Matt McKillip
Section B
1/21/15

Lab 1: Linux Review

Part 1
This part was mostly review from my experience with Linux from previous labs and personal experience, but served as a nice refresher.

Part 2
I have never used GitHub before, so the walk through was very helpful. I knew what GitHub did and why it was used, but I had never used it before. So in this section I learned how to initialize a ssh connection, access lab files, and submit lab files. This section will be very helpful for me when figuring out GitHub in the next few labs.

Part 3
Again, GNU debugging is a tool I have never used before. I have used debuggers in other programs such as Eclipse IDE and Pycharm IDE. I have never used a debugger in a command line. After going through the tutorial, the debugger seemed very useful. The commands are very similar to the other debuggers I have used, just using command line instead of a GUI.

Part 4
I had never used Valgrind before. It ties in nicely to what we learned in lecture about heap and stack. I am still trying to learn the intricacies of the two. Valgrind seems to output a user friendly rundown of the memory allocations and memory problems.

Part 5
To debug this program, I first ran it normally and found out there was an error. Next I tried using Valgrind to narrow down the error and I found:

> "You entered the 5 character line matt
> ==3482== Invalid read of size 1
> ==3482== at 0x4006F7: main (rand_string.c:40)
> ==3482== Address 0x5 is not stack'd, malloc'd or (recently) free'd"

This output was very similar to the program in part 4. The invalid write of size 1 means that the program tired to write 1 byte past the end of the buffer.

Next I used the GNU debugging tool, ran the program and got the output:

> "Program received signal SIGSEGV, Segmentation fault.
> 0x00000000004006f7 in main (arg=1, argv=0x7fffffffe148) at rand_string.c:40
> 40 putchar(string[rand_array[i]]);
> Missing separate debuginfos, use: debuginfo-install glibc-2.12-1.149.el6_6.4.x86_64"

Then printed the backtrace to narrow down the problem

> (gdb) backtrace
> #0 0x00000000004006f7 in main (arg=1, argv=0x7fffffffe148) at rand_string.c:40
> (gdb) frame 0
> #0 0x00000000004006f7 in main (arg=1, argv=0x7fffffffe148) at rand_string.c:40
> 40 putchar(string[rand_array[i]]);

From this output I figured the issue was caused in the rand_array[i] section of the code, which was inside a for loop. Once opening the code I found that the for loop terminates at i<length which is causing the error. It should be fixed with i<=length.