# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **01/21/25** | **Chiara Aziz** |  |

## 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

Chiara Aziz

**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 deals with client investments. The monetary accounts, amounts and investments of client’s means there is a lot of sensitive information to be exchanged or communicated. It would be beneficial for Artemis to ensure they have secure communications to protect clients and their financial data. Artemis deals in investments so there very well could be international transactions that the company would need to complete on behalf of their clients. There are laws related to industry that put safeguards in place to protect consumers and their data. Considerations will need to be made to ensure there are company policies/plans that meet requirements when communicating securely with clients. Since Artemis Financial deals with sensitive data role restrictions and data access limitations should be in place to ensure they are resistant to injection attacks. Phishing attacks, malware, and communication interception are all outside threats that Artemis could face now and in the future. The use of libraries and other web applications opens a whole new world of possibilities for an application, but it is important to consider security. Ensuring the third-party libraries are up to date and don’t include any known vulnerabilities will help keep applications secure. A robust security plan that covers all areas is the best way to help protect Artemis and their clients.

**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.

Input validation- Input validation would be beneficial for Artemis Financial as uses have to input information about themselves into forms and someone with malicious intent could easily use this route to gain access to the system if this is not protected.

API- It would be a good idea if the APIs that Artemis uses utilize protection via authorization and access controls. This will help keep those that are not supposed to have access to certain areas of the website out.

Secure coding patterns- Since Artemis deals with client financial information strong secure coding practices would be good to help keep the site safe. Secure practices protect the company and client data.

Cryptography-Since Artemis Financial has lots of sensitive data it would be beneficial to integrate the principles of encryption into their code base to help manage

**3. Manual Review**

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

When reviewing the code there are a couple of things I noticed. It looks like there is no input validation. This should be addressed to prevent injection or input attacks. Input validation ensures users can only input string of certain lengths or other requirements so a user can’t input malicious commands. The code doesn’t seem to exhibit any encryption of any sort of cryptograph. The code also has plain text username and password listed in the code this is more of a coding practice as these values shouldn’t be coded in the files but in environment variables or pulling in through the properties file environment file or other secure means. There are also some issues with the controller, it is taking the input directly from the parameter allowing the user to tell to program who the user is. There should be verifications ensuring the user is who they say they are.

**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

|  |  |  |
| --- | --- | --- |
| **Vulnerability** | **Description/Solution** | **Documentation** |
| bcprov-jdk15on-1.46.jar | Bouncy castle – a java package used for the implementation of cryptographic algorithms. Switch to updated version. | cpe:2.3:a:bouncycastle:fips\_java\_api:-:\*:\*:\*:\*:\*:\*:\*  cpe:2.3:a:bouncycastle:fips\_java\_api:1.0.1:\*:\*:\*:\*:\*:\*:\*  cpe:2.3:a:bouncycastle:fips\_java\_api:1.0.2:\*:\*:\*:\*:\*:\*:\*  cpe:2.3:a:bouncycastle:fips\_java\_api:1.0.2.3:\*:\*:\*:\*:\*:\*:\* |
| jackson-databind-2.10.2.jar | General data-binding functionality for Jackson: works on core streaming API. Threats to data integrity. Upgrade packages. | cpe:2.3:a:fasterxml:jackson-databind:\*:\*:\*:\*:\*:\*:\*:\*  Show Matching CPE(s) Up to (excluding)  2.12.7.1  cpe:2.3:a:fasterxml:jackson-databind:\*:\*:\*:\*:\*:\*:\*:\*  Show Matching CPE(s) From (including)  2.13.0 Up to (excluding)  2.13.4  Configuration 2 ( hide )  cpe:2.3:a:quarkus:quarkus:\*:\*:\*:\*:\*:\*:\*:\*  Show Matching CPE(s) |
| logback-classic-1.2.3.jar | Sets up logging framework.  In logback version 1.2.7 and prior versions, an attacker with the required privileges to edit configurations files could craft a malicious configuration allowing to execute arbitrary code loaded from LDAP servers. Update to a more recent verision. | |  |  | | --- | --- | | **cpe:2.3:a:qos:logback:\*:\*:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2021-42550#range-15434892) | **Up to (including) 1.2.7** | | **cpe:2.3:a:qos:logback:1.3.0:alpha0:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2021-42550#match-15434893) | | | **cpe:2.3:a:qos:logback:1.3.0:alpha1:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2021-42550#match-15434894) | | | **cpe:2.3:a:qos:logback:1.3.0:alpha10:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2021-42550#match-15434891) | | | **cpe:2.3:a:qos:logback:1.3.0:alpha2:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2021-42550#match-15434895) | | | **cpe:2.3:a:qos:logback:1.3.0:alpha3:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2021-42550#match-15434896) | | | **cpe:2.3:a:qos:logback:1.3.0:alpha4:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2021-42550#match-15434897) | | | **cpe:2.3:a:qos:logback:1.3.0:alpha5:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2021-42550#match-15434898) | | | **cpe:2.3:a:qos:logback:1.3.0:alpha6:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2021-42550#match-15434887) | | | **cpe:2.3:a:qos:logback:1.3.0:alpha7:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2021-42550#match-15434888) | | | **cpe:2.3:a:qos:logback:1.3.0:alpha8:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2021-42550#match-15434889) | | | **cpe:2.3:a:qos:logback:1.3.0:alpha9:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2021-42550#match-15434890) | | |
| logback-core-1.2.3.jar | Logging features in an application.  In logback version 1.2.7 and prior versions, an attacker with the required privileges to edit configurations files could craft a malicious configuration allowing to execute arbitrary code loaded from LDAP servers. Update to a more recent version. | |  |  | | --- | --- | | **cpe:2.3:a:qos:logback:\*:\*:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2021-42550#range-15434892) | **Up to (including) 1.2.7** | | **cpe:2.3:a:qos:logback:1.3.0:alpha0:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2021-42550#match-15434893) | | | **cpe:2.3:a:qos:logback:1.3.0:alpha1:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2021-42550#match-15434894) | | | **cpe:2.3:a:qos:logback:1.3.0:alpha10:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2021-42550#match-15434891) | | | **cpe:2.3:a:qos:logback:1.3.0:alpha2:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2021-42550#match-15434895) | | | **cpe:2.3:a:qos:logback:1.3.0:alpha3:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2021-42550#match-15434896) | | | **cpe:2.3:a:qos:logback:1.3.0:alpha4:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2021-42550#match-15434897) | | | **cpe:2.3:a:qos:logback:1.3.0:alpha5:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2021-42550#match-15434898) | | | **cpe:2.3:a:qos:logback:1.3.0:alpha6:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2021-42550#match-15434887) | | | **cpe:2.3:a:qos:logback:1.3.0:alpha7:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2021-42550#match-15434888) | | | **cpe:2.3:a:qos:logback:1.3.0:alpha8:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2021-42550#match-15434889) | | | **cpe:2.3:a:qos:logback:1.3.0:alpha9:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2021-42550#match-15434890) | | |
| snakeyaml-1.25.jar | Improper input validation. Deserializing yaml provided by attackers can lead to remote code execution. Recommended to switch to version 2.2 and beyond. | cpe:2.3:a:snakeyaml\_project:snakeyaml:\*:\*:\*:\*:\*:\*:\*:\* |
| spring-boot-2.2.4.RELEASE.jar | Makes the development of applications easier. Update to 2.2.11 or later at least as this was patched. | cpe:2.3:a:vmware:spring\_boot:\*:\*:\*:\*:\*:\*:\*:\* versions up to (excluding) 2.5.15  cpe:2.3:a:vmware:spring\_boot:\*:\*:\*:\*:\*:\*:\*:\* versions from (including) 2.6.0; versions up to (excluding) 2.6.14  cpe:2.3:a:vmware:spring\_boot:\*:\*:\*:\*:\*:\*:\*:\* versions from (including) 2.7.0; versions up to (excluding) 2.7.11  cpe:2.3:a:vmware:spring\_boot:\*:\*:\*:\*:\*:\*:\*:\* versions from (including) 3.0.0; versions up to (excluding) 3.0.6 |
| spring-expression-5.2.3.RELEASE.jar | Has vulnerability against RFD attacks. Upgrade to the latest version. | |  |  | | --- | --- | | **cpe:2.3:a:vmware:spring\_framework:\*:\*:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2020-5421#range-15156850) | **Up to (excluding) 4.3.29** | | **cpe:2.3:a:vmware:spring\_framework:\*:\*:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2020-5421#range-15156849) | **From (including) 5.0.0** | **Up to (excluding) 5.0.19** | | **cpe:2.3:a:vmware:spring\_framework:\*:\*:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2020-5421#range-15156847) | **From (including) 5.1.0** | **Up to (excluding) 5.1.18** | | **cpe:2.3:a:vmware:spring\_framework:\*:\*:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2020-5421#range-15156848) | **From (including) 5.2.0** | **Up to (excluding) 5.2.9** | |
| spring-web-5.2.3.RELEASE.jar | Provides support like rest services to the spring framework. Upgrade to at least 5.2.19. | |  |  |  | | --- | --- | --- | | **cpe:2.3:a:vmware:spring\_framework:\*:\*:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2021-22060#range-15285865) | **From (including) 5.2.0** | **Up to (including) 5.2.18** | | **cpe:2.3:a:vmware:spring\_framework:\*:\*:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2021-22060#range-15285864) | **From (including) 5.3.0** | **Up to (including) 5.3.13** | |
| spring-webmvc-5.2.3.RELEASE.jar | Helpful for creating endpoints in a spring application. Upgrade to 5.2.21. | |  |  | | --- | --- | | **cpe:2.3:a:vmware:spring\_framework:\*:\*:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2022-22968#range-15521376) | **Up to (excluding) 5.2.0** | | **cpe:2.3:a:vmware:spring\_framework:\*:\*:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2022-22968#range-15521377) | **From (including) 5.2.0** | **Up to (including) 5.2.20** | | **cpe:2.3:a:vmware:spring\_framework:\*:\*:\*:\*:\*:\*:\*:\***    [Show Matching CPE(s)](https://nvd.nist.gov/vuln/detail/CVE-2022-22968#range-15521378) | **From (including) 5.3.0** | **Up to (including) 5.3.18** | |

**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.

I would start by upgrading all the versions of necessary dependencies based off the above report. Getting everything up to date would be a great start on increasing the security of the application. The greeting controller on line 16 is taking the name from the parameter . Currently I can type anything into the url and inject it so there should be a some input validation and the controller could get that data not from the url but from a logged in use or some sort of validation authentication service. In addition to that the controller could benefit from authentication since anyone with the address can access it on line 15. On line 27 of the DocData.java file the username and password are hard coded into the file. These should be moved to the application.properties filed and pulled from an .env or pipeline to keep them secure and could benefit from encryption. These changes should help make the application more secure.