# CS 305 Module Two Written Assignment - Kyle Cortez

## 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 most relevant areas of the seven areas of security to asses in relation to the task of implementing an expressive command input function is Input Validation with a secondary area most relevant as Code Error Handling. In a supremely robust input function, both are handled in with the other. An input must meet the criteria to pass through the validator function. Any error that may arise from this must be handled in a general sense so that something like an overflow error could not serve to bypass the input validation and escape to a different part of the application. In this way both should be handled at the same time and cover the other.

## Areas of Security Justification

Input validation – Relevancy is set within ensuring that any input introduced to the program is validated by a criteria that allows no bypass or through available common mechanism such as query parameterization.

APIs – Ensuring that each API used is up to date and secured against common attack methods is critical to the health of the application and any failures must either be addressed with a fix or migration to a more secure version/library.

Cryptography – Encryption is vital to the submission and receipt of data so that no unwanted/tainted data can be passed to a server or client through secured tunnels or by encrypted packets.

Client/Server – Security with client-server distributed systems ensure that a compromised network can be isolated by encapsulating each connected network. While connections do still exist between each network, they are limited and secured in depth.

Code Error – Error handling must be accounted for such that an attacker may not bypass a secure system by inducing an error that escapes outside of the guarded connection.

Code Quality – reduction of antipatterns and induction of industry standard positive patterns at the start of the project and throughout ensure, or at least lessen the probability, that insecurities are not unknowingly introduced.

Encapsulation – Encapsulation of data structures ensure that a breach in one structure does not ensure a breach throughout a system and data exposed is limited.

## Code Review Summary

Upon review, findings indicate that there should be an extra layer of defense where inputs are validated before being passed. This would ensure that unwanted/malicious input cannot be passed. Additionally, any unknown errors that may be passed through the code must be able to be caught to ensure the program receives valid inputs.

## Mitigation Plan

Firstly, a way to validate inputs to ensure they meet criteria aligned with what is expected for the program is necessary to develop. By ensuring that input is congruent standards, such as length restrictions, ensures that unnecessary or malicious input poses less risk to this entry point to the program. Secondly, error handling may need to be expanded so that errors may not escape or bypass/circumvent the path of the program. Standard practices for mitigating induced errors, accidental or otherwise, must be set as a layer of security to the program to further the defense in depth.