# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **9/21/2024** | **Jacob Virgilio** | **First Iteration** |

## 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

Jacob Virgilio

**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?

Artemis Financial is a financial firm, which means that security is naturally paramount. Since this firm handles sensitive information such as banking records and account numbers, security is their number one priority.

* Are there any international transactions that the company produces?

Artemis Financial has the ability to secure and produce both domestic and international transactions, which further strengthens the need for enhanced security.

* Are there governmental restrictions on secure communications to consider?

Current governmental restrictions on secure communications are not specific to a country, aside from the possibility of leaking sensitive governmental information. This further solidifies the need for improved overall security.

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

Since Artemis is handling financial information, the possibility of hacking, data hostage scenarios, and overall information leaks is more predominant than for a company that does not handle financial information. These issues will almost always be present, requiring ever evolving countermeasures to be in place.

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

Artemis must ensure that all libraries remain up to date, and any found compromises must be fixed as quickly as possible, to prevent sensitive data leaking.

**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.

API - Since this application runs externally on users web browsers, a hearty API will be a requirement. This ensures that only specific users can see data that is relative to them.

Input Validation - Input validation is a necessary step to prevent arbitrary code execution. This includes limiting input strings, as well as keeping a lock on attempts to crack passwords.

Code Quality - Code must run smoothly, with as little risk for errors as possible. Code must be commented, and kept secure through certain types of cryptokeys.

Cryptography - In the same vein, data transfers must be untraceable, with a frequently updated data code so hackers cannot zero in on a certain area to attack.

**3. Manual Review**

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

* CRUDController class is sent business names as request parameters
* DocData has built-in connection parameters for its database, which can cause security issues.
* HTTPS is not used, meaning connection is not secure
* No authentication is present
* Requests are not validated by the server, causing security issues.
* Greeting controller uses no validation
* Input is handled in the URL, causing leaks.

**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
* FasterXML Jackson Databind 2.10.2. Fixed with updating
  + CVE-2020-25649
* Apache Log4j API 2.12.1. Fixed with updating
  + CVE-2020-9488
* SnakeYAML 1.25. Fixed with updating to 1.26
  + CVE-2017-18640
* Spring Core 5.2.3. Fixed with updating to 5.2.8
  + CVE-2020-5421
* Apache Tomcat 9.0.30. Multiple vulnerabilities. Most of these are fixed by updating to 9.0.4
  + CVE-2019-17569
  + CVE-2020-11996
  + CVE-2020-13934
  + CVE-2020-13935
  + CVE-2020-13943
  + CVE-2020-17527
  + CVE-2020-1935
  + CVE-2020-1938
  + CVE-2020-8022
  + CVE-2020-9484
  + CVE-2021-24122
* Bouncy Castle 1.46. Multiple vulnerabilities. Fixed by updating to 1.6
  + 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

**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.

A swap to HTTPS is critical. Without this change, all communications have a limit to their security. All requests must be validated by the server before any data is sent. Built-in database parameters must also be removed to improve security. Add an authorization key, since one is currently not present. Update greeting controller to have input to limits to reduce arbitrary code execution. Remove URL handled input to reduce vulnerabilities. Update attached dependencies to their most recent release, fixing the attached errors.