Sean Poston

Dr. Mitra

CY201 Lab 02

https://repl.it/@seanposton4/CY201Lab2

Q2:

```
    Enter the   password    :
bytebytebytebytebyte
98
    Wrong Password

    Root privileges given to the user
```

Obviously, this isn't the desired result. If a user inputs too many characters for the password, it will cause a buffer overflow that causes the "pass" flag to flip and allow root privileges for the user. There are a couple things that can make this more secure.

Q3:

```
26      else {
27          printf ("\n Correct Password    \n");
28          localinfo.pass = 1; // Set a flag denoting correct pass
29
30          if(localinfo.pass){ /*  Now give root or admin right user*/
31              printf ("\n Root privileges given to the user \n");
32          }
33      }
```

One is to move the root privileges statement inside of the else statement with the "Correct Password" line. You don't want to give root privileges without receiving a proper password, so it doesn't make sense for the root privileges to be separate from the correct password.

```
17      printf("\nEnter the password: \n");
18      scanf("%19s", localinfo.buff); //password input
```

Another move to secure this more, is to limit the input of the user, as to avoid segmentation faults. scanf is much more secure than gets, and it allows for the limiting of input values.

```
4   #define BUFFERSIZE 20
5
6   int main(void)  {
7       //  Use a   struct to   force   local   variable    memory  ordering
8       struct {
9       char buff[BUFFERSIZE];
10      char pass;
11      } localinfo;
```

Finally, just to help matters, we can increase the size of the buffer to help avoid users accidentally putting in too many characters. This isn't necessary after the other steps are taken, but it's still helpful.

Input Tests:

```
Enter the password:
byt
-101
    Wrong Password
```

```
Enter the password:
asdfrtyufhghfg
-1
    Wrong Password
```

```
Enter the password:
bytebytebyte
98
    Wrong Password
```

One more for good measure:

```
Enter the password:
al;kjfdsl;kjafsdlkj;fadskl;jfadskjl;fdasklj;fdsajkl;fsdajkl;fdsjkla;fskdljafjdksla;fdsjkl;afjkdlsa;afklj;dskjl;fasdjkl;fasdkjl;fsda;lkasdfjlk;asjdflkjkjaslkfdjalskdjfla;ksjdf
-1
    Wrong Password
```

Correct Password:

```
Enter the password:
byte
0
    Correct Password

    Root privileges given to the user
```