# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **9/20/2024** | **Kristin Tucker** | **Initial draft** |

## 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

Kristin Tucker

**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?

Artemis financial has indicated a need to acquire the most effective software security for their web application programming interface, RESTful. Artemis financial company is a consulting company that develops financial plans for customers that include savings, retirement, investments, and insurance. It is imperative to Artemis that their custom software is top notch as it plays a crucial role to the company’s success. The company wants to protect their application from external threats including international risks that they may encounter during financial transactions. Although there are no governmental restrictions listed in this scenario, it is important to be aware that some rules may apply when dealing with international accounts and customers and to also consider external threats that may be government related. Modern and future threats that cannot be ignored may include malware, phishing, data breaching, ransomware, and DDoS attacks. During the development process, communication breakdowns between the build team and the company may pose a risk along with ineffective project management among the developers. Resources to consider for modernization requirements include utilizing a more common up to date coding language and requiring all third-party resources to keep versions up to date. Utilizing open-source libraries are beneficial for facilitating and enabling customization to allow the developer to customize written code. Web applications may also prove useful to integrate machine learning and artificial intelligence which would provide more adaptive and end user centered web experiences.

**2. Areas of Security**

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

After review of what is required for Artemis financials web-based application software, there are several areas of security which raises concern.

* Input validation.

Accurate input validation should be evaluated to prevent vulnerabilities such as buffer overflow attacks, cross site scripting attacks, and SQL injection attacks.

* Authentication

The software should be able to provide authentication methods such as 2 factor or MFA to ensure only authorized users can access the data.

* Code Quality

Poor code ethics can lead to security issues that allow hackers to exploit software vulnerabilities, which can lead to data theft, sending malicious commands, and security breaches.

* Secure APIs

APIs are the pillar of web development systems and services. Lacking API security can suffer consequences such as exposing customer data, disrupt services and compromise systems. Common API security threats include SQLi, injection attacks, MITM attacks, CSRF, and path manipulations.

**3. Manual Review**

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

* There is no authentication or multifactorial authentication system available for verification.
* Cryptography is not encrypted.
* The system architecture does not use HTTPS, which is prudent to have when sharing personal an sensitive information.
* The presence of an API is not suggested anywhere. APIs would be wise to use since

**4. Static Testing**

Run a dependency check on Artemis Financials’ 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.

Bcprovjdk15on-1.46.jar- A total of 13 dependencies were discovered here. This dependency has several issues associated with it, including cryptography. The severity of the vulnerability is high on the radar, where it involves improper verification of cryptographic key, verification checks, and cryptographic signatures. This affects security information and leaves personal data open to discovery by hackers. A good solution to start here would be to update the version to the latest.

Snakeyaml-1.25.jar- The YAML vulnerability is associated with denial of service and resource consumption. The website or application gets more information than usual and is unable to handle the large amounts of data and can crash making it vulnerable to attack. Updating the YAML to the latest version would be the place to start to resolve the issue here.

Log4j-api-2.12.1.jar- The Apache Log4j API is related to improper certificate validation. A session can be hijacked which would allow log message to be compromised. The recommended solution would be to update the latest version which has a built in feature for verification.

Spring-boot-2.2.4.RELEASE.jar- This vulnerability affects the spring boot framework and unsupported versions. An application that id deployed could be susceptible to a security bypass. The fix for this would be to update the to version 3.0.6+.

Logback-core-1.2.3.jar- An attacker with privileges can edit config files and create malicious configurations allowing to execute arbitrary code. This involved certain versions. Updating to the latest version will resolve this issue.

Jackson-databind-2.10.2.jar- This affects data integrity from a vulnerable XML external entity, denial of services, resource exhaustion due to vulnerable software versions. Updates to the latest versions would address the issues listed.

Tomcat-embed-core-9.0.30.jar- This dependency has multiple vulnerabilities as well. One involves multiple sessions being made and leads to a denial of service. Users would also see responses from unexpected resources. The lower versions of Tomcat could re-use a HTTP and request header from a previous stream, this would open up the possibility that information could leak between requests. Updating to the latest Tomcat versions is a sure-fire way to fix the issue

Hibrernate-validator-6.0.18.Final.jar- A flaw was discovered in Hibernates lower versions allowing hackers to bypass input controls. Recommend update to Hibernates latest software 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 Financials’ software application.

After review of the multiple dependencies along with their vulnerabilities an effective mitigation plan is essential for Artemis Financials’ software development to be successful. It appears that the biggest issue with the vulnerabilities is the numerous versions needed upgraded to their corresponding latest versions. A strong combination of alphanumeric digits for both the username and password would also be a wise option to create. A articulate and thorough code review should be performed and modified as needed. The quality of the code needs to be top level to prevent improper authentication and data parsing. Verification and validation of certificates ensures that the code executed correctly.

**References**

Website <https://bugcrowd.com/glossary/input-validation-attacks/>