# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **1-23-2025** | **Jonathan abramavage** |  |

## 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

Jonathan Abramavage

**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?

The value of sure commination is to protect their client files as they will have access to sensitive files. If these files got accessed from the outside their clients could have their money drained or their identities stolen. As they work with businesses and government agencies from all over, they would have international clients. Yes, there are a lot governmental restrictions to consider. Since they might have government agencies from all over, they will have to interact with all kinds of restrictions on their communications. External threats can come from people wanting to steal money or the identities of the people that have savings accounts. They might want to hack into the governmental retirements accounts or the insurance they use in the hopes they can access other systems. They need use modern encryptions methods to keep the data secure. They also need to use modern polices to keep people from getting access to the physical’s items like the servers and the data cables.

**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.

Code quality, Having quality code is important as it limits vulnerabilities it also makes sure the program works without issues.

Code error, Having no errors in the code is important to keeping the private data secure an example is when to typing your ssn it immediately turns to \* so no one can see what you have typed.

Client/server, The server has a bunch of vulnerability’s as this is where the info is saved so we have to make sure no one can get through the client to steal data from the server but we also have to protect the physical server so they can take a flash drive and get the data.

Input validation, is important as the program needs to determine what the input needs to be an example would be putting numbers in the name field or trying to upload a hyper link in a field that doesn’t need it.

**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 code has a problem in the greeting controller file as it takes inputs and I could nit find where they are checked for validity. There seems to be no api either. That’s a problem because we don’t know how outside things interact with the code. It looked like the quality of the code was executed well as I didn’t see anything wrong in that area. I also could not find any form of data security to encryption any data that was entered.

**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 | Vulnerability | Description | Solution |
| bcprov-jdk15on-1.46.jar | Bouncy Castle in Android before 5.1.1 LMY49F and 6.0 before 2016-01-01 allows attackers to obtain sensitive information via a crafted application, aka internal bug 24106146. | A bug in the application will allows for data to be stolen. | Update the applications |
| spring-boot-2.2.4.RELEASE.jar | In Spring Boot versions 3.0.0 - 3.0.5, 2.7.0 - 2.7.10, and older unsupported versions, an application that is deployed to Cloud Foundry could be susceptible to a security bypass. Users of affected versions should apply the following mitigation: 3.0.x users should upgrade to 3.0.6+. 2.7.x users should upgrade to 2.7.11+. Users of older, unsupported versions should upgrade to 3.0.6+ or 2.7.11+ | Its possible to bypass the security in the older versions | Update to the newest versions |
| logback-classic-1.2.3.jar | In logback version 1.2.7 and prior versions, an attacker with the required privileges to edit configurations files could craft a malicious configuration allowing to execute arbitrary code loaded from LDAP servers | When the user has certain permissions, they can change the program and add malicious configuration | Back round checks on the users that need access and limit access to only people that need it. |
| logback-core-1.2.3.jar | 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 | The person sends corrupted data and take over the application and prevent the actual user from accessing their data | Limit people that can send data in and upgrade to the newest version |
| log4j-api-2.12.1.jar | Improper validation of certificate with host mismatch in Apache Log4j SMTP appender. This could allow an SMTPS connection to be intercepted by a man-in-the-middle attack which could leak any log messages sent through that appender. Fixed in Apache Log4j 2.12.3 and 2.13.1 | The user can insert corruption in the middle of the email sending or receiving | Update to the newest one and make the validation of data is good. Make sure you don’t send important stuff in emails. |

**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.

I would start with upgrading everything so we can fix the vulnerabilities. I would also check the data validation to make sure its secure after the updates