# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **9/19/24** | **Joseph Caron** |  |

## Client



## Instructions

Submit this completed vulnerability assessment report. Replace the bracketed text with the relevant information. In this report, identify your security vulnerability findings and recommend 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 include images or supporting materials. If you include them, make certain to insert them in 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

Joseph Caron

**1. Interpreting Client Needs**

Artemis Financial is a financial consulting company. They develop individual financial plans for their customers, which include savings, retirement, investments and insurance planning. They have come to Global Rain to modernize their operations with custom made software, and due to the sensitive nature of their business, security must be at least part of the main focus. As a company that deals mainly in financial planning, much of their communications will involve private information such as bank account information, customers’ names, addresses and SSNs, and tax information. The information we have been given does not explicitly state if the company does work internationally, though as it deals with financial planning, I think it is safe to assume that they do have international business. There are no governmental restrictions about secure communications that have any bearing on this, except for ensuring that the encryption used to secure the communications meets international communication standards.

In terms of external threats, there is always concern of identity theft and phishing attacks when working with personal identity, financial and insurance information. Of course communications are the most likely target, as information in transit by necessity has less overall security than a locked down database. Because this software is all about modernizing the operation of Artemis Financial, there are certain modernization requirements to be utilized. Open-source libraries are an absolute must in the building of the software, and these must be kept up to date as software constantly evolves. Using secure coding practices and DevSecOps during the entire SDLC is critical to ensuring the security of the software.

**2. Areas of Security**

Areas of security that need to be focused on are:

* Input validation: it is crucial to validate user input to ensure there aren’t failures or SQL injection.
* APIs: As interactions will be both internal and external, as well as potentially communicating with 3rd party software, a secure API is necessary.
* Cryptography: There will be a lot of personal, confidential information in motion in this software, so secure encryption is required.
* Code Error: Crucial and should be used in tandem with input validation and the API to prevent unauthorized access.
* Code Quality: This is crucial to ensure that no unauthorized privilege access is granted, and that there is no accidental data exposure.

**3. Manual Review**

* The GreetingController.java file does not have any kind of input validation built in.
* There is no working API found in the code.
* Data encryption still needs to be implemented
* There are no error handling methods in place in the DocData.java file apart from try and catch.
* Business names are included as request parameters in the CRUDController.java file.
* Program uses http rather than https.
* There is a lack of authentication in the code.

**4. Static Testing**

|  |  |  |  |
| --- | --- | --- | --- |
| **Dependency** | **Vulnerability IDs** | **Description** | **Solution** |
| bcprov-jdk15on-1.46.jar | 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:  bouncy\_castle\_for\_java:  1.46:\*:\*:\*:\*:\*:\*:\*  cpe:2.3:a:bouncycastle:  legion-of-the-bouncy-castle-java  -crytography-api:1.46:\*:\*:\*:\*:\*:\*:\*  cpe:2.3:a:bouncycastle:  the\_bouncy\_castle\_crypto\_  package\_for\_java:1.46:\*:\*:\*:\*:\*:\*:\* | Bouncy Castle for Java before 1.73 contains a potential Denial of Service (DoS) issue within the Bouncy Castle org.bouncycastle.openssl.PEMParser class. This class parses OpenSSL PEM encoded streams containing X.509 certificates, PKCS8 encoded keys, and PKCS7 objects. Parsing a file that has crafted ASN.1 data through the PEMParser causes an OutOfMemoryError, which can enable a denial of service attack. | Update to version 1.73 |
| hibernate-validator-6.0.18.Final.jar | cpe:2.3:a:redhat:hibernate  \_validator:6.0.18:\*:\*:\*:\*:\*:\*:\* | 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 latest version |
| jackson-databind-2.10.2.jar | cpe:2.3:a:fasterxml:jackson-databind:2.10.2:\*:\*:\*:\*:\*:\*:\*  cpe:2.3:a:fasterxml:jackson-modules-java8:2.10.2:\*:\*:\*:\*:\*:\*:\* | jackson-databind through 2.15.2 allows attackers to cause a denial of service or other unspecified impact via a crafted object that uses cyclic dependencies. | Update to latest version |
| log4j-api-2.12.1.jar | cpe:2.3:a:apache  :log4j:2.12.1:\*:\*:\*:\*:\*:\*:\* | Apache Log4j2 versions 2.0-beta7 through 2.17.0 (excluding security fix releases 2.3.2 and 2.12.4) are vulnerable to a remote code execution (RCE) attack when a configuration uses a JDBC Appender with a JNDI LDAP data source URI when an attacker has control of the target LDAP server. This issue is fixed by limiting JNDI data source names to the java protocol in Log4j2 versions 2.17.1, 2.12.4, and 2.3.2. | Limit JNDI data source names to the java protocol in Log4j2 versions 2.17.1, 2.12.4, and 2.3.2. |
| logback-core-1.2.3.jar | cpe:2.3:a:qos:logback  :1.2.3:\*:\*:\*:\*:\*:\*:\* | A serialization vulnerability in logback receiver component part of logback version 1.4.11 allows an attacker to mount a Denial-Of-Service attack by sending poisoned data. | Update to version 1.5.8 |
| snakeyaml-1.25.jar | cpe:2.3:a:snakeyaml\_project  :snakeyaml:1.25:\*:\*:\*:\*:\*:\*:\* | SnakeYaml's Constructor() class does not restrict types which can be instantiated during deserialization. Deserializing yaml content provided by an attacker can lead to remote code execution. We recommend using SnakeYaml's SafeConsturctor when parsing untrusted content to restrict deserialization. We recommend upgrading to version 2.0 and beyond. | Upgrade to latest version |
| spring-boot-2.2.4.RELEASE.jar | cpe:2.3:a:vmware:spring  \_boot:2.2.4:release:\*:\*:\*:\*:\*:\* | In Spring Boot versions 3.0.0 - 3.0.6, 2.7.0 - 2.7.11, 2.6.0 - 2.6.14, 2.5.0 - 2.5.14 and older unsupported versions, there is potential for a denial-of-service (DoS) attack if Spring MVC is used together with a reverse proxy cache. | Update to version 3.3.4 |
| spring-core-5.2.3.RELEASE.jar | cpe:2.3:a:pivotal\_software  :spring\_framework:5.2.3:  release:\*:\*:\*:\*:\*:\*  cpe:2.3:a:springsource:  spring\_framework:5.2.3:  release:\*:\*:\*:\*:\*:\*  cpe:2.3:a:vmware:  spring\_framework:5.2.3:  release:\*:\*:\*:\*:\*:\* | In spring framework versions prior to 5.2.24 release+ ,5.3.27+ and 6.0.8+ , it is possible for a user to provide a specially crafted SpEL expression that may cause a denial-of-service (DoS) condition. | Update to version 6.1.13 |
| tomcat-embed-core-9.0.30.jar | cpe:2.3:a:apache:tomcat  :9.0.30:\*:\*:\*:\*:\*:\*:\*  cpe:2.3:a:apache\_tomcat  :apache\_tomcat:  9.0.30:\*:\*:\*:\*:\*:\*:\* | Generation of Error Message Containing Sensitive Information vulnerability in Apache Tomcat.This issue affects Apache Tomcat: from 8.5.7 through 8.5.63, from 9.0.0-M11 through 9.0.43. Users are recommended to upgrade to version 8.5.64 onwards or 9.0.44 onwards, which contain a fix for the issue. | Update to version 9.0.95 |

**5. Mitigation Plan**

* Update all dependencies shown above in section 4
* Build input validation into the greeting controller
* Build in data encryption
* Switch to HTTPS rather than HTTP
* Build in user and privilege authentication
* Implement API