# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **06-07-24** | **J. Sullo** | **Analyzing dependencies and recommending solutions** |

## 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 Sullo

**1. Interpreting Client Needs**

Determine your client’s needs and potential threats and attacks associated with the company’s application and software security requirements. Consider the following questions regarding how companies protect against external threats based on the scenario information:

* What is the value of secure communications to the company?
* Are there any international transactions that the company produces?
* Are there governmental restrictions on secure communications to consider?
* What external threats might be present now and in the immediate future?
* What modernization requirements must be considered, such as the role of open-source libraries and evolving web application technologies?

Other than the accuracy of their transactions, secure communications are the most important thing a financial institution such as Artemis must focus on. Their communications contain many different types of sensitive personal information such as social security numbers, bank account information, and medical records and it is of the upmost importance that these remain secure from unauthorized viewers not only for the security of clients but to maintain Artemis’s good reputation. Such personal information is often targeted in cyberattacks, being sold to others or used directly to steal identities and rack up huge expenses which the affected clients could be responsible for. There are also state and federal regulations, such as the Federal Trade Commission Act, that must be adhered to in order to avoid costly fines or, at worst, closure of the business. Modernization of the applications used for communication security will be paramount to the ongoing protection of Artemis and its clients, such as with open-source libraries and the vast number of users who have access to the base code thus increasing the chance that security threats are located and fixed. Keeping protocols and dependencies up to date is critical to keep security strong against the ever-evolving threat of cyber criminals and hackers.

**2. Areas of Security**

Refer to the vulnerability assessment process flow diagram. Identify which areas of security apply to Artemis Financial’s software application. Justify your reasoning for why each area is relevant to the software application.

There are several areas of security that apply to the software application that Artemis Financial has asked us to analyze. First is input validation; as stated, securing client information is crucial to the success of Artemis. Since the application allows for string input, strong validation is required to ensure that errors are handled properly, access is not granted to unauthorized parties, and things such as SQL injections that could affect many more clients are not allowed to happen. APIs are also very important as much of the use for this application will be done by the end user. The API will dictate how users are able to interact with the program, so special care needs to be taken to ensure that only acceptable methods of accessing data can be used. By extension of these first two, cryptology will be a necessary area to focus on for any international business conducted. Regulations in both North America and the country of the client must be adhered to, and any security measures up to and including encryption of sensitive data much comply. Code errors are important for any system and Artemis’s is no different. Proper error handling must work together with the API and input verification to ensure that any errors that do occur are quickly recognized and resolved, otherwise the application might not function as intended and grant access to sensitive information or program dependencies which can then be exploited for further data loss. Finally, as with code errors, code quality is something that all applications need to consider. Not only will efficient, well written code help the overall user experience and make the application run smoothly, it will also ensure that users are only permitted access to the data their level applies to, i.e. a client not gaining access to manager, or even developer, levels of data within the program.

**3. Manual Review**

Continue working through the vulnerability assessment process flow diagram. Identify all vulnerabilities in the code base by manually inspecting the code.

* The Greeting Controller lacks any sort of input validation
* DocData has a simple try/catch block, but no real error handling
* The program uses HTML to access data rather than the POST method
* HTMLS is not utilized
* There does not seem to be any cryptology methods employed anywhere in the program
* Business names are simple request parameters in the CRUDController class
* Multiple parameters in the POM.xml are out of date and require the latest version

**4. Static Testing**

Run a dependency check on Artemis Financial’s software application to identify all security vulnerabilities in the code. Record the output from the dependency-check report. Include the following items:

* The names or vulnerability codes of the known vulnerabilities
* A brief description and recommended solutions provided by the dependency-check report
* Any attribution that documents how this vulnerability has been identified or documented previously

|  |  |  |  |
| --- | --- | --- | --- |
| Dependency | Vulnerabilities | 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:\*:\*:\*:\*:\*:\*:\* | Earlier versions of Bouncy Castle bring issues such as errors with validation, poor key generation security and performance, allowance of unsafe user modes, and vulnerabilities to attacks such as padding oracle and timing | Update Bouncy Castle to the latest version |
| hibernate-validator-6.0.18.Final.jar | cpe:2.3:a:redhat:hibernate\_validator:6.0.18:\*:\*:\*:\*:\*:\*:\* | A bug can cause invalid EL expressions to be seen as valid, allowing users to bypass input sanitation | Update Hibernate Validator to the 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:\*:\*:\*:\*:\*:\*:\* | Older versions leave vulnerabilities to XXE attacks, allow for denial of service, and cause resource exhaustion | Update FasterXML to the latest version |
| log4j-api-2.12.1.jar | cpe:2.3:a:apache:log4j:2.12.1:\*:\*:\*:\*:\*:\*:\* | There is an improper validation of certificate issue, allowing connections to be intercepted by man in the middle attacks | Update Apache Log4j to the latest version |
| logback-core-1.2.3.jar | cpe:2.3:a:qos:logback:1.2.3:\*:\*:\*:\*:\*:\*:\* | Serialization vulnerabilities allow for denial of service attacks | Update logback to the latest version |
| snakeyaml-1.25.jar | cpe:2.3:a:snakeyaml\_project:snakeyaml:1.25:\*:\*:\*:\*:\*:\*:\* | The Constructor() class has no restriction on type instantiation, leading to the possibility of remote code execution. It also is vulnerable to denial of service attacks. | Update SnakeYami to current version |
| spring-boot-2.2.4.RELEASE.jar | cpe:2.3:a:vmware:spring\_boot:2.2.4:release:\*:\*:\*:\*:\*:\* | Older versions are vulnerable to security bypasses, temporary directory hijacking, and denial of service attacks | Update Spring Boot to latest version |
| spring-boot-starter-web-2.2.4.RELEASE.jar | cpe:2.3:a:vmware:spring\_boot:2.2.4:release:\*:\*:\*:\*:\*:\*  cpe:2.3:a:web\_project:web:2.2.4:release:\*:\*:\*:\*:\*:\* | The same vulnerabilities as the previous dependency | Update Spring Boot to latest version |
| 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:\*:\*:\*:\*:\*:\* | There are multiple vulnerabilities in the Spring Framework to create denial of service conditions such as specifically crafted SpEL expressions. It is also vulnerable to remote code execution and insertion of additional log entries through malicious input. There is also the possibility that security features might be bypassed | Update Spring Framework to the latest version |
| spring-web-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:\*:\*:\*:\*:\*:\*  cpe:2.3:a:web\_project:web:5.2.3:release:\*:\*:\*:\*:\*:\* | This dependency allows for multiple ways that unauthorized users can access sensitive information via open redirect or SSRF attack via URL. It also shares the same vulnerabilities as the previous dependency | Update Spring Framework to latest version |
| spring-webmvc-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:\*:\*:\*:\*:\*:\*  cpe:2.3:a:web\_project:web:5.2.3:release:\*:\*:\*:\*:\*:\* | This shares the same vulnerabilities as the spring-core dependency | Update Spring Framework to latest version |
| 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:\*:\*:\*:\*:\*:\*:\* | Apache Tomcat allows for multiple ways of inducing a denial of service attack, resource loss, poor validation, duplicate request headers, and data smuggling | Update Apache Tomcat to the latest version |
| tomcat-embed-websocket-9.0.30.jar | cpe:2.3:a:apache:tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*  cpe:2.3:a:apache\_tomcat:apache\_tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\* | This dependency shares all of the same vulnerabilities as the previous tomcat dependency does | Update Apache Tomcat to the latest version |

**5. Mitigation Plan**

Interpret the results from the manual review and static testing report. Then identify the steps to mitigate the identified security vulnerabilities for Artemis Financial’s software application.

All of the various issues plaguing Artemis Financial’s software seem to stem from old versions of dependencies that have been revealed to have security risks. Bringing them all to the latest release version is paramount to solving them, and a new class should be added in order to automatically compare the software version to its current release. This will ensure that all security features are as up to date as possible.