# 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

The areas of security that are relevant to assess in this scenario from the first level of the vulnerability assessment process flow diagram are:

* Input validation
* APIs
* Code errors
* Encapsulation

## Areas of Security Justification

Input validation

* Validating user input is crucial for mitigating vulnerabilities; Incoming data, even from sources that might be trustworthy, should be sanitized. Any errant input can lead to unexpected application behavior.

APIs

* Securing the API adds an additional layer of protection, ensuring both the connection and the user’s authenticity have been verified.

Code errors

* Error handling should be generic to avoid revealing details that could help attackers gain insights into the application’s structure.

Encapsulation

* Utilizing encapsulation protects data from unauthorized modification. This can be done by declaring variables as private and using getter and setter methods for controlled access.

## Code Review Summary

* The number() method in the GreetingController class retrieves array elements without a boundary check that throws a generic error exception, unlike ArrayIndexOutOfBoundsException, which could be exploited by an attacker.
* The number() method is missing input type validation. The method expects int type input for id, but there is no input validation in place to handle non-integer value input.
* Methods have get/set functions available in the Greeting class and parameters are set to private. GreetingController does not use the get/set functions.
* In the number() method, the statement *String message = "Element in the given index is :: "+myArray[id]* requires sanitization to prevent injection attempts via the id variable obtained from the user.
* User inputs are not checked for length, running the risk of buffer overruns.
* An outdated Spring framework is being invoked in the pom.xml file.

## Mitigation Plan

* Implement error sanitization to prevent leaking sensitive information about the program.
* Implement getter/setter methods to protect from unauthorized modification to the program.
* Implement input sanitation and type validation.
* Update Spring framework to current version before attempting dependency check; Address further vulnerabilities found in the dependency scan.