# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **5/19/2025** | **Nick Botteicher** |  |

## 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

Nick Botteicher

**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 secure communications is very important to Artemis Financial because they handle very sensitive information like social security numbers and tax information as well as just handling large amounts of money. There is no solid evidence that Artemis acts only in the US so to be on the safe side we should assume they do business internationally as well. The main restriction from the government would be that there is no exposure to information regarding trade secrets. The major external threat at hand is hackers gaining the information of clients so the data will need to be heavily encrypted to anyone outside of the company. The big modernization they would need is to keep updating bugs and security measures found to be weak.

**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’s are going to be necessary to function internally and externally so there is an easy and efficient way to communicate data and decide which data is acceptable for access.

Input Validation will be required for the users to validate themselves before being able to access information.

Code Quality is always important so it can easily be maintained and function correctly.

Error handling methods would be needed to understand what parts of the API would need to be updated so the company would not have to worry about information being accessed by someone other than the user.

Cryptography would be needed so user information couldn’t be accessed from other parts of the world since we are assuming they are acting internationally and different currencies would be involved.

**3. Manual Review**

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

During mu vulnerability assessment I looked at both the POM.XML and the Greeting Controller. In the greeting controller I noticed there is a lack of input validation. The code quality is pretty good however I noticed there is no error handling included. The API needs a lot of updates, one main problem I see is there was a breach that could expose user input because it was not written with a POST method. Finally, there is no cryptography included.

**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

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

Bouncy Castle V1.46 has the following vulnerabilities:

CVE-2024-34447

CVE-2016-1000338

CVE-2016-1000342

CVE-2016-1000343

CVE-2024-29857

CVE-2016-1000341

CVE-2016-1000345

CVE-2024-30171

CVE-2020-15522

CVE-2020-0187

[CVE-2023-33202](https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2023-33202)

CVE-2020-26939

CVE-2023-33201

[CVE-2015-7940](https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2015-7940)

CVE-2018-5382

CVE-2013-1624

CVE-2016-1000346

CVE-2015-6644

From a quick search I believe the latest version would be 2.6.1 so I would recommend updating to this.

Faster.XML Jackson Data bind V2.10.2 Has the following vulnerabilities:

CVE-2020-25649

CVE-2020-36518

CVE-2021-46877

CVE-2022-42003

CVE-2022-42004

CVE-2023-35116

I believe the latest version of this is 2.9.5 so I would update to that.

Hibernate Validator V 6.0.18 has the following vulnerabilities:

CVE-2023-1932

CVE-2020-10693

I believe the latest version of this is 9.0.0 so I would update to that.

Apache Log4 V 2.12.1 has the following vulnerabilities:

CVE-2020-9488

I believe the latest version of this is 2.24.3 so I would update to that.

Logback Classic V 1.2.3 has the following vulnerabilities:

CVE-2023-6378

CVE-2021-42550

I believe the latest version of this is 1.5.18 So I would update to that.

Logback Core 1.2.3 Has the Following vulnerabilities:

CVE-2023-6378

CVE-2021-42550

CVE-2024-12798

CVE-2024-12801

I believe the latest version of this is 1.5.13 so I would update that.

Snake YAML V 1.25 has the following Vulnerabilities:

CVE-2022-1471

CVE-2017-18640

CVE-2022-25857

CVE-2022-38749

CVE-2022-38751

CVE-2022-38752

CVE-2022-41854

CVE-2022-38750

I believe the latest version is 2.4 so I would update to that.

Spring Boot 2.2.4 Has the following vulnerabilities:

CVE-2023-20873

CVE-2022-27772

CVE-2023-20883

I believe the latest version is 3.4.6 so I would update to that.

Spring Core 5.2.3 has the following vulnerabilities:

CVE-2022-22965

CVE-2021-22118

CVE-2020-5421

CVE-2022-22950

CVE-2022-22971

CVE-2023-20861

CVE-2023-20863

CVE-2022-22968

CVE-2022-22970

CVE-2021-22060

CVE-2021-22096

I believe the latest version is 7.0 So I would update to that.

Apache Tomcat 9.0.3 has the following vulnerabilities:

CVE-2020-1938

CVE-2024-52316

CVE-2025-24813

CVE-2025-31651

CVE-2020-11996

CVE-2020-13934

CVE-2020-13935

CVE-2020-17527

CVE-2021-25122

CVE-2021-41079

CVE-2022-29885

CVE-2022-42252

CVE-2023-44487

CVE-2024-24549

CVE-2024-38286

CVE-2020-9484

CVE-2021-25329

CVE-2021-30640

CVE-2024-23672

CVE-2022-34305

CVE-2023-41080

CVE-2021-24122

CVE-2021-33037

CVE-2023-42795

CVE-2023-45648

CVE-2024-21733

CVE-2019-17569

CVE-2020-1935

CVE-2020-13943

CVE-2023-28708

CVE-2021-43980

The latest version is 11.0.7 so I would update to that.

**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.

The quick remedy to the identified security vulnerabilities for Artemis Financial would be simply

running the current versions everything because continuing to run outdated versions will cause increasing security risks.