# CS 305 Project One – Analysis of Advanced Security Concepts and Recommendations

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **September 22, 2024** | **Niyati Shetty** | **Analysis of advanced security concepts** |

## 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

**Niyati Shetty**

**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 handles sensitive financial data of customer savings plan, investment accounts, retirement accounts, and insurance details. The breach of security can lead to financial loss, trust of the customer, severe penalties from regulators. So ensuring security, integrity of data is very critical for the company.

* Are there any international transactions that the company produces?  
    
  As a Financial Company Artemis can have clients from different countries and clients can do transactions from different countries or domestic clients when travelling aboard can do the transaction. So, financial industry compliance like PCI DSS, GDPR, FINRA, should be maintained.
* Are there governmental restrictions on secure communications to consider?  
    
  Yes, there are several regulations and restriction for secure communications, like General Data Protection Regulation (GDPR) in European union, Financial Industry Regulatory Authority (FINRA) and necessary and it should maintain Encryption regulations standards of different countries.
* What external threats might be present now and in the immediate future?  
    
  There are several external threats, like Phishing attack, Bot manager attacks to steal user id and password, XSS – cross site scripting, Denial of service attack. Slow post attack.
* What modernization requirements must be considered, such as the role of open-source libraries and evolving web application technologies?  
    
  As the company Artemis Financials’ software will rely on open-source libraries. The open source libraries can create lots of security vulnerabilities, so updating these library to the latest version and checking for security vulnerabilities, is very important. The web frameworks like “Struts” and “Spring” frameworks had major security breaches, so regular scanning of these libraries and keeping up to date, these libraries with security fixes are very important. When RESTful API are used, checking the endpoints are secure and no vulnerabilities exist is very important. API security is a very important part of securing the software. Need to follow good security practice for maintaining the Container security.

**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: This is a very high risk vulnerability, the software should validate the users input, before accepting and storing it in the database. Without proper validation there are several attacks can happen to the system, like SQL Injection, Cross-site scripting, etc., this can lead to data leak.
* APIs: The API security is very critical, since Artemius Financial uses RESTful APIs, securing the API endpoints is very essential. Proper Authentication and authorization are very important. The API should be rate limited and should prevent unauthorized access. The API should have limited functionalities and should not be able to access entire data set. It should have layered access to the data.
* Cryptography: The company should use TLS 1.2+ secure protocol to transmit data along with strong encryption of AES-256. This helps to protect sensitive information from being tampered or intercepted. Having proper encryption is very high in priority. This will prevent man in the middle attack.
* Client/Server: The web-based application is based on client-server model. The clients can be various web browsers, mobile app, desktop apps, etc. Having proper security is very important. The Clients accessing the data should be authenticated, and based on the clients authorization, the server should only provide the data as needed. The client session should be time limited, and it should re-authenticate very set period. The clients also should have multi-factor authentication, not just simple password authentication.
* Code Error: The software should handle the error gracefully, when the error occurs, the entire stack trace should not be exposed to the external users. It can be logged in the secure place, where developer can access. Exposing the entire stack trace or very verbose error messages to users can expose the software stack and attacker can exploit that information. When code error occurs, displaying or sending back some type of error msg with error code, where user can understand and report back the code to support staff, to help them debug the issue.
* Code Quality: Bad code can introduce several security vulnerabilities. The developer and testers should be trained in secure coding practices. This will ensure quality code, and the code design, data storage, data management and code review with people trained in secure coding methods can ensure quality code, which is less vulnerable. The development team should also use industry standard scanning tools to ensure the code developed is secure.
* Encapsulation: The well-defined encapsulation can ensure the data is secure and if any breach happens, the attacker will not be able to obtain all data. The sensitive data is protected from un-authorized access, and it will avoid unintentional expose. This will also ensure only authorized people with high level of access can view sensitive data of all customers etc.

**3. Manual Review**

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

After manual review of code, the following vulnerabilities were identified:

* SQL injection risk due to lack of input validation.
* Insecure password storage and the credentials are hardcoded.
* Risk of cross-site scripting – the user’s input is not sanitized before displayed back to the user.
* Insecure data encryption – the encryption is very weak; it is recommended to use a strong algorithm.
* Error handling is not good, it can lead to information leak and help attacker with inside information of the software stack.
* Session management is not good. The session does not time out, and the session can be very long and no multifactor authentication is done.
* Week password algorithm, if the attacker had access to database, they can easily decode the passwords
* Un-secure data transmission of sensitive and financial data.
* Insecure dependency – the open-source libraries can be outdated and contain lot of vulnerabilities.

**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

**[CVE-2023-20873](https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2023-20873" \t "_blank)** - In Spring Boot versions 3.0.0 - 3.0.5, 2.7.0 - 2.7.10, and older unsupported versions. The VMware Aria operations for network. The attacker with admin privileges can exploit and gain unauthorized access to the system. Upgrading the high version is recommended.

[**CVE-2020-1938**](https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-1938) – Apache Tomcat - Apache Tomcat Improper Privilege Management Vulnerability. When using Apache JServ Protocol (AJP) care must be taken when trusting the incoming connection to the Apache Tomcat server. If the port 8009 has access to the attacker, they can craft bad AJP request and get sensitive configuration. The recommendation is to disable the AJP connector port. And comment out the AJP connector configuration in server.xml. Upgrade tomcat to latest 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 Financial’s software application.

The Artemis Financial Web application contains several vulnerabilities, which was generated by “dependency check report” and manual review and static test report. Below are some of the steps identified to help fix these vulnerabilities.

* CVE-2023-20873 – Gaining unauthorized access. This can be avoided by upgrading to VMWare Aria operation to 6.9.0 or later version. Restrict admin access and implement multi-factor authentication.
* CVE-2020-1938 – File inclusion vulnerability which can expose sensitive files. This vulnerability can be fixed by disabling AJP in server.xml config file and upgrade Tomcat to 9.0.31+ version
* Train the development team (developer and QA) with OWASP secure coding practices.
* Conduct code review
* Scan code using the security tools which scan for vulnerabilities , like Veracode,Webinspect, sqlmap, Netsparker, Bandit, CodeQL etc.
* Conduct security audit and implement the recommendations