# CS 305 Module Two Written Assignment – Armon Wilson

## 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

* Input Validation
* APIs
* Cryptography
* Client/Server
* Code Quality
* Encapsulation

## Areas of Security Justification

* Input Validation: The application's command input function should require input validation to prevent command injection attacks, where malicious users could submit harmful commands.
* APIs: If the application exposes APIs, they become potential entry points for attackers. A thorough review is needed to ensure proper authentication, authorization, and input validation mechanisms.
* Cryptography: It's essential to consider how sensitive data (if any) would be protected during transmission and storage.
* Client/Server: The client/server architecture of the web application introduces potential vulnerabilities in the communication and interaction between the client and server components.
* Code Quality: While not directly a vulnerability, poor code quality can lead to unintended security flaws.
* Encapsulation: Proper encapsulation of data and logic within classes and methods is vital for maintaining security. It helps prevent unintended data exposure and unauthorized access to sensitive functionality.

## Code Review Summary

* Architecture
  + Old version of spring framework used (2.6.5) which may contain unpatched vulnerabilities.
    - **Vuln ID** [**CVE-2022-31679**](https://nvd.nist.gov/vuln/detail/CVE-2022-31679)
      * “Applications that allow HTTP PATCH access to resources exposed by Spring Data REST in versions 3.6.0 - 3.5.5, 3.7.0 - 3.7.2, and older unsupported versions, if an attacker knows about the structure of the underlying domain model, they can craft HTTP requests that expose hidden entity attributes.”
    - **Vuln ID** [**CVE-2021-22047**](https://nvd.nist.gov/vuln/detail/CVE-2021-22047)
      * “In Spring Data REST versions 3.4.0 - 3.4.13, 3.5.0 - 3.5.5, and older unsupported versions, HTTP resources implemented by custom controllers using a configured base API path and a controller type-level request mapping are additionally exposed under URIs that can potentially be exposed for unauthorized access depending on the Spring Security configuration.”
* GreetingController.java
  + SpEL Injection Vulnerability
    - The greeting method is vulnerable to SpEL injection due to lack of input sanitization/validation, allowing attackers to execute malicious code

## Mitigation Plan

* Architecture
  + Upgrade Spring Framework to latest version to alleviate identified vulnerabilities.
    - **Vuln ID** [**CVE-2022-31679**](https://nvd.nist.gov/vuln/detail/CVE-2022-31679)
      * Users of affected versions should apply the following mitigation: 3.6.x users should upgrade to 3.6.7+ (included in Spring Boot 2.6.12+). 3.7.x users should upgrade to 3.7.3+ (included in Spring Boot 2.7.4+). No other steps are necessary. Releases that have fixed this issue include:
        + Spring Data REST
        + 3.6.7+
        + 3.7.3+
    - **Vuln ID** [**CVE-2021-22047**](https://nvd.nist.gov/vuln/detail/CVE-2021-22047)
      * Users of affected versions should upgrade to one of the versions below. No other steps are necessary.
        + Spring Data REST
        + 3.4.14+ (included in Spring Boot 2.4.12+)
        + 3.5.6+ (included in Spring Boot 2.5.6+)
* GreetingController.java
  + SpEL Injection Vulnerability
    - Implement input validation for the “name” parameter in the “greeting” method. Input validation techniques:
      * Whitelist Validation
      * Blacklist Validation
      * Length Limitation

References/Websites/Resources:

<https://0xn3va.gitbook.io/cheat-sheets/framework/spring/spel-injection>

<https://cve.mitre.org/cve/search_cve_list.html>

<https://spring.io/security/cve-2022-31679>

<https://spring.io/security/cve-2021-22047>

<https://learning.oreilly.com/library/view/iron-clad-java/9780071835886/ch07.html#ch07lev1sec1>