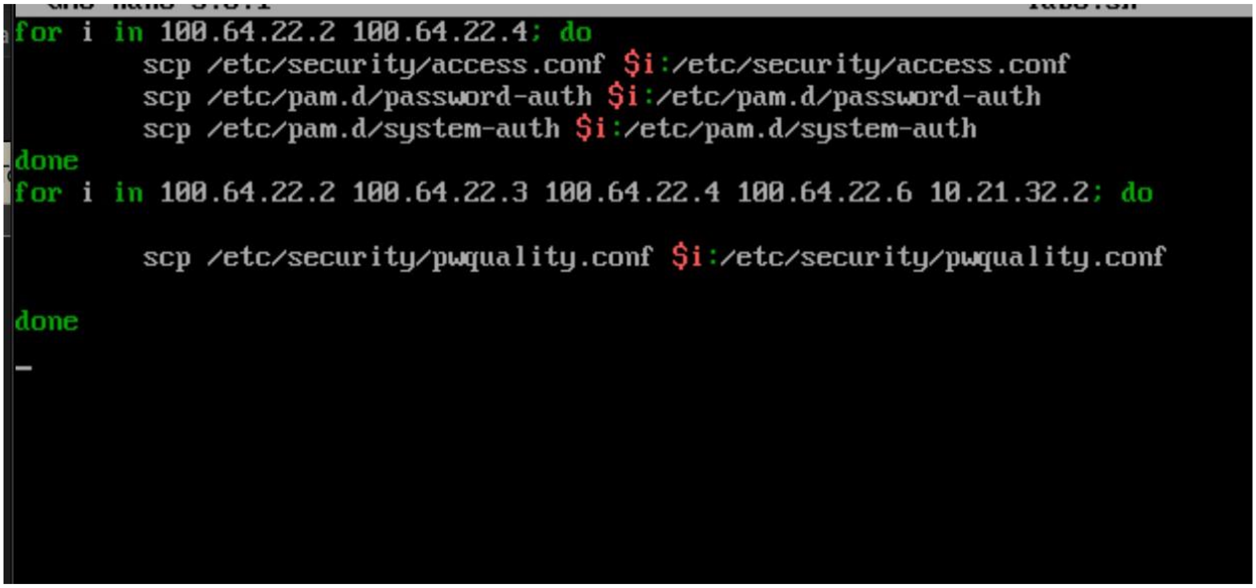
I added account required pam_access.so to /etc/pam.d/system-auth and /etc/pam.d/password-auth in machine A. Then, I allowed the required users one by one in the /etc/security/access.conf as such: +:root:ALL and disallowed all others as such: -:ALL:ALL. Then, I made a script to scp those three files to all the other Redhat machines except for machine E where all the users should be able to login. After that, I enabled pam_acces in the /etc/pam.d/system-auth and /etc/pam.d/password-auth as I did with machine A and the other Redhat machines. Then I edited /etc/security/access.conf by adding +:ALL:ALL to give all users login access. After that, I enabled pam access in Machine C by uncommenting account required pam_access.so in /etc/pam.d/login and /etc/pam.d/login. Then, I allowed the required users using the +:"user":ALL lines in /etc/security/access.conf in Machine C. Since I had trouble with scp from machine C to machine D I decided to go to Machine D and do the same things over there. I had to edit system-auth, password-auth, and access.conf. Furthermore, I went to machine E and edited those 3 files. The only thing that was different was allowing the accounting group as +:(accounting):ALL. For all the access.conf files I edited, I added -:ALL:ALL to disallow others to login.

For the password policy, first, I installed libpam-pwquality in the Debian machines. After that, I edited /etc/security/pwquality.conf on machine A. I set minlen to 10, lcredit to 0, dcredit to -2, ucredit to -2, and ocredit to -1. I added the (-) so that the length requirement is affirmed and no extra credits are given to any character. After that, I copied the file to all the other machines using scp.

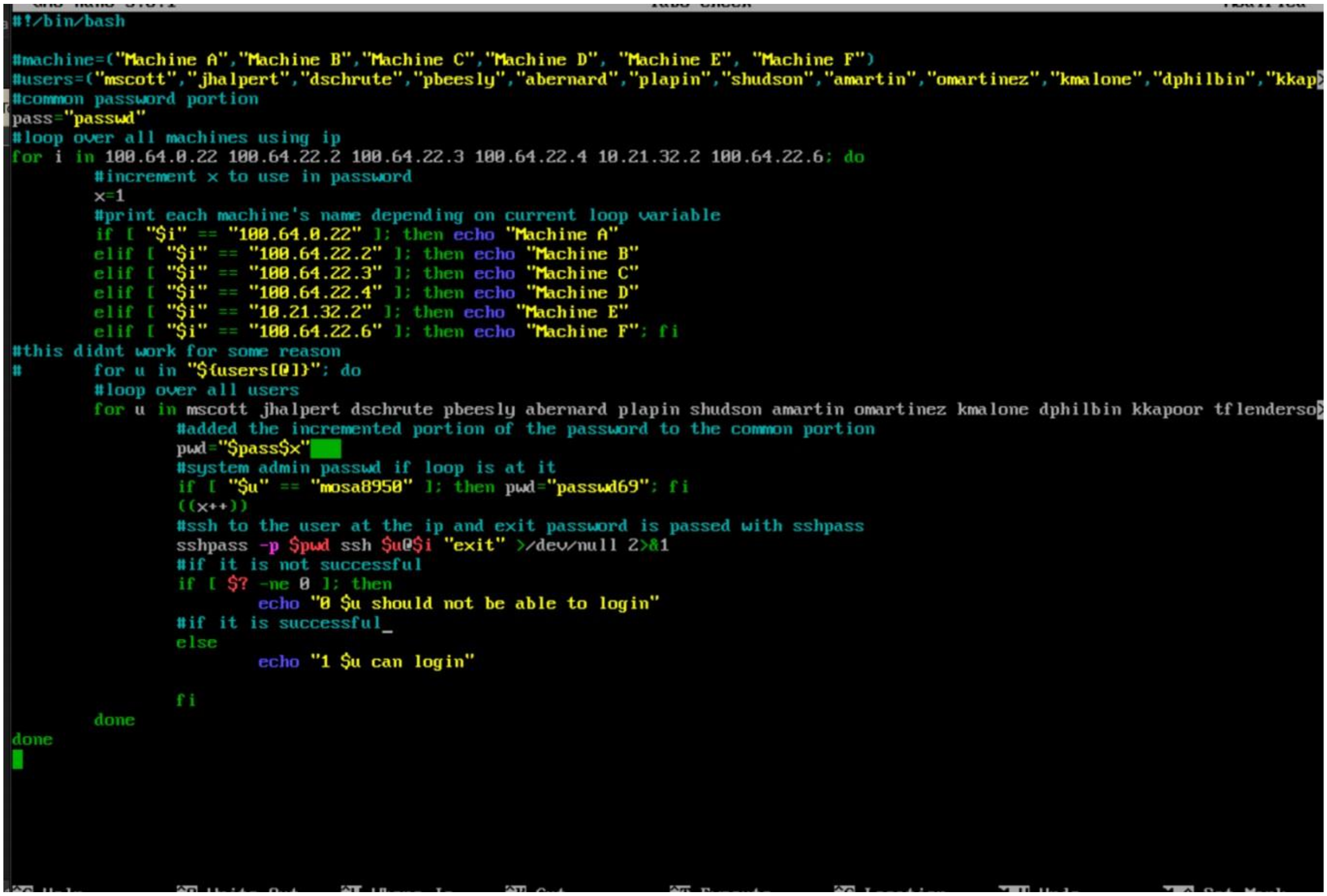
The extra credit was easy to implement since the user passwords are all similar. I made a variable for the mutual portion of the password. Then I created a loop on all the machines from A to F using the ip address to use it in ssh. In the loop I had if statements to print the machine's name depending on the variable of the loop. Next, I made an inner loop to go over all the users I set another password variable which is a concatenation of the mutual portion and the other remaining portion which is a variable that gets incremented. For the system admin, the password was different so I had an if statement to set the password as needed for this user. Then I used sshpass to pass the password into ssh and then performed the exit to check the success of the command in the current terminal. I checked the success using "$?". Then I printed the user with success or failure.



```
for i in 100.64.22.2 100.64.22.4; do
        scp /etc/security/access.conf $i:/etc/security/access.conf
        scp /etc/pam.d/password-auth $i:/etc/pam.d/password-auth
        scp /etc/pam.d/system-auth $i:/etc/pam.d/system-auth
done
for i in 100.64.22.2 100.64.22.3 100.64.22.4 100.64.22.6 10.21.32.2; do

        scp /etc/security/pwquality.conf $i:/etc/security/pwquality.conf

done
```



```
#!/bin/bash

#machine=("Machine A","Machine B","Machine C","Machine D", "Machine E", "Machine F")
#users=("mscott","jhalpert","dschrute","pbeesly","abernard","plapin","shudson","amartin","omartinez","kmalone","dphilbin","kkap>
#common password portion
pass="passwd"
#loop over all machines using ip
for i in 100.64.0.22 100.64.22.2 100.64.22.3 100.64.22.4 10.21.32.2 100.64.22.6; do
        #increment x to use in password
        x=1
        #print each machine's name depending on current loop variable
        if [ "$i" == "100.64.0.22" ]; then echo "Machine A"
        elif [ "$i" == "100.64.22.2" ]; then echo "Machine B"
        elif [ "$i" == "100.64.22.3" ]; then echo "Machine C"
        elif [ "$i" == "100.64.22.4" ]; then echo "Machine D"
        elif [ "$i" == "10.21.32.2" ]; then echo "Machine E"
        elif [ "$i" == "100.64.22.6" ]; then echo "Machine F"; fi
#this didnt work for some reason
#       for u in "${users[@]}"; do
        #loop over all users
        for u in mscott jhalpert dschrute pbeesly abernard plapin shudson amartin omartinez kmalone dphilbin kkapoor tflenderso>
                #added the incremented portion of the password to the common portion
                pwd="$pass$x"
                #system admin passwd if loop is at it
                if [ "$u" == "mosa8950" ]; then pwd="passwd69"; fi
                ((x++))
                #ssh to the user at the ip and exit password is passed with sshpass
                sshpass -p $pwd ssh $u@$i "exit" >/dev/null 2>&1
                #if it is not successful
                if [ $? -ne 0 ]; then
                        echo "0 $u should not be able to login"
                #if it is successful
                else
                        echo "1 $u can login"

                fi
        done
done
```