****

# Artemis Financial Vulnerability Assessment Report

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 4](#_Toc32574613)

[4. Static Testing 4](#_Toc32574614)

[5. Mitigation Plan 4](#_Toc32574615)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **03/14/2023** | **Christine Emerson** | **Artemis Financial wants to modernize their operations. As a crucial part of the success of their custom software, they also want to use the most current and effective software security. Artemis Financial has a RESTful web application programming interface (API). They are seeking Global Rain’s expertise about how to protect the organization from external threats.** |

## Client



## Interpreting Client Needs

There are many considerations that need to be thought out to interpret Artemis Financials needs such as:

**Secure Communications**

Artemis Financial is a consulting firm that develops individualized financial plans for its clients. Among the financial plans they offer are savings, retirement, investments, and insurance. A financial institution must secure all transactions and communications more than any other business. Secure communication refers to communication between two entities without wanting a third party to hear what they are saying. Artemis Financial and its clients must be able to communicate securely to ensure end-to-end protection. In particular, Artemis Financial must maintain a secure channel for exchanging client files. The financial plans provided by Artemis Financial are tailored to the individual. Therefore, a plan must be designed based on information about an individual, so data and communication must be exchanged. If a company cannot keep sensitive information safe and secure from prying eyes, people will not be willing to share sensitive information with them. That is why having secure communications is pertinent to Artemis Financial.

**International Transactions**

Based on the documents provided by Artemis Financial, nothing indicates that international transactions will occur. In addition, nothing in the existing code base suggests that there will not be international transactions either. The principles of security dictate that even though it is not explicitly stated, we should still regard it as a possibility since it is also not expressly excluded. Therefore, Artemis Financial will require high levels of security for domestic and foreign communications.

**Governmental Restrictions**

Finance is one of the most highly regulated industries in the world, which should come as no surprise. As a financial industry dealing with sensitive data, Artemis Financial must maintain high data security standards. Aside from GLBA/FFIEC, other compliance laws, including PCI DSS, SOX, and state privacy laws, can also be relevant. Despite this, they all share that encryption and key management can make the difference between notifying the public and having a safe harbor in the event of a data breach. Likely, Artemis Financial will routinely handle nonpublic personal information such as income, credit score, collection history, and the PII of relatives alongside obvious PII like names, addresses, and social security numbers.

GLBA mandates explicitly that Artemis Financial protects NPIs with appropriate security and confidentiality measures since it performs business in the US. Customer records and information are expected to be secured and confidential, protected against any anticipated threats or hazards threatening the security or integrity of these records, and protected from unauthorized access to information that could result in substantial harm to customers. In addition, Artemis Financial needs to use encryption during the transit and storage of sensitive information to mitigate the risk of it being disclosed or altered. To comply with FFIEC rules and the GLBA, we need to encrypt any sensitive information given to Artemis Financial by an individual to receive financial products and services, such as name, address, income, Social Security number, and other private details as well as any information received about an individual from a transaction involving the financial products or services( the fact that an individual is your customer, account numbers, payment history, deposit/investment balances, etc.).

**External Threats**

Modern society is plagued by external threats that grow with the advancement of technology. For example, Artemis Financial works with sensitive financial and personal information, substantially increasing the risk of an attack. Phishing, ransomware, SQL injections, local file inclusion, cross-site scripting, OGNL java injections, DDoS attacks, and supply chain attacks are some of the most. In addition, recent years have seen the rise of bad bots as a significant threat to application security. A bot can be programmed to attempt passwords and usernames repeatedly, resulting in a denial of service if done rapidly and repeatedly. Furthermore, there are risks associated with authorized users accessing data they shouldn't. For example, this could be an end user who gains access to administrator files. Defending against potential attacks would also require a robust API.

**Modernization Requirements**

It is possible for open-source libraries and actively maintained software to pose a threat to the system without being anticipated. There are many challenges associated with modernization. For example, a company such as Artemis Financial deals with financial planning. Another company may be able to provide them with stock and bond information compiled from their records. A system integration could result in additional threats. There may be additional risks posed by third-party software if their APIs and other structures lack security. Modernizing web applications also requires consideration of evolving web technologies. The capabilities of web applications are increasing now; however, different applications often require permissions or have vulnerabilities that expose the application to security risks. Furthermore, attackers become increasingly clever, developing new methods for penetrating systems as time passes.

## Areas of Security

* **Input Validation:** Input validation is of paramount importance. The program will provide users with access to confidential and private data. Consequently, anyone who accesses this data must be authorized to do so. Credentials must be appropriately validated before the program authorizes access to the system. Data breaches and phishing attacks will be prevented through compliance with this section, which will also lead to client trust in the organization. In addition, the authentication of each user allows Artemis Financial to ensure compliance with all global government regulations.
* **API’s:** A robust API must be implemented for Artemis Financial to run in an outside environment, such as a web browser. The RESTful web application programming interface (API) used in this application is crucial to its success. Data and methods will be determined by the API, along with their acceptability. Aside from that, this project may use third-party software on which this software may rely. For REST APIs to be secure, a secure socket layer is essential. In this way, the API will be safer and less likely to be attacked maliciously. In addition to their flexibility, REST APIs can handle various requests and send data in multiple formats. Despite their size or capabilities, REST APIs are designed to communicate between any two pieces of software. An application's REST API will be able to handle the increasing number and variety of requests as it grows and adds more resources. Since REST APIs are built with existing web technologies, creating and using them is relatively straightforward. A REST API only requires the URL of a resource to be requested. We need to be extra cautious with RESTful web services to prevent credentials from leaking. URLs should not contain passwords, security tokens, or API keys because these can be captured in web server logs, making them incredibly valuable. Request headers or request bodies should contain sensitive data in POST / PUT requests.
* **Cryptography:** A cryptographic system aims to guarantee authentication, privacy, confidentiality, integrity, and non-repudiation. As well as protecting data from theft and alteration, cryptography can authenticate users. Financial institutions such as Artemis Financial extensively use cryptographic techniques to meet compliance requirements and implement data security norms. Additionally, cryptography is regulated as a foreign export item because international transfers are uncertain, so if Artemis Financial does business overseas and uses secure, encrypted communications, it will have to take steps to secure it while complying with the laws of both the United States and the country it is seeking to reach.

* **Code Quality:** The code's quality impacts the software's quality as a whole. A codebase's quality affects its safety, security, and reliability. Artemis Financial’s application is a safety-critical system, so high quality is crucial. Code that stands the test of time is clean and of high quality. The lifespan of lousy code is short because it is of low quality. As a rule of thumb, good code performs as expected, follows a uniform style, is easily understandable, well documented, and is testable.
* **Code Error:** Artemis Financial won't have to worry about its user information being exposed or accessed if error handling is implemented in the API. This way, they can know what areas of the API need to be fixed. In addition, runtime errors can be resolved or minimized by adopting reasonable countermeasures, which allows them to recover efficiently from unexpected errors.

## Manual Review

Below are my findings following a thorough manual review:

* Authentication of account owners does not exist.
* The system does not appear to use HTTPS, a critical component of internet security.
* As far as I could tell, cryptography was not part of the code base. Since I did not find any encryption, this was a quick review.
* Upon evaluating how this API worked, I looked for it to see if it was available, but I didn't find it. Despite the lack of a visual interface or API, the program can accept data. This program accepts that data in an unsecured manner; instead of receiving data as a POST method, the data is received as a URL, which can leak into the browser history and be exploited. Aside from that, there is no obvious way to interact with this program. An interactive portal or document would be necessary if this were published. RESTful APIs must interact in a way that is discernible to the user.

The following are additional findings I have provided, along with visuals from the code base and explanations:

***MavenWrapperDownloader.java***

***Text

Description automatically generated***

* An attacker may be able to read arbitrary files by exploiting the path traversal vulnerability in the input since the input is not validated.
* When an exception occurs between opening/using and closing a resource, the resources will be closed in the try block, which can lead to leaking resources.

* Catching java.lang.Throwable will also catch JVM internal errors like OutOfMemoryErrors or StackOverflowErrors; this is a bad practice since it is not likely the intended behavior. Rather than catching a more general exception, it is recommended to catch one that is less general. The issue can be found here:

Graphical user interface, text, application

Description automatically generated

***customer.java***

***Graphical user interface

Description automatically generated with low confidence***

* An attacker may be able to access confidential information in multiple instances due to incorrect access control definitions within the customer class. In the example of the account\_balance variable, the variable inherits the public access level. Therefore, saving personal data as a private variable is recommended to prevent external programs from reading it. Even though the account number variable is accessible only to this class, attackers could still penetrate the program and obtain the account number's secret value by using the "showInfo" method, thereby jeopardizing the confidentiality of the client's data, and corrective action should be taken.

***DocData.java***

***Graphical user interface, text, application, email

Description automatically generated***

* An attacker could gain access with a hardcoded string password (found in this class), gain access to data, take down sites or apps, or hold them for ransom if given this privilege. In addition, this class had a method for reading documents when given a key and a value. If the code containing the credentials is reused across multiple projects, they will all be compromised; this includes a try/catch block but has no actual error handling. Since no other classes handle errors, evaluating them was unnecessary.

Lastly, I examined the quality of the code. Though not entirely functional, the existing code is satisfactory. Much more work must be done for this program to be fully developed. APIs do not inform end users what is expected, as mentioned. There is a need to implement input validation. Input should be moved from the URL to the POST method to prevent data from leaking into the browser history. API and input validation are subsets of code quality. This aspect presents a challenge in coding the subsets and coding them securely.

## Static Testing

After running static testing, the following 13 dependencies and 101 vulnerabilities have been found:

**Scan information:**

Text

Description automatically generated

**Summary:**

Graphical user interface, text, application, email

Description automatically generated

* **Dependency:** [bcprov-jdk15on-1.46.jar](file:///C:\Users\chris\Desktop\SNHU%20Classes\Tenth%20Term\CS%20305\Week%20Three\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l1_991c96a4e31e6c19e2b9136c8955bd423f2dc4c7)

**Vulnerability ID:** cpe:2.3:a:bouncycastle:bouncy-castle-crypto-package:1.46:\*:\*:\*:\*:\*:\*:\*

cpe:2.3:a:bouncycastle:bouncy\_castle\_crypto\_package:1.46:\*:\*:\*:\*:\*:\*:\*  
[cpe:2.3:a:bouncycastle:legion-of-the-bouncy-castle-java-crytography- api:1.46:\*:\*:\*:\*:\*:\*:\*](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)  
cpe:2.3:a:bouncycastle:the\_bouncy\_castle\_crypto\_package\_for\_java:1.46:\*:\*:\*:\*:\*:\*:\*

**Description:** The Bouncy Castle Crypto package is a Java implementation of cryptographic

algorithms. This jar contains JCE provider and lightweight API for the Bouncy Castle Cryptography APIs for JDK 1.5 to JDK 1.7**.**

**Code:** [CVE-2016-1000338](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000338) , [CVE-2016-1000342](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000342), [CVE-2016-1000343](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000343), [CVE-2016-1000344](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000344),

[CVE-2016-1000352](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000352), [CVE-2016-1000341](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000341), [CVE-2016-1000345](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000345),  [CVE-2017-13098](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2017-13098), [CVE-2020-15522](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-15522), CVE-2020-0187 (OSSINDEX), [CVE-2016-1000339](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000339), CVE-2020-26939 (OSSINDEX), [CVE-2015-7940](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2015-7940), [CVE-2018-5382](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2018-5382), [CVE-2013-1624](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2013-1624), [CVE-2016-1000346](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000346), CVE-2015-6644 (OSSINDEX)

* **Dependency:** [hibernate-validator-6.0.18.Final.jar](file:///C:\Users\chris\Desktop\SNHU%20Classes\Tenth%20Term\CS%20305\Week%20Three\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l3_7fd00bcd87e14b6ba66279282ef15efa30dd2492)

**Vulnerability ID:** [cpe:2.3:a:redhat:hibernate\_validator:6.0.18:\*:\*:\*:\*:\*:\*:\*](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)

**Description:** Hibernate's Bean Validation (JSR-380) reference implementation.

**Code:** [CVE-2020-10693](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-10693)

* **Dependency:** [jackson-databind-2.10.2.jar](file:///C:\Users\chris\Desktop\SNHU%20Classes\Tenth%20Term\CS%20305\Week%20Three\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l5_0528de95f198afafbcfb0c09d2e43b6e0ea663ec)

**Vulnerability ID:** [cpe:2.3:a:fasterxml:jackson-databind:2.10.2:\*:\*:\*:\*:\*:\*:\*](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)

cpe:2.3:a:fasterxml:jackson-modules-java8:2.10.2:\*:\*:\*:\*:\*:\*:\*

**Description:** General data-binding functionality for Jackson: works on core streaming API.

**Code:** [CVE-2020-25649](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-25649), [CVE-2020-36518](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-36518),  [CVE-2022-42003](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-42003), [CVE-2022-42004](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-42004)

* **Dependency:** [log4j-api-2.12.1.jar](file:///C:\Users\chris\Desktop\SNHU%20Classes\Tenth%20Term\CS%20305\Week%20Three\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l10_a55e6d987f50a515c9260b0451b4fa217dc539cb)

**Vulnerability ID:** [cpe:2.3:a:apache:log4j:2.12.1:\*:\*:\*:\*:\*:\*:\*](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)

**Description:** The Apache Log4j API

**Code:** [CVE-2020-9488](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-9488)

* **Dependency:** [logback-core-1.2.3.jar](file:///C:\Users\chris\Desktop\SNHU%20Classes\Tenth%20Term\CS%20305\Week%20Three\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l12_864344400c3d4d92dfeb0a305dc87d953677c03c)

**Vulnerability ID:** [cpe:2.3:a:qos:logback:1.2.3:\*:\*:\*:\*:\*:\*:\*](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)

**Description:** logback-core module

**Code:** [CVE-2021-42550](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-42550)

* **Dependency:** [snakeyaml-1.25.jar](file:///C:\Users\chris\Desktop\SNHU%20Classes\Tenth%20Term\CS%20305\Week%20Three\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l14_8b6e01ef661d8378ae6dd7b511a7f2a33fae1421)

**Vulnerability ID:** [cpe:2.3:a:snakeyaml\_project:snakeyaml:1.25:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Asnakeyaml_project&cpe_product=cpe%3A%2F%3Asnakeyaml_project%3Asnakeyaml&cpe_version=cpe%3A%2F%3Asnakeyaml_project%3Asnakeyaml%3A1.25)

**Description:** YAML 1.1 parser and emitter for Java

**Code: CVE-2022-1471 (OSSINDEX),** [CVE-2017-18640](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2017-18640), [CVE-2022-25857](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-25857), [CVE-2022-38749](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-38749), [CVE-2022-38751](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-38751), [CVE-2022-38752](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-38752), [CVE-2022-41854](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-41854), [CVE-2022-38750](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-38750)

* **Dependency:** [spring-boot-2.2.4.RELEASE.jar](file:///C:\Users\chris\Desktop\SNHU%20Classes\Tenth%20Term\CS%20305\Week%20Three\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l15_225a4fd31156c254e3bb92adb42ee8c6de812714)

**Vulnerability ID:** [cpe:2.3:a:vmware:spring\_boot:2.2.4:release:\*:\*:\*:\*:\*:\*](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)

**Description:** Apache License, Version 2.0: https://www.apache.org/licenses/LICENSE-2.0

**Code:** [CVE-2022-27772](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-27772)

* **Dependency:** [spring-boot-starter-web-2.2.4.RELEASE.jar](file:///C:\Users\chris\Desktop\SNHU%20Classes\Tenth%20Term\CS%20305\Week%20Three\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l16_ec75d01d212b5229c16d872fb127744c0ed46ed8)

**Vulnerability ID:** [cpe:2.3:a:vmware:spring\_boot:2.2.4:release:\*:\*:\*:\*:\*:\*](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)

[cpe:2.3:a:web\_project:web:2.2.4:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aweb_project&cpe_product=cpe%3A%2F%3Aweb_project%3Aweb&cpe_version=cpe%3A%2F%3Aweb_project%3Aweb%3A2.2.4)

**Description:** Starter for building web, including RESTful, applications using Spring

MVC. Uses Tomcat as the default embedded container.

**Code:** [CVE-2022-27772](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-27772)

* **Dependency:** [spring-core-5.2.3.RELEASE.jar](file:///C:\Users\chris\Desktop\SNHU%20Classes\Tenth%20Term\CS%20305\Week%20Three\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l17_3734223040040e8c3fecd5faa3ae8a1ed6da146b)

**Vulnerability ID:** [cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Apivotal_software&cpe_product=cpe%3A%2F%3Apivotal_software%3Aspring_framework&cpe_version=cpe%3A%2F%3Apivotal_software%3Aspring_framework%3A5.2.3)

[cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aspringsource&cpe_product=cpe%3A%2F%3Aspringsource%3Aspring_framework&cpe_version=cpe%3A%2F%3Aspringsource%3Aspring_framework%3A5.2.3)  
 [cpe:2.3:a:vmware:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*](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)

**Description:** Spring Core

**Code:** [CVE-2022-22965](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22965),[CVE-2021-22118](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22118), [CVE-2020-5421](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-5421), [CVE-2022-22950](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22950), [CVE-2022-22971](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22971) , [CVE-2022-22968](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22968), [CVE-2022-22970](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22970), [CVE-2021-22060](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22060), [CVE-2021-22096](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22096)

* **Dependency:** [spring-web-5.2.3.RELEASE.jar](file:///C:\Users\chris\Desktop\SNHU%20Classes\Tenth%20Term\CS%20305\Week%20Three\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l18_dd386a02e40b915ab400a3bf9f586d2dc4c0852c)

**Vulnerability ID:** [cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Apivotal_software&cpe_product=cpe%3A%2F%3Apivotal_software%3Aspring_framework&cpe_version=cpe%3A%2F%3Apivotal_software%3Aspring_framework%3A5.2.3)

[cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aspringsource&cpe_product=cpe%3A%2F%3Aspringsource%3Aspring_framework&cpe_version=cpe%3A%2F%3Aspringsource%3Aspring_framework%3A5.2.3)  
 [cpe:2.3:a:vmware:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*](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)  
 [cpe:2.3:a:web\_project:web:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aweb_project&cpe_product=cpe%3A%2F%3Aweb_project%3Aweb&cpe_version=cpe%3A%2F%3Aweb_project%3Aweb%3A5.2.3)

**Description:** Spring Web

**Code:** [CVE-2016-1000027](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000027), [CVE-2022-22965](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22965), [CVE-2021-22118](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22118), [CVE-2020-5421](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-5421), [CVE-2022-22950](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22950), [CVE-2022-22971](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22971), [CVE-2022-22968](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22968), [CVE-2022-22970](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22970), [CVE-2021-22060](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22060),  [CVE-2021-22096](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22096)

* **Dependency:** [spring-webmvc-5.2.3.RELEASE.jar](file:///C:\Users\chris\Desktop\SNHU%20Classes\Tenth%20Term\CS%20305\Week%20Three\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l19_745a62502023d2496b565b7fe102bb1ee229d6b7)

**Vulnerability ID:** [cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Apivotal_software&cpe_product=cpe%3A%2F%3Apivotal_software%3Aspring_framework&cpe_version=cpe%3A%2F%3Apivotal_software%3Aspring_framework%3A5.2.3)

[cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aspringsource&cpe_product=cpe%3A%2F%3Aspringsource%3Aspring_framework&cpe_version=cpe%3A%2F%3Aspringsource%3Aspring_framework%3A5.2.3)  
 [cpe:2.3:a:vmware:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*](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)  
 [cpe:2.3:a:web\_project:web:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aweb_project&cpe_product=cpe%3A%2F%3Aweb_project%3Aweb&cpe_version=cpe%3A%2F%3Aweb_project%3Aweb%3A5.2.3)

**Description:** Spring Web MVC

**Code:** [CVE-2022-22965](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22965), [CVE-2021-22118](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22118), [CVE-2020-5421](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-5421),  [CVE-2022-22950](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22950), [CVE-2022-22971](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22971), [CVE-2022-22968](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22968), [CVE-2022-22970](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22970), [CVE-2021-22060](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22060), [CVE-2021-22096](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22096)

* **Dependency:** [tomcat-embed-core-9.0.30.jar](file:///C:\Users\chris\Desktop\SNHU%20Classes\Tenth%20Term\CS%20305\Week%20Three\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l20_ad32909314fe2ba02cec036434c0addd19bcc580)

**Vulnerability ID:** [cpe:2.3:a:apache:tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*](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)

[cpe:2.3:a:apache\_tomcat:apache\_tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aapache_tomcat&cpe_product=cpe%3A%2F%3Aapache_tomcat%3Aapache_tomcat&cpe_version=cpe%3A%2F%3Aapache_tomcat%3Aapache_tomcat%3A9.0.30)

**Description:** Core Tomcat implementation

**Code:** [CVE-2020-1938](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-1938), [CVE-2020-11996](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-11996),  [CVE-2020-13934](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13934), [CVE-2020-13935](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13935), [CVE-2020-17527](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-17527),  [CVE-2021-25122](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-25122), [CVE-2021-41079](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-41079),  [CVE-2022-29885](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-29885), [CVE-2022-42252](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-42252),  [CVE-2020-9484](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-9484), [CVE-2021-25329](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-25329), [CVE-2021-30640](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-30640),  [CVE-2022-34305](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-34305), [CVE-2021-24122](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-24122), [CVE-2021-33037](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-33037), [CVE-2019-17569](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2019-17569), [CVE-2020-1935](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-1935), [CVE-2020-13943](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13943),  [CVE-2021-43980](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-43980)

* **Dependency:** [tomcat-embed-websocket-9.0.30.jar](file:///C:\Users\chris\Desktop\SNHU%20Classes\Tenth%20Term\CS%20305\Week%20Three\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l22_33157f6bc5bfd03380ebb5ac476db0600a04168d)

**Vulnerability ID:** [cpe:2.3:a:apache:tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*](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)

[cpe:2.3:a:apache\_tomcat:apache\_tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aapache_tomcat&cpe_product=cpe%3A%2F%3Aapache_tomcat%3Aapache_tomcat&cpe_version=cpe%3A%2F%3Aapache_tomcat%3Aapache_tomcat%3A9.0.30)

**Description:** Core Tomcat implementation

**Code:** [CVE-2020-1938](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-1938),  [CVE-2020-8022](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-8022),  [CVE-2020-11996](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-11996), [CVE-2020-13934](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13934),  [CVE-2020-13935](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13935), [CVE-2020-17527](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-17527),  [CVE-2021-25122](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-25122),  [CVE-2021-41079](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-41079), [CVE-2022-29885](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-29885), [CVE-2022-42252](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-42252) , [CVE-2020-9484](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-9484), [CVE-2021-25329](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-25329), [CVE-2021-30640](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-30640),  [CVE-2022-34305](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-34305), [CVE-2021-24122](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-24122), [CVE-2021-33037](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-33037), [CVE-2019-17569](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2019-17569), [CVE-2020-1935](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-1935),  [CVE-2020-13943](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13943), [CVE-2021-43980](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-43980)

## Mitigation Plan

This DevOps pipeline will be transformed into a DevSecOps pipeline as part of the mitigation plan. Due to the early stage of the project, it would be beneficial for everyone on the development team to adopt a security mindset. Everything will be built with security in mind, and we can move forward correctly. As a result, it reduces the chances of input validation being a cause for concern for NIST before the project becomes operational. When looking at the vulnerabilities, we must eliminate any false positives by verifying if the CPE values are inaccurate. False positives can be suppressed if the CPE values are incorrect. We must evaluate each entry to determine whether any of the identified CVE entries can be exploited in this environment. To be safe, I suggest upgrading the library first. Changing configuration options or upgrading the library can fix some CVE entries. In addition, these tools have some exploits that arise because of how they are designed. Additionally, code injection risks, incorrect error handling issues, and insufficient input validation contribute to many exploits. When these problems are resolved, the program can be used by the client and their clients with confidence.

Mitigation measures are also required for the dependencies identified above. Below is a list of those measures:

* **Dependency:** [bcprov-jdk15on-1.46.jar](file:///C:\Users\chris\Desktop\SNHU%20Classes\Tenth%20Term\CS%20305\Week%20Three\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l1_991c96a4e31e6c19e2b9136c8955bd423f2dc4c7)
* Mitigating this threat will involve using version 1.56 or later because the later versions check the critical parameters on agreement calculation. Keeping keys secure is what makes cryptography worth it in the first place. Therefore, it is of vital importance to have those keys appropriately calculated and stored. It is possible to inject extra elements in the sequence making up the signature and still have it validated, which in some cases may allow the introduction of 'invisible' data into a signed structure.
* **Dependency:** [hibernate-validator-6.0.18.Final.jar](file:///C:\Users\chris\Desktop\SNHU%20Classes\Tenth%20Term\CS%20305\Week%20Three\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l3_7fd00bcd87e14b6ba66279282ef15efa30dd2492)

Mitigating this threat will involve passing user input as an expression variable by unwrapping the context to HibernateConstraintValidatorContext or update to the current version 6.0.20.

* **Dependency:** [jackson-databind-2.10.2.jar](file:///C:\Users\chris\Desktop\SNHU%20Classes\Tenth%20Term\CS%20305\Week%20Three\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l5_0528de95f198afafbcfb0c09d2e43b6e0ea663ec)
* Mitigating this threat will involve configuring the XML parser and validator to disable eternal entity expansion, meaning an attacker can no longer submit altered DTD files within the XML documents because the parser will catch it and ignore the malicious code if done correctly. An alternative method would be updating to the current version, 2.13.4.
* **Dependency:** [log4j-api-2.12.1.jar](file:///C:\Users\chris\Desktop\SNHU%20Classes\Tenth%20Term\CS%20305\Week%20Three\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l10_a55e6d987f50a515c9260b0451b4fa217dc539cb)

Mitigating this threat involves updating to the current Log4j2 version 2.12.4.

* **Dependency:** [logback-core-1.2.3.jar](file:///C:\Users\chris\Desktop\SNHU%20Classes\Tenth%20Term\CS%20305\Week%20Three\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l12_864344400c3d4d92dfeb0a305dc87d953677c03c)

Mitigating this threat involves updating to the current Logback version 1.2.8 and slf4j version 1.7.32

* **Dependency:** [snakeyaml-1.25.jar](file:///C:\Users\chris\Desktop\SNHU%20Classes\Tenth%20Term\CS%20305\Week%20Three\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l14_8b6e01ef661d8378ae6dd7b511a7f2a33fae1421)

Mitigating this threat involves updating to the current snakeyaml version 1.32.

* **Dependency:** [spring-boot-2.2.4.RELEASE.jar](file:///C:\Users\chris\Desktop\SNHU%20Classes\Tenth%20Term\CS%20305\Week%20Three\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l15_225a4fd31156c254e3bb92adb42ee8c6de812714)

Mitigating this threat can involve setting the java.io.tmpdir system environment variable to a directory that is exclusively owned by the executing user to fix this vulnerability for all operating systems or updating to version v2.2.11 RELEASE or later.

* **Dependency:** [spring-boot-starter-web-2.2.4.RELEASE.jar](file:///C:\Users\chris\Desktop\SNHU%20Classes\Tenth%20Term\CS%20305\Week%20Three\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l16_ec75d01d212b5229c16d872fb127744c0ed46ed8)

Mitigating this threat involves either setting the java.io.tmpdir system environment variable to a directory that is exclusively owned by the executing user will fix this vulnerability for all operating systems or updating to the current Spring-boot v2.2.11.RELEASE or later same as the above dependency.

* **Dependency:** [spring-core-5.2.3.RELEASE.jar](file:///C:\Users\chris\Desktop\SNHU%20Classes\Tenth%20Term\CS%20305\Week%20Three\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l17_3734223040040e8c3fecd5faa3ae8a1ed6da146b)

Mitigating this threat involves upgrading to the spring framework 5.2.22.

* **Dependency:** [spring-webmvc-5.2.3.RELEASE.jar](file:///C:\Users\chris\Desktop\SNHU%20Classes\Tenth%20Term\CS%20305\Week%20Three\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l19_745a62502023d2496b565b7fe102bb1ee229d6b7)

Mitigating this threat would be taken care of by upgrading the spring framework from the dependency above.

* **Dependency:** [tomcat-embed-core-9.0.30.jar](file:///C:\Users\chris\Desktop\SNHU%20Classes\Tenth%20Term\CS%20305\Week%20Three\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l20_ad32909314fe2ba02cec036434c0addd19bcc580)
* Mitigating this threat involves either ensuring rejectIllegalHeader is set to true or upgrading Apache Tomcat to 9.0.68 or later. My recommendation is to update Apache Tomcat. Always using updated software helps to ensure the latest patches to known vulnerabilities are applied.
* **Dependency:** [tomcat-embed-websocket-9.0.30.jar](file:///C:\Users\chris\Desktop\SNHU%20Classes\Tenth%20Term\CS%20305\Week%20Three\CS%20305%20Project%20One%20Code%20Base\rest-service\target\dependency-check-report.html#l22_33157f6bc5bfd03380ebb5ac476db0600a04168d)

Mitigating this threat would be taken care of by upgrading the Apache Tomcat version from the dependency above.

**REFERENCES**

National Vulnerability Database. (2019). NIST. <https://nvd.nist.gov/vuln/search>

Probasco, L. (2017, April 25). *Encryption Requirements for Banks & Financial Services*. Townsend Security. Retrieved from https://info.townsendsecurity.com/encryption-requirements-for-banks-financial-services#:~:text=There%20are%20strong%20data%20security%20requirements%20for%20banking,SOX%2C%20and%20state%20privacy%20laws%20can%20also%20apply.

*Search CVE List*. (2019). CVE. <https://cve.mitre.org/cve/search_cve_list.html>