulnerability with a high impact.

**Logback-core** MEDIUM CVE-2021-42550

* This vulnerability allows the attacker to edit configurations and execute arbitrary code from LDAP servers.
* This vulnerability exists in logback version 1.2.7 and prior, updated versions should not have this issue

**SnakeYaml** HIGH CVE-2022-38751, CVE-2022-38750, CVE-2022-38749, CVE-2022-25857, CVE-2017-18640

* When the parser is running user supplied input or untrusted YAML files, the attacker could supply content causing the parser to crash or opening the opportunity for a denial of service attack.
* Updating YAML to a nonvulnerable version is the simplest way to avoid this issue

**Spring-boot** HIGH CVE-2022-27772

* Prior versions of spring-boot are vulnerable to temporary directory hijacking
* Versions v2.2.11.RELEASE and on should not have this issue, and as noted in the NVD, this vulnerability only affects products and versions that are no longer supported

**Spring-core & Spring-web** CRITICAL CVE-2022-22971, CVE-2022-22970, CVE-2022-22968, CVE-2022-22965, CVE-2022-22950, CVE-2021-22060, CVE-2021-22096, CVE-2021-22118, CVE-2020-5421, CVE-2016-1000027

* Applications using the spring-core dependency are vulnerable to denial of service attacks, remote code executions attacks or able to view and modify files
* This vulnerability can be avoided by updating to versions 5.3.18+ and 5.2.20+ (Chierici 2022)

**Apache Tomcat** CRITICAL CVE-2022-34305, CVE-2022-29885, CVE-2021-41079, CVE-2021-33037, CVE-2021-30640, CVE-2021-25329, CVE-2021-25122, CVE-2021-24122, CVE-2020-17527, CVE-2020-13943, CVE-2020-13935, CVE-2020-13934, CVE-2020-8022, CVE-2020-11996, CVE-2020-9484, CVE-2020-1938, CVE-2020-1935, CVE-2019-17569

* The apache tomcat vulnerability allows for cross scripting attacks, denial of service, request smuggling and source code disclosure, as well as allows for an attacker to authenticate using variations of valid user information to bypass protections.
* Apache Tomcat does a thorough job of documenting the affected versions and recommends using unaffected versions. Additionally for vulnerabilities that have been incorrectly reported against Tomcat, they often offer patches to mitigate those security issues. (Apache 2022)

## Mitigation Plan

Generally speaking, the approach when it comes to security and vulnerability reports would be to identify and disregard any false positives. From there we will see which vulnerabilities can be remediated by either upgrading or downgrading the version to a non vulnerable version. Third, if no non vulnerable version exists we will want to see if a patch exists, or if we can create a patch. As a last resort we will remove any vulnerable dependencies that cannot be patched or remedied.

Bouncy Castle: We need to first ensure we are not using a vulnerable or unpatched version. We can test a new, non vulnerable version to make sure it does not break the code we have, and still does what we need it to do. We are currently running version 1.46, upgrading to version 1.55 should be our first step to fix this.

Hibernate Validator: Currently the only mitigation plan that exists is updating to a newer version. We will need to update and then test our product to ensure that this update does not break anything along the way. We will need to upgrade to version 6.1.5.

Jackson Databind: This vulnerability was fixed in version 2.9.8 and will therefore require us to update this dependency and test it with our code before pushing to production. We should first test upgrading to 2.13.2.1.

Log4j: Due to the number of implications the log4j vulnerability can cause, it is in our best interest to update to a newer version to ensure that our application is not susceptible to these attacks. Please upgrade to version 2.15.0.

Logback-core: We are currently running 1.2.3, and we will need to update to a version after 1.2.7 in order to mitigate this vulnerability.

SnakeYAML: The simplest way to avoid this dependency would be to update to version 1.31, we are currently running 1.25.

Spring-boot: We are currently running an unsupported version, 2.2.4.RELEASE. It is the recommendation to upgrade to 2.2.11 or newer.

Spring-core & Spring-web: We will need to update to version 5.3.18 to avoid this dependency vulnerability.

Apache Tomcat: Our application is currently running 9.0.30. Apache Tomcat recommends running updated versions to avoid these vulnerabilities. We should first upgrade and test version newer than 9.0.30.

Citations

Apache Tomcat. (2022). *Apache Tomcat®*. Apache Tomcat® - Apache Tomcat 9 vulnerabilities. Retrieved September 15, 2022, from https://tomcat.apache.org/security-9.html

Chierici, S. (2022, April 21). *Critical vulnerability in spring core: CVE-2022-22965 a.k.a. spring4shell*. Sysdig. Retrieved September 15, 2022, from https://sysdig.com/blog/cve-2022-22965-spring-core-spring4shell/

Enterprise Engineering Solutions. (2022, July 13). Cybersecurity laws and regulations in US 2022. Retrieved September 14, 2022, from https://www.eescorporation.com/cybersecurity-laws-and-regulations-in-us/

Fortinet. (2022). *What is API security?* Fortinet. Retrieved September 14, 2022, from https://www.fortinet.com/resources/cyberglossary/api-security

Fruhlinger, J. (2022, May 22). *What is cryptography? how algorithms keep information secret and safe*. CSO Online. Retrieved September 14, 2022, from https://www.csoonline.com/article/3583976/what-is-cryptography-how-algorithms-keep-information-secret-and-safe.html

Gold Sky Security. (2020, December 17). *The advantages of secure coding best practices*. GoldSky Security | Cyber Security Solutions. Retrieved September 14, 2022, from https://www.goldskysecurity.com/security-by-design-the-advantages-of-secure-coding-best-practices/

Hoffman, K. (2022, June 23). *Financial advisers face increased breach threats with remote work, Sensitive Data*. SC Media. Retrieved September 14, 2022, from https://www.scmagazine.com/analysis/breach/financial-advisers-face-increased-breach-threats-with-remote-work-sensitive-data

IBM. (2020, December 9). *Security Bulletin: Vulnerability in Hibernate Validator affects Liberty for Java for IBM Cloud (CVE-2020-10693)*. Security bulletin: Vulnerability in hibernate validator affects liberty for Java for IBM Cloud (CVE-2020-10693). Retrieved September 15, 2022, from https://www.ibm.com/support/pages/security-bulletin-vulnerability-hibernate-validator-affects-liberty-java-ibm-cloud-cve-2020-10693#:~:text=Vulnerability%20Details&text=DESCRIPTION%3A%20Hibernate%20Hibernate%20Validator%20could,in%20the%20message%20interpolation%20processor.