# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **5/18/2024** | **Zach Chilgren** |  |

## 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

Zach Chilgren

**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?

Having secure communications can help prevent possible attacks as well as keeping sensitive information private and away from people it should not reach. Keeping information private is also a big reason some would choose to either use or not use a company and if internal information were to be able to get obtained easily it could deter people from using the companies’ services.

It does not appear that there are any internal transactions that take place within the company as of now but would be best to plan for as it is possible that they will happen eventually, and this information should be kept just as secure as any other bit of information within the company.

There are no government laws on secure communication, but a lot of individual states have laws and regulations around secure information and data breaches as well as many other foreign countries have their own laws and regulations based on data security and would need to make sure they are adhering to all rules if doing business in those areas.

For a company that deals with sensitive information there are many groups that would wish to access that information and either sell it ransom the access to that data from the company and having a system that can protect from that would be the best to build to protect from.

[Modernization]

**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.

* CRUDController.java takes in input but is not validating that input in any form which could be a possible area for security failure.
* Any information that does not need to be seen by the end user but the system does need in some form should be made sure to keep private as to make sure no unnecessary information gets released by the program.
* Cryptography in some form will be necessary if business is wished to be done outside of the US as some other countries have rules and regulations around data security unlike the US
* The use of an API for this application will be necessary as this application will not be ran entirely off of the end users devices so information will need to be accessible from some form of a data base on request and this needs to be kept secure to prevent anyone from accessing others information.

**3. Manual Review**

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

1. CRUDController does not validate any of its input which could eventually cause issues
2. There is not presence of any api which means the data is unsecured and can be easily accessed by unintended people.
3. GreetingController takes in an input but does not valid or show that It correctly took an input in any way making it hard to deduce if the system worked properly
4. myDateTime is set in seconds, minutes, hours which is just an unorthodox order to check, input or show the time for someone to read or give
5. The deposit for customer.java takes in an integer only allowing from even and round deposits to be made instead of a float which would allow for decimals and cents to be added in a deposit
6. The account\_balance variable should also be kept private until it is requested

**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
* bcprov-jdk15on-1.46.jar ( pkg : maven / org.bouncycastle/bcprov-jdk15on@1.46 , cpe:2.3: a:bouncycastle:legion-of-the-bouncy-castle-java-crytography- api:1.46 : \*: \*: \*: \*: \*: \*: \* ) : CVE-2013-1624 , CVE-2015-6644 , CVE-2015-7940 , 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 , CVE-2018-1000613 , CVE -2018-5382 , CVE-2020 – 26939
  + Cryptography algorithms can have issues when it comes to securing sensitive data and this version does not validate one of the public keys which could be used to gain access to sensitive information
* log4j-api-2.12.1.jar ( pkg: maven/org.apache.logging.log4j/log4j-api@2.12.1 , cpe:2.3: a : apache : log4j:2.12.1:\* : \*: \*: \*: \*: \*: \* ) : CVE-2020-9488
  + The check that determines if the host as a safe connection does not function properly and can connect people to a not safe connection putting their data at risk
* snakeyaml-1.25. jar ( pkg: maven/org.yaml/snakeyaml@1.25 , cpe:2.3:a:snakeyaml\_project:snakeyaml:1.25 : \*: \*: \*: \*: \*: \*: \* ) : CVE-2017-18640
  + This could cause the information that is being taken in by the system to not be properly parsed leading to the eventuality of denying data causing the system to no longer take in information and shut down.
* jackson-databind-2.10.2.jar ( pkg : maven/com.fasterxml.jackson.core/jackson - databind@2.10.2, cpe:2.3: a:fasterxml:jackson-databind:2.10.2: \*: \*: \*: \*: \*: \*: \* ) : CVE - 2020-25649
  + This error could allow someone to add an XML file to the program which could potentially be used to gain access to secure information
* tomcat-embed-core-9.0.30.jar (pkg:maven/org.apache.tomcat.embed/tomcat-embed-core@9.0.30, cpe:2.3:a:apache:tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*,cpe:2.3:a:apache\_software\_foundation:tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*,cpe:2.3:a:apache\_tomcat:apache\_tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*) : CVE-2019-17569,CVE-2020-11996, CVE-2020-13934, CVE-2020-13935, CVE-2020-13943, CVE-2020-6 17527, CVE-2020-1935, CVE-2020-1938, CVE-2020-8022, CVE-2020-9484, CVE-2021-24122, CVE-2021-25122, CVE-2021-25329, CVE-2021-30640, CVE-2021-33037
  + The possible risk of this is that people could gain access to information without the need for proper security clearance meaning that if there is information that is meant to be access by a certain party but no others that information could potentially be accessed by the unintended party allowing for easy access to secure and sensitive information.

**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 Financials’ software application.

* bcprov-jdk15on-1.46.jar
  + an updated version would help fix the issue with validation of secure keys when accessing information
* log4j-api-2.12.1.jar
  + needs to be able to ensure all connections are safe before allowing access to the application
* snakeyaml-1.25. jar
  + information needs to be properly parsed to help prevent an overflow of information
* jackson-databind-2.10.2.jar
  + by preventing people from adding an XML file this vulnerability could be completely avoided.
* tomcat-embed-core-9.0.30.jar
  + This would help secure information and make certain information only accessible by the correct people keeping personal data and information secure from other people.