# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **03/21/2025** | **Dalton Johnson** | **Initial Assessment- N/A** |

## 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

Dalton Johnson

**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?

To estimate the value of Secure communications to the company, I am going to use the hypothetical loss of secure communications as a metric. The primary role of ARTEMIS FINANCIAL is to consult customers on how to manage their finances, including savings, investments, retirement, etc. Considering this is the role of the company, a loss in security would result in the loss of several, if not all financial information of the company’s customers. This will both result in the company being liable for their customers’ losses, as well as a significant loss of reputation. This loss of reputation could result in the loss of potential future customers, as well as current customers. Considering this, I would consider the security of communications to be extremely valuable to the company.

The company has international clients, including government agencies. This means that there will need to be international governmental restrictions placed based on the regions the clients reside / operate in. External threats include third party data breach, Outdated Software inconsistent with revised governmental restrictions, DOS / DDOS attacks restricting clients from managing their finances, along with insider threats from within the client organization, as well as its customers.

With the modernization of the client’s software, additional caution should be used to ensure the security of critical data. The use of open-source software should be kept to a minimum, and if it is used it must be examined for vulnerabilities that would jeopardize security. Additionally, a web application firewall can be utilized to protect the application by detecting abnormal requests. A WAF would provide additional protection against XSS attacks, SQL Injection, DoS / DDOS, as well as other attack methods. Encryption, data backups, as well as HTTPS are other methods that would be beneficial in building a secure web application for the client.

**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.

Referring to the Vulnerability Assessment Flow Chart, the areas that most closely apply to Artemis Financial’s software application are APIs, Cryptography, Client/Server, and Encapsulation. While each area is still very important, these specific areas listed above are concerned with securing the data from outside threats that may make use of the information in a malicious manner. The sensitivity of the information that the system will contain, along with the variety and number of customers that the software will be assisting with, makes the primary concern to containing their data safely from external actors. Input Validation, Code Error, and Code Quality are still important, considering that these aspects are concerned with data integrity, as well as handling internal problems that may arise with the program. Each of the aspects within the flow chart is important to consider with the application’s development, however I suggest an emphasis be placed on the aspects regarding security from third parties and information leaks, to maintain the client’s reputation with their customers.

**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.java line 13 – SQL Injection is possible
2. Customer.java line 7 – no username check to access account number
3. Customer.java line 7 – no password check to access account number
4. Customer.java line 12 – no verification of account to process a deposit
5. DocData.java line 14 – no verification of account to access ID
6. DocData.java line 27 – SQL Injection is possible
7. GreetingController.java line 16 – SQL injection is possible

**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 service

AI-generated content may be incorrect.Dependency Report Summary:

**Vulnerability Report Assessment**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Vulnerability Code** | **Vulnerability Description** | **Severity** | **Recommended Solutions** | **Attribution Documents** |
| [**CVE-2020-10693**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-10693)  **CVE-2023-1932** | * **Allows attackers to bypass input sanitation controls** * **Allows XSS attacks** | **M** | **Appply Critical Patch Update Security Patches ASAP** | [**Oracle Critical Patch Update Advisory - April 2022**](https://www.oracle.com/security-alerts/cpuapr2022.html) |
| **CVE-2022-22968** | * **Field is not protected unless listed with both upper and lower case for first character of the field** | **M** | **NetApp Security patches, no workaround** | [**CVE-2022-22968 Spring Framework Vulnerability in NetApp Products | NetApp Product Security**](https://security.netapp.com/advisory/ntap-20220602-0004/) |
| **CVE-2022-22950**  **CVE-2023-20861**  **CVE-2023-20863**  **CVE-2024-38808** | * **User can provide SpEL expression to cause DoS** * **See above** * **See above** * **See above** | **M** | Upgrade to 5.3.17+ or 5.2.20+ | [**https://spring.io/security/cve-2022-22950**](https://spring.io/security/cve-2022-22950) |
| [**CVE-2021-42550**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-42550)  [**CVE-2023-6378**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2023-6378) | * **An attacker with privlileges could edit config files maliciously** * **Allows attacker to mount a DoS attack through poisoned data** | **H** | **NetApp Security Patches, no workaround** | [**https://security.netapp.com/advisory/ntap-20211229-0001/**](https://security.netapp.com/advisory/ntap-20211229-0001/) |
| [**CVE-2021-42550**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-42550)  [**CVE-2023-6378**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2023-6378)  **CVE-2024-12798**  **CVE-2024-12801** | * **Allows attacker tyo craft malicious config to execute code** * **Allows attacker to mount DoS** * **Allows attacker to compromise existing logback or inject an environment variable before program execution** * **Allows attacker to forge requests by compromising logback config files** | **H** | **Netapp Security Patches, no workaround** | [**https://security.netapp.com/advisory/ntap-20211229-0001/**](https://security.netapp.com/advisory/ntap-20211229-0001/) |
| [**CVE-2020-25649**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-25649)  [**CVE-2020-36518**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-36518) | * **Allows vulnerability to XXE attacks** * **Allows Java StackOverflow exception and DoS** | **H** | **Update to Jackson-databind-2.11.0 or 2.10.5.1** | [**https://bugzilla.redhat.com/show\_bug.cgi?id=1887664**](https://bugzilla.redhat.com/show_bug.cgi?id=1887664) |
| [**CVE-2013-1624**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2013-1624)  **CVE-2015-6644**  **CVE-2015-7940** | * **Allows remote attackers to conduct attacks** * **Allows attackers to obtain sensitive information via crafted application** * **Does not validate a point, making it easier to obtain private keys via a series of crafted Elliptic curve Diffie hellman (ECDH) key exchanges (invalid curve attack)** | **H** | **Install Updates after backing up your existing installation** | [**https://access.redhat.com/errata/RHSA-2014:0371.html**](https://access.redhat.com/errata/RHSA-2014:0371.html) |
| **CVE-2021-22060**  **CVE-2024-38816** | * **User can provide malicious input to cause insertion of additional log entries** * **an attacker can craft malicious HTTP requests to obtain any file on the system** | **H** | **Update Spring Framework past 5.3.13** | **Found in report** |
| **CVE-2020-5421**  **CVE-2021-22096** | * **protections against RFD attacks may be bypassed depending on browser** * **it is possible to provide malicious input to insert additional log entries** | **H** | **Update Spring Framework past 5.3.10** | **Found in report** |
| **CVE-2022-22965** | * **vulnerable to remote code execution (RCE) via data binding** | **H** | **Workaround - Block both incoming and outgoing connections between the system and the internet**    **Protect network access to devices with appropriate mechanisms** | [**https://cert-portal.siemens.com/productcert/pdf/ssa-254054.pdf**](https://cert-portal.siemens.com/productcert/pdf/ssa-254054.pdf) |
| [**CVE-2022-27772**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-27772)  [**CVE-2023-20873**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2023-20873) | * **vulnerable to temporary directory hijacking** * **susceptible to security bypass** | **C** | **Upgrade to Spring-Boot past 3.0.6+ or 2.7.11+** | **Found in report** |
| [**CVE-2021-4235**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-4235)  [**CVE-2022-1471**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-1471) | * **A maliciously crafted YAML file can cause the system to consume significant resources, causing DoS** * **Deserializing yaml content by attacker can lead to remote code execution** | **C** | **Upgrade SnakeYaml past 2.0** | **Found in report** |
| [**CVE-2019-17569**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2019-17569)  [**CVE-2020-11996**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-11996) | * **Invalid Transfer-Encoding headers are incorrectly processed leading to HTTP request smuggling** * **A craftyed HTTP/2 sequence of requests could trigger high CPU usage for several seconds, which if repeated, the server could become unresponsive** | **C** | **Update to Tomcat 9.0.31** | [**https://lists.opensuse.org/archives/list/security-announce@lists.opensuse.org/message/F3FOVKJAK2YR7UVBYBATR7JKLD5IA6WI/**](https://lists.opensuse.org/archives/list/security-announce@lists.opensuse.org/message/F3FOVKJAK2YR7UVBYBATR7JKLD5IA6WI/) |
| [**CVE-2020-9488**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-9488) | * **Improper validation could allow an SMTPS connection to be intercepted by a man-in-the-middle attack** | **C** | **Fixed in Apache Log4j 2.12.3 and 2.13.1** | [**[SECURITY] [DSA 5020-1] apache-log4j2 security update**](https://lists.debian.org/debian-security-announce/2021/msg00206.html) |

**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.

To mitigate the risk that the listed security vulnerabilities provided above, it would be ideal to go forward with implementing the recommended solutions from within the table, or similar solutions to remove the threats of the vulnerabilities found within the code. For many of the vulnerabilities found, the solution can be easily implemented by updating the software to a newer version with the correct patches. Additionally, system network analysis should be performed to determine what machines have access to what information, and whether they should be connected directly to the internet. Access controls and permissions will also need to be assessed, with the ideal structure following the principle of least privilege, to reduce the risk of the human element.