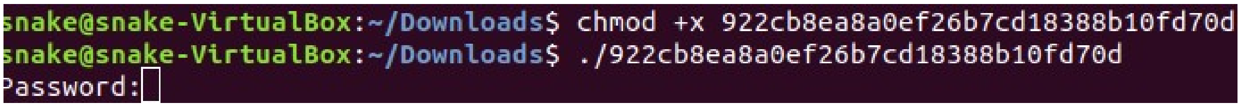
1) Once you obtained the ELF binary from the website, change it to an executable by doing this linx command “**CHMOD +X 922cb8ea8a0ef26b7cd18388b10fd70d**”

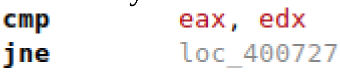
2) Then executing by the dot forward slash.



```
snake@snake-VirtualBox:~/Downloads$ chmod +x 922cb8ea8a0ef26b7cd18388b10fd70d
snake@snake-VirtualBox:~/Downloads$ ./922cb8ea8a0ef26b7cd18388b10fd70d
Password:
```

3) You will notice it asking for a password. Let’s opening in a dissembler! I will be using the hopper because of its decompiler and nice GUI!

4) Take a look its main function then all the way down its to cmp instruction before it makes it jump to the other functions.



```
cmp         eax, edx
jne         loc_400727
```

5) The instruct test if edx is equal to eax, so lets manipulate it in debugger named GDB!

6) At the same time, we have to find the address of the instruction. So click on the mov/add button on hopper. mov add

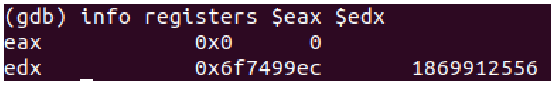
7) You will see that the cmp function has the address 0x04006ef.

8) Knowing the address, run GDB and input any input and it will say that it is wrong. So this lets the operating system know the virtual addresses of the executable.

9) After running it the second time, set a breakpoint on the address we found earlier as so

“b *0x04006ef”

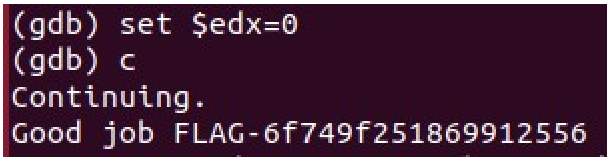
10) Run it again and print the register value: “**info register $eax $edx”**



```
(gdb) info registers $eax $edx
eax            0x0      0
edx            0x6f7499ec       1869912556
```

11) Notice that eax is zero, since eax is the destination register. We have to set edx to zero as so : **“set $edx=0”**

12) Woo got the flag!



```
(gdb) set $edx=0
(gdb) c
Continuing.
Good job FLAG-6f749f251869912556
```