Cpr E 430 lab 4

ASLR

I believe the Kali machine uses ASLR. I read online that hackers try to take advantage of this by trying to use their programs to write outside of the intended region in hopes of breaking the code. After going through the lab and seeing how the buffer overflow coming from our user input would end up writing into the command memory space I was able to determine that the Kali machine must use ASLR since this kind of attack was very similar to the simplified one we worked through during Experiments 2 and 3 before fixing the buffer overflow issue in 4.

Experiment 1

Graphical user interface, text, application

Description automatically generated

Experiment 2

Text

Description automatically generated

Experiment 3

Graphical user interface, text, application

Description automatically generated

Experiment 4

Graphical user interface, text

Description automatically generated

Reflection

I think this lab exercise helped me understand the concept of buffer attacks and how it relates to buffer overflows. In my 288 class we programmed in C and I was getting a buffer overflow because I wasn’t indexing my array correctly. The TAs explained why I was seeing “trash” as well as my input and then told me about the java trash collector but it didn’t really connect in my head then about how it was the memory being allocated. Experiment 2 helped me really understand the idea of memory and how it relates more to buffer overflows and how that can be used in attacks. Experiment 4 helped my understanding of how the attacks work and then how to mitigate against them since we capped the value of name at 32 which was the difference in memory between command and name and didn’t allow the users input to flow into command which could have been harmful to the device. I was surprised in part 3 when we printed out command to actually see that it flowed into its memory space and then set command to the overflow from name.

Works Cited

*Address Space Layout Randomization (ASLR)*, 2020, https://www.tutorialspoint.com/address-space-layout-randomization-aslr.