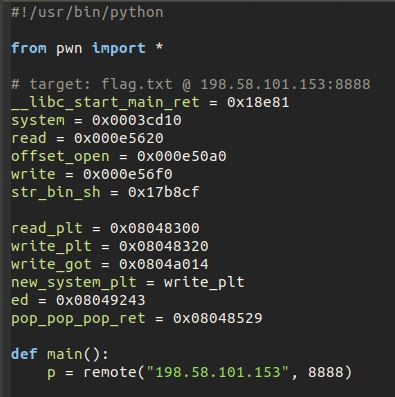
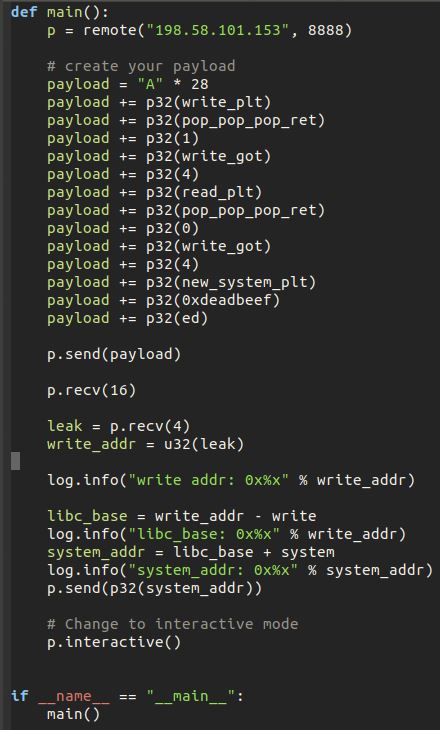
In this lab, I found that I needed to obtain the offsets and addresses as such things as the libc base, system, open, read, write, and the string bin/sh. Once I had these addresses, I also needed to obtain the read\_plt, write\_plt write\_got, ed, and pop pop pop ret addresses. Once I had all these I was able to build the attack script using them. All these addresses can be seen in the first code picture. Since this is a remote attack, I needed to define the process as a remote process, with the IP and Port the process is running on.



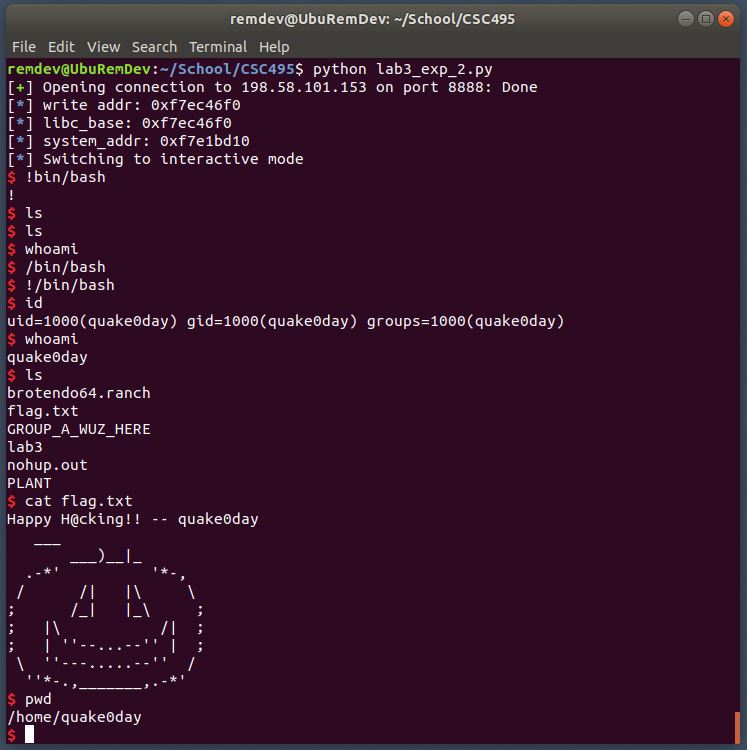
*This is the code from the first part of the attack script.*

The rest of the attack script was building the injection to get the exploit to behave in a way that allowed for a shell to be obtained through ed. By appending things in a correct order, it allows for the exploit to break through by acting in a specified way. By being able to correctly communicate with other programs by getting and sending proper memory addresses and their size, the attack can get to the point where it used the predefined memory locations as injections to get to inject ed and get access to a form of console.



*This is the code from the second part of the attack script.*

By running this attack script, it gives what looks like a console. This console requires “!/bin/bash” to be entered, to get a proper shell. This shell has the desired file in already in the same directory. I then used cat, to be able to view the contents. Once done, I could exit and ctrl+c to end the script.



*This is the console of the run of the attack.*