# CS 305 Module Two Written Assignment Template

## Instructions

Replace the bracketed text with the relevant information in your own words. If you choose to include images or supporting materials, make certain to insert them in all the relevant locations in the document.

## Areas of Security

According to the Vulnerability Assessment Process Flow Diagram, the seven areas of security relevant to assess at the first level of vulnerability are as follows: Input Validation, APIs, Cryptography, Client/Server, Code Error, Code Quality, and Encapsulation.

## Areas of Security Justification

To begin our flow chart, we start with Architecture Review. Although this was not listed previously, it is worth noting that examining and analyzing the application's architecture can help identify any obvious flaws and vulnerabilities that need immediate repair.

**Input Validation -** According to cheatsheetseries.owasp.org, there are two main input validation strategies. These strategies, both syntactic and semantic, ensure that, first, the correct syntax is followed, and second, that the information is accurate when compared to the rest of the context. For the case of security, the application could contain easily obtained information that introduces vulnerabilities into the program, allowing unauthorized users to gain access. Once inside, user information can be stolen, and the program itself changed and corrupted.

**APIs** – APIs stand for Application Programming Interface. What APIs do is communicate from application to application, which helps prevent data from being leaked. Many APIs on the web use HTTP protocols, which are used in request-response cycles. APIs, in this case, are used to help maintain privacy by providing the user with only what is needed.

**Cryptography -** Cryptography is a way for computers to encode text in between readable formats. This helps maintain data privacy when browsing the web. This not only allows the user confidence, but also integrity when it comes to the information itself, ensuring that it cannot be modified. Adaptability also ensures that the information remains encrypted in the face of any threats, along with authentication, which confirms that it was both received and sent by the intended parties. This ensures that the sending and receiving by both the client and the computer remain secure and protected.

**Client/Server** – There are numerous vulnerabilities on both the client and server sides. The client side can become extremely vulnerable if the server side gets hacked. Once the server-side is down, it essentially provides free access to the client-side. Any information should follow the previous steps to ensure that all data is safe and protected before being sent to the client side, and vice versa, to ensure the safety of both the client and server sides.

Code Error – This can commonly occur on the client side, so it is essential to review the code to ensure that there are no threats that can be exploited. It is important to have secure error handling.

Code Quality – Practicing good coding techniques and patterns enables a safer experience, which in turn enhances the program's protection. Any flaws have the potential to be exploited by hackers, which can lead to major issues later on.

Encapsulation – This adds the final layer of protection to the first layer of security. Encapsulation allows for the authorization of specific areas of the program, while not allowing access to unauthorized parts. By not using encapsulation, this would enable users to access the code that they are not supposed to have.

## Code Review Summary

The first issue with the code is the pom.xml file. The issue is that the version that is currently in the code, 2.6.5 RELEASE, is overriding the more up-to-date and current 3.2.4 RELEASE. The next noticeable flaw is in the code quality. The code in GreetingController.java allows for the user to input a string, while this is fine, it doesn’t have any countermeasures for an input that’s not allowed. This can create a security breach that can be exploited.

## Mitigation Plan

We can start by updating the pom.xml file to the current release, which would help patch any new hacks. To fix the second issue, we can add a verification, which would add another layer of security.