****

# CS 305 Project One

**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** | **May 21, 2021** | **Madeline Neel** |  |

## Client



## Instructions

Deliver this completed vulnerability assessment report, identifying your findings of security vulnerabilities and articulating recommendations for next steps to remedy the issues you have found.

Respond to the five steps outlined below and include your findings. Replace the bracketed text on all pages with your own words. If you choose to include images or supporting materials, be sure to insert them throughout.

## Developer

Madeline Neel

## 1. Interpreting Client Needs

* *What is the value of secure communications to the company?*

Secure communications allows a sense of trust to form between a company and their client. Companies intake a lot of personal and private information from their clients, so it is imperative that there is secure communications to protect their information.

* *Are there any international transactions that the company produces?*

I’m unsure if there are any international transactions that the company produces as it was not stated, but from assumption – I assume that there would be. Their customers may travel, which may result in them accessing the data from an international setting. Otherwise, it may be beneficial to count on international transactions incase the company decides to broaden their client-base.

* *Are there governmental restrictions about secure communications to consider?*

Each state has their own restrictions and laws on secure communications and data security. Since cyberattacks and threats have gained traction among hackers within the last several years, many laws have only recently been enacted (*Data Security Laws: State Government*, 2020).

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

External threats will constantly be present. They’re different from internal threats in that they are threats that occur from a source outside of the company. Threats can range from spam phone calls, emails, text messages to malicious software and programs. Oftentimes, the spam emails and text messages contain links to websites that download a malicious program onto a computer in an attempt to gain personal information and if an employee accidentally or purposefully clicks on it, the company is at risk for a security breach. Some of the different types of threats include Malware, Spyware, Trojan horses, phishing, DDoS (distributed denial of service) and viruses. Cybercriminals are advancing and improving just as well as network security is advancing and improving.

* *What are the “modernization” requirements that must be considered, such as the role of open source libraries and evolving web application technologies?*

Our society is constantly changing and evolving. Modernizing secure communications and security within companies is essential. Specifically, the global pandemic that has taken place over the past year; many companies have resorted to remote work, which has required companies to update and increase their security measures that they have in place. Open source libraries are hunting grounds for cybercriminals to attack and put companies at risk. The codes that are used in open source can be breached by cybercriminals in some instances. Once a cybercriminal gains access to an open source library, they have easier access to other personal information.

## 2. Areas of Security

* The Vulnerability Assessment Process Flow, also known as VAPF, will be used to determine the critical vulnerabilities that may occur in that system. Determining these vulnerabilities will allow you to distinguish what needs to be prioritized (high-risk vulnerabilities) and what will possibly need to be fixed on a scheduled basis.
* For this system, the areas of security that are relevant are APIs, code quality, input validation, and cryptography. The Spring Framework web application supports API usage. Code quality is important in every, coded, system. Input validation and cryptography go hand in hand in this section as the data transferred is confidential as well as the need to ensure the application is giving the correct information to the correct person.

## 3. Manual Review

One of the vulnerabilities found by manually inspecting the code is the Spring Core version used. In the pom.xml file, it shows that that the version used is 2.2.4 but when looking on spring.io – it shows that the current updated version is 5.3.7. When frameworks go through updates, they fix previous errors and add features to help with security, etc. making it essential to have the most recent framework. As shown in the customer class, they can request to show information. It is important to verify more than just the account number to show someone’s’ personal data.

## 

## 4. Static Testing

Information taken from dependency check report.

* Bcprov-jdk15on-1.46.jar
  + Bouncy Castle Crypto package is a Java implementation of cryptographic algorithms
    - Bouncy Castle 1.0.3 and earlier, an attacker can recover the private key from a vulnerable application.
    - Bouncy Castle 1.51 and below does not validate within the elliptic curve, allowing remove attackers to obtain private keys.
    - Bouncy Castle 1.55 and earlier: does not validate ASN, data channel on CPU can be monitored and leak information, vulnerable to timing attack, key pair generator generates a weak private key, use of ECB mode (unsafe), possibility of padding oracle attack, and DH public key in not fully validated.
    - Bouncy Castle 1.61 and before allows attackers to obtain sensitive information because of Observable Differences in Behavior to Error Inputs.
* Log4j-api-2.12.1.jar
  + The Apache Log4j API
    - Improper validation of certificate with host could allow SMTPS connection to be intercepted by a man-in-the-middle attack. This could leak log messages sent through that appender.
* Snakeyaml-1.25.jar: False Positive due to us using 1.25 update, not 1.18
  + YAML 1.1 parser and emitter for Java
    - Alias feature in 1.18 allows entity expansion during load operation.
* Jackson-databind-2.10.2.jar
  + General data-binding functionality for Jackson: works on core streaming API
    - Threat of data integrity by not securing an entity expansion properly.
* Tomcat-embed-core-9.0.30.jar
  + Core Tomcat implementation.
    - 9.0.30 contains: a regression that incorrectly processed transfer-encoding headers, triggers high CPU usage, possibility of denial of service when a sufficient number of requests were made, invalid payload lengths could trigger an infinite loop, possibility of users seeing responses for unexpected resources, information could leak between requests, improper input validation, incorrect default permissions, deserialization of untrusted data, susceptible to JSP source code disclosure, and duplicate request headers.
* Hibernate-validator-6.0.18.Final.jar
  + Hibernate's Bean Validation (JSR-380) reference implementation.
    - Enables invalid expressions to be evaluates as if they were valid – improper input validation.
* Spring-core-5.2.3.RELEASE.jar
  + Spring Core
    - Does not support protections against RFD attacks.

## 

## 5. Mitigation Plan

* For this assignment, I used the dependency-check version 6.1.6. There were 7 dependencies and 34 vulnerabilities found. Two of the dependencies had above one CVE (Common Vulnerabilities and Exposures) while the remaining five only had one.
* Many of the vulnerabilities under the Bouncy Castle Crypto dependency have already been modified since it was last analyzed by the NVD and is awaiting reanalysis. The vulnerabilities included in this situation are: CVE-2013-1624, 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, and CVE-2018-1000613.
  + For the remaining vulnerability and according to the bouncy castle website, CVE-2018-5382 has been fixed. Although, the current update is on 2.68.3 which has many additional functionality, security, and defects fixed from the current used update (1.46).
* Similarly to the Bouncy Castle, many of the singular vulnerabilities have already been modified and is waiting for reanalysis from the NVD. These include: CVE-2020-9488 under the Apache Log4j API dependency, CVE-2017-18640 under the YAML 1.1 parser and emitter for Java dependency, CVE-2020-25649 under the Jackson data-bind dependency, and CVE-2020-5421 under the Spring Core dependency.
  + The remaining vulnerability, CVE-2020-10693 under the Hibernate's Bean Validation dependency, can be addressed by using a fix pack containing APAR PH29942 for each product.
* For Core Tomcat Implementation, the vulnerabilities that have already been modified and are waiting for reanalysis include: CVE-2019-17569, CVE-2020-13934, CVE-2020-13935, CVE-2020-13943, CVE-2020-17527, CVE-2020-1938, and CVE-2020-9484.
  + CVE-2020-11996, CVE-2020-1935, CVE-2020-8022, CVE-2021-24122, CVE-2021-25122, and CVE-2021-25329 were not included in the NVD reanalysis but after further examination from third party vendors, they have been reexamined and fixed.

Reference:

Data Security Laws: State Government. (2020, February 14). Retrieved from

https://www.ncsl.org/research/telecommunications-and-information-technology/data-security-laws-state-government.aspx