Critical Thinking 3

Gregory Norris

CSC 450 – Haseltine

Colorado State University Global Campus

Critical Thinking 3

Critical Thinking 3 requires a C++ program which takes three integer numbers from a user. The program then assigns three pointers to each integer and returns the pointer’s destination value, destination memory address and the pointer’s memory address. Some care is taken to prevent some types of known exploits from occurring within the program. The program should be able to take a string value from the user, convert it into an integer value and perform the required operations. When a value is supplied which is not an integer, or is too large of an integer, the program will use try/catch statements to resolve the issue.

The first part of the program is a for loop which runs three times. This loop’s purpose is to prompt the user for an integer input, receive input, perform basic error checking and return the required values from the pointers. To keep everything tidy, I chose to store the integer values within an array, and to store the pointer values within another array. The program uses nested for loops to determine the relationship between the integer values and the pointer values and store the pointers within a secondary array according to their corresponding integer number.

This program uses cin and cout to handle input and output since both of these functions are relatively easy to protect and contain some built-in validation and error checking. I did find that it was easier to accept the user input as a string as convert to an integer, rather than allowing for cin to input directly into an integer value.

Currently if the user inputs a valid integer followed by garbage data (or simple formatting exploits) the program will process the valid integer part of the string, then either discard the additional data or throw an error, assuming that the subsequent data is also not a valid integer.

One thing I have been trying to figure out is that the catch-all try statement tends to run the parent for loop twice every time there is a caught exception, this could affect program performance, but it seems to catch most exploits I have attempted as input. I am currently unaware of any exploit using the std::stoi() function, so I have used this to both validate my string data and to translate the string data into an integer value to be stored in the program.