HGAME-Week3-Writeup

Re

1.FAKE

拖到ida

```
puts("Give me your true flag:");
__isoc99_scanf("%50s");
if ( strlen(s) != 36 )                          // 长度
{
  puts("Wrong length.");
  exit(0);
}
for ( i = 0; i <= 35; ++i )
  v4[i] = s[i];
if ( sub_401216(v4) == 1 )                      // 关键函数
  puts("Ohhhhhhhhhh!");
else
  puts("Wrong flag. Keep looking!");
return 0LL;
```

关键函数里有36个方程，36个变量，那就不可能普通的解出来，

查找字符串得到提示 **Try angr or z3**，我用的是 z3库，脚本太长了就不放了（用z3库解的时候一直跑不出结果，就去问了 r3n0学长，学长说可能需要点时间，但之后我重新弄了一下add的部分很快就跑出来了，应该是我自己中间操作上的问题....）

ASCII转换到 ASCII (例: a b c)

```
hgame{@_FAKE_flag!-do_YOu_know_SMC?}
```

添加空格 删除空格 将空白字符转换

十六进制转换到16进制(例:0x61或61或61/62) 删除 0x

```
0x680x670x610x6d0x650x7b0x400x5f0x460x410x4b0x450x5
f0x660x6c0x610x670x210x2d0x640x6f0x5f0x590x300x750x
5f0x6b0x6e0x6f0x770x5f0x530x4d0x430x3f0x7d
```

十进制转换到 10进制 (例: 97 98 99)

```
104103971091011236495706575699510210897103334510011
195894811795107110111119958377676312 5
```

二进制转换到 2进制(例:01100001 01100010 01100011)

```
01101000011001110110000101101101011001010111011 0
10000000101111101000110010000010100101101000101 01
01111011001100110110001100001011001110010000100 1
01101011001000110111101011111010110010011000001 11
```

假flag，查一下**SMC**

SMC(self-Modifying Code)，就是在真正执行某一段代码时，程序会对自身的该段代码进行自修改，只有在修改后的代码才是可汇编，可执行的。在程序未对该段代码进行修改之前，在静态分析状态下，均是不可读的字节码，IDA之类的反汇编器无法识别程序的正常逻辑。是一种反调试代码技术。

在ida里找一下smc自解密的部分

```
__int64 sub_40699B()
{
  __int64 result; // rax
  unsigned int i; // [rsp+Ch] [rbp-4h]

  mprotect(&dword_400000, 0x10000uLL, 7);
  for ( i = 0; ; ++i )
  {
    result = i;
    if ( i > 0x43E )
      break;
    *(sub_401216 + i) ^= byte_409080[i];
  }
  return result;
}
```

这段代码执行后才得到真正的代码，写 idc 脚本来解密 ida 反编译后错误的代码

```
#include <idc.idc>

static main()
{
    auto addr = 0x00401216; //函数地址
    auto addr2 = 0x00409080;    //byte数组地址
    auto i = 0;
    for(i=0;i<=0x43E;i++)
    {
        PatchByte(addr+i,Byte(addr+i)^Byte(addr2+i));
    }
}
```

得到运行过程中真正的代码

```
 79  v2[29] = 119;
 80  v2[30] = 95;
 81  v2[31] = 83;
 82  v2[32] = 77;
 83  v2[33] = 67;
 84  v2[34] = 63;
 85  v2[35] = 125;
 86  v10 = 1;
 87  for ( i = 0; i <= 5; ++i )
 88  {
 89    for ( j = 0; j <= 5; ++j )
 90    {
 91      for ( k = 0; k <= 5; ++k )
 92        v4[6 * i + j] += v2[6 * k + j] * *(4LL * (6 * i + k) + a1);
 93    }
 94  }
 95  for ( l = 0; l <= 5; ++l )
 96  {
 97    for ( m = 0; m <= 5; ++m )
 98    {
 99      if ( v4[6 * l + m] != v3[6 * l + m] )
100        v10 = 0;
101    }
102  }
103  return v10;
104}
```

这部分继续用z3库写脚本去解

```
from z3 import *

s = Solver()
a1 = [0] * 36
for i in range(36):
    a1[i] = Int('a1[' + str(i) + ']')
v3 = [0]*36
v2 = [0]*36
v4 = [0]*36
v3[0] = 55030
v3[1] = 61095
v3[2] = 60151
v3[3] = 57247
v3[4] = 56780
v3[5] = 55726
v3[6] = 46642
```

```
v3[7] = 52931
v3[8] = 53580
v3[9] = 50437
v3[10] = 50062
v3[11] = 44186
v3[12] = 44909
v3[13] = 46490
v3[14] = 46024
v3[15] = 44347
v3[16] = 43850
v3[17] = 44368
v3[18] = 54990
v3[19] = 61884
v3[20] = 61202
v3[21] = 58139
v3[22] = 57730
v3[23] = 54964
v3[24] = 48849
v3[25] = 51026
v3[26] = 49629
v3[27] = 48219
v3[28] = 47904
v3[29] = 50823
v3[30] = 46596
v3[31] = 50517
v3[32] = 48421
v3[33] = 46143
v3[34] = 46102
v3[35] = 46744
v2[0] = 104
v2[1] = 103
v2[2] = 97
v2[3] = 109
v2[4] = 101
v2[5] = 123
v2[6] = 64
v2[7] = 95
v2[8] = 70
v2[9] = 65
v2[10] = 75
v2[11] = 69
v2[12] = 95
v2[13] = 102
v2[14] = 108
v2[15] = 97
v2[16] = 103
v2[17] = 33
v2[18] = 45
v2[19] = 100
v2[20] = 111
v2[21] = 95
v2[22] = 89
v2[23] = 48
v2[24] = 117
v2[25] = 95
v2[26] = 107
v2[27] = 111
v2[28] = 110
```

```
v2[29] = 119
v2[30] = 95
v2[31] = 83
v2[32] = 77
v2[33] = 67
v2[34] = 63
v2[35] = 125

for i in range(0,6):
    for j in range(0,6):
        for k in range(0,6):
            v4[6 * i + j] += v2[6 * k + j] * a1[6 * i + k]

for i in range(0,6):
    for j in range(0,6):
        s.add(v4[6*i+j] == v3[6 * i + j])

s.check()
result = s.model()

for i in range(0, 36):
     print(result[a1[i]], end=' ')
```

Crypto

1.LikiPrime

还是 RSA，不过素数 N 更大了，但是放到网站里分解还是能分解出 q和 p，之后就是用上周的脚本跑出flag

2.HappyNewYear!!

e = 3，发送内容相同，自然就想到低指数广播攻击，但是广播攻击只需要 e组数据，那么猜测给的数据里解出来的明文不是全部相同。

低指数广播攻击需要用到中国剩余定理（孙子定理），脚本如下

```
from libnum import n2s
import gmpy2
from functools import reduce

def modinv(a, m):
    return int(gmpy2.invert(gmpy2.mpz(a), gmpy2.mpz(m)))


def chinese_remainder(n, a):
    sum = 0
    prod = reduce(lambda a, b: a * b, n)
    for n_i, a_i in zip(n, a):
        p = prod // n_i
        sum += a_i * modinv(p, n_i) * p
    return int(sum % prod)

n1 =
```

```
c1 =

n2 =
c2 =

n3 =
c3 =

n4 =
c4 =

n5 =
c5 =

n6 =
c6 =

n7 =
c7 =

nset = [n2,n4,n6]
cset = [c2,c4,c6]

m = chinese_remainder(nset, cset)
m = gmpy2.iroot(m,3)
m = int(m[0])
print(n2s(m))
```

c1,c3,c5 解密得到一个片段，c2,c4,c6解密得到剩下的flag片段（本来写的是随机在给出的7组数据里挑三组解密，但是没跑出结果，后来手动试了一下试出来了）

```
b'I am afraid the dishes in the second grade are too fragrant, you will not reply my text messages, \nso I won't give yo
u New Year greetings this year, I hope you don't know how to praise, good night.\n\nhgame{!f+yOu-pl4y_rem"
>>>
```

```
>>> m = int(m[0])
>>> print(n2s(m))
b'Hello Liki4:\n\nI am afraid that there are too many blessings on the 30th night, you will not see my greetings, \nI am
 afraid that the firecrackers in the first grade are too noisy, you will not hear my blessings, \n\ngind3r_YOu 9ot=i7}'
>>>
```

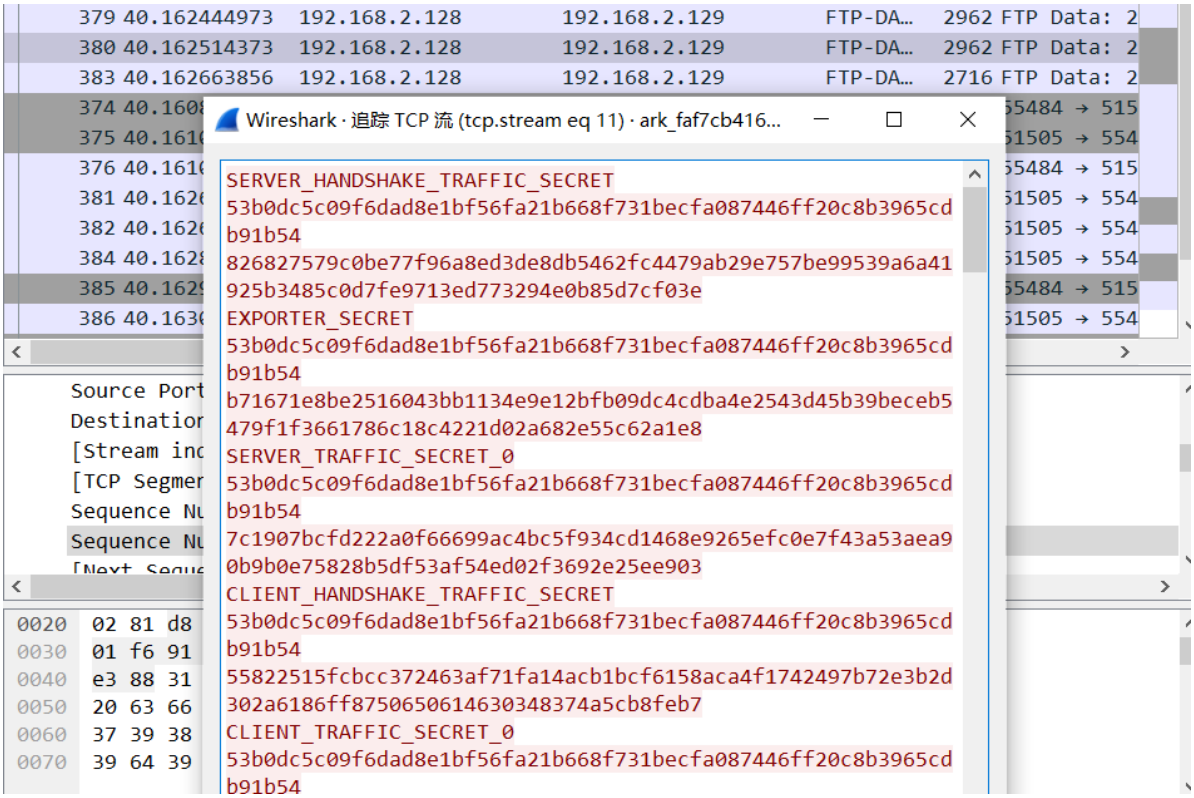
Misc

1.A R K

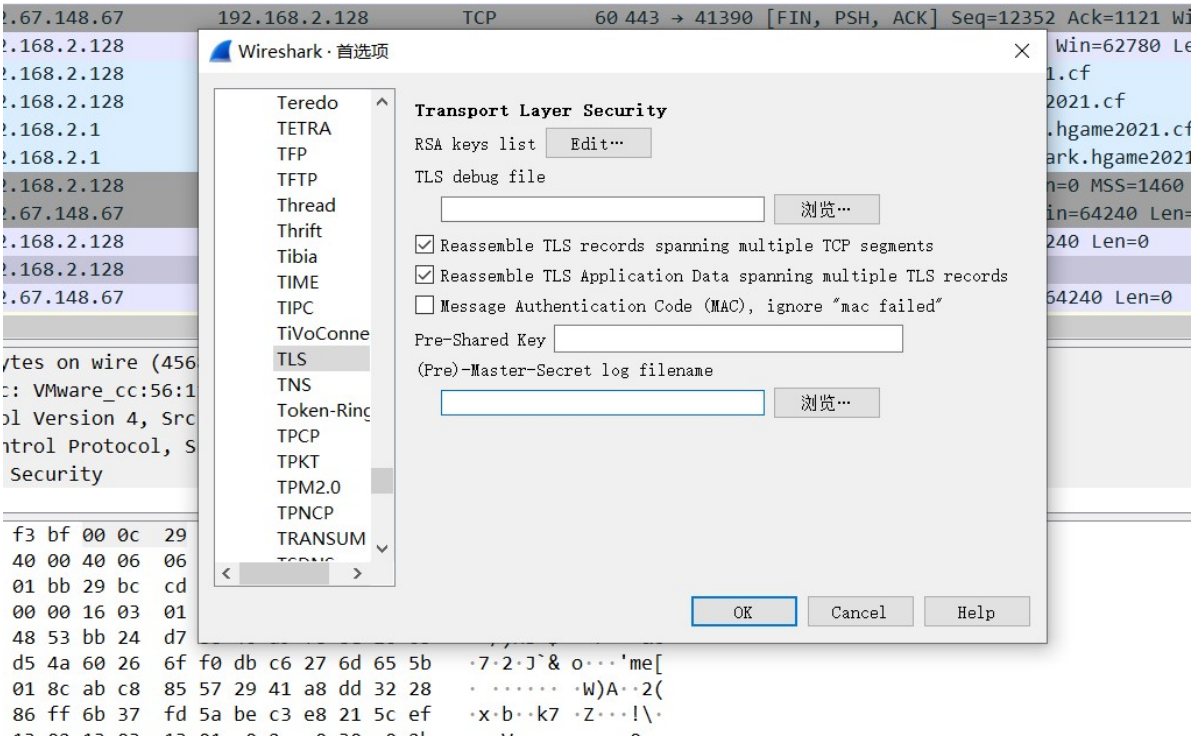
打开下载的流量包，首先看到dns里的域名

1	6.002364746	192.168.2.128	192.168.2.1	DNS	76 Standard query 0x9155 AAAA ark.hgame2021.cf
2	6.002511732	192.168.2.1	192.168.2.128	DNS	140 Standard query response 0x7769 A ark.hgame2021.cf A 172.67.14…
3	6.002511772	192.168.2.1	192.168.2.128	DNS	76 Standard query response 0x9155 AAAA ark.hgame2021.cf
4	6.002840830	192.168.2.128	172.67.148.67	TCP	74 41392 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 …
5	6.003483666	172.67.148.67	192.168.2.128	TCP	60 443 → 41392 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
6	6.003554338	192.168.2.128	172.67.148.67	TCP	54 41392 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
7	6.010562959	192.168.2.128	172.67.148.67	TLSv1.3	571 Client Hello
8	6.010864614	172.67.148.67	192.168.2.128	TCP	60 443 → 41392 [ACK] Seq=1 Ack=518 Win=64240 Len=0

刚开始以为跟网址有关，但后来题目里说 “**本题目所有解题操作均只用流量，与网址无关**”，那就接着看其他包，查资料知道 TLS 加密尝试找到流量包里的证书



导出原始数据保存为 ssl.log ,再重新导入解密



找到解密后的 HTTP 流，导出后得到一堆json文件，导出来后卡了一段时间，之后看到题目的补充信息

补充说明：明日方舟是一款塔防游戏，可以将可部署单位放置在场地中。并且具有自律功能，可以记录部署的操作。

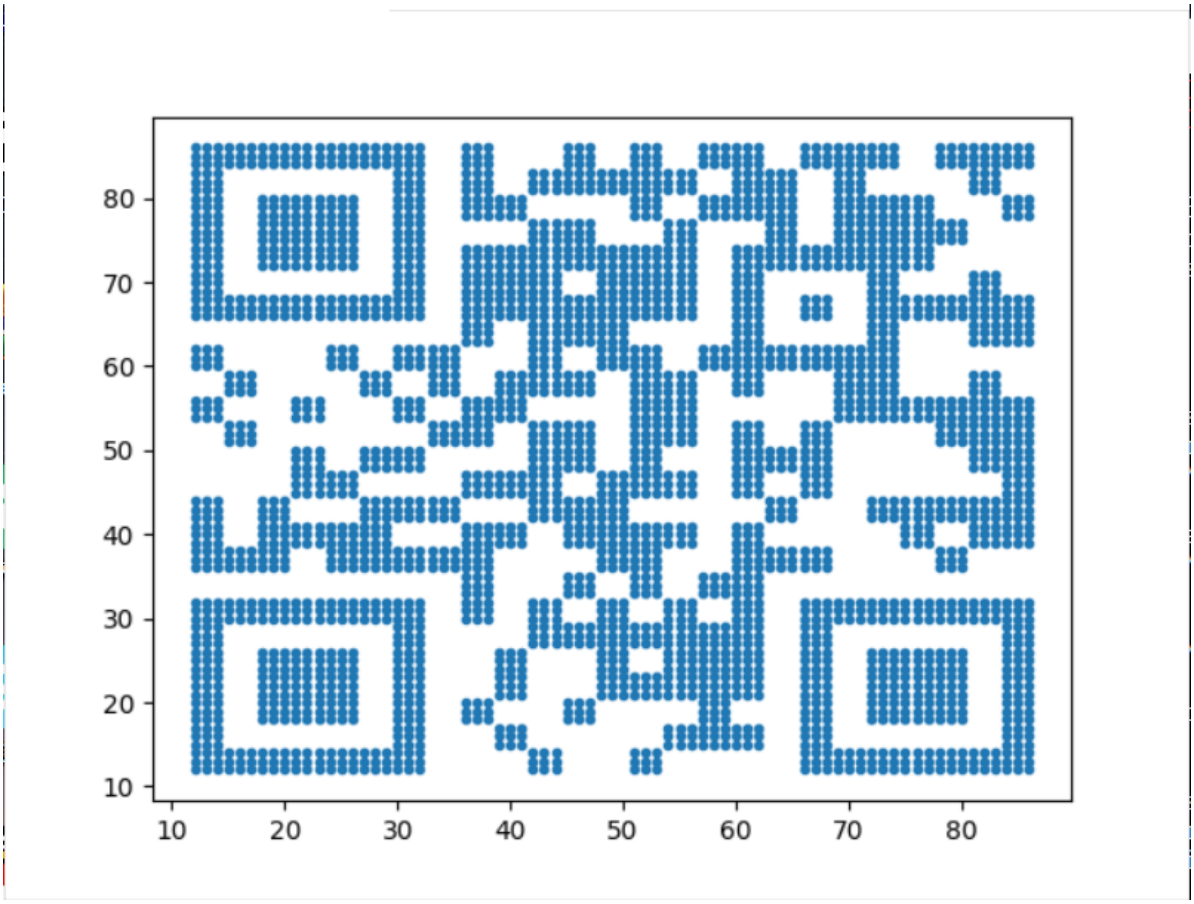
翻译：没用 没用 出题人用可部署单位画了个东西 背景是白色的

记录部署操作？！那就打开 getBattleReplay 文件，看到像是 base64 编码的字符串，解码看到 zip 文件头标识另存为 zip，解压打开发现配对的 row 和 col ，应该是部署的坐标，再根据题目里的 **画了个东西** 猜测这些坐标是二维码黑色部分，处理一下再运行脚本

```
import matplotlib.pyplot as plt
import numpy as np

x, y = np.loadtxt('./xy.txt', delimiter=',', unpack=True)
plt.plot(x, y, '.')
plt.show()
```

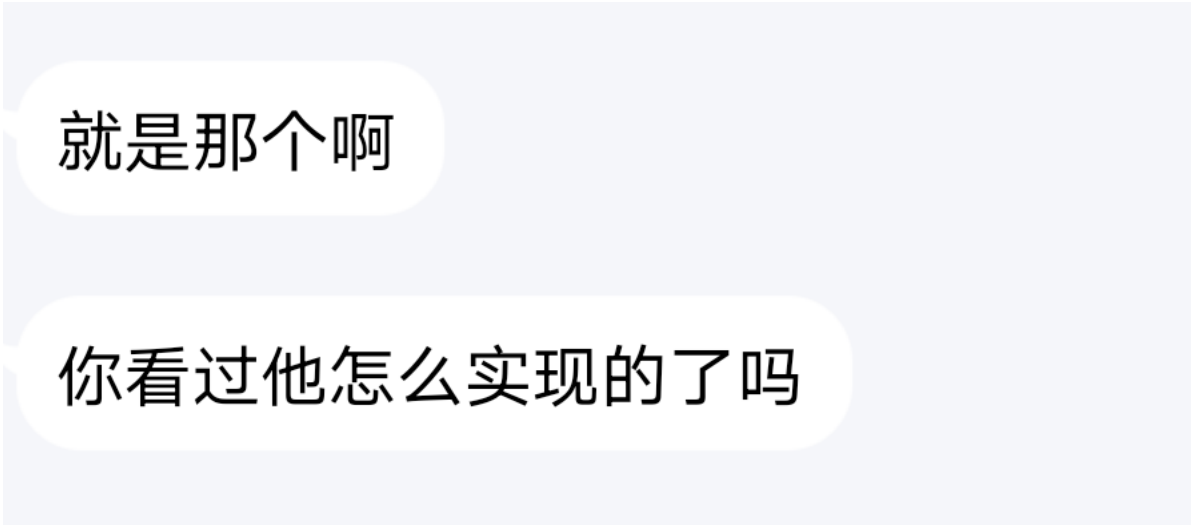
扫码得flag



2.A R C

压缩包加密了，那就先看图片找密码

刚开始误以为 BV 是某种 CTF 中的加密方式，在网上搜索了一圈没有收获，于是去问 Akira 学长 BV 除了 b站BV号还有什么意思，学长的答复是



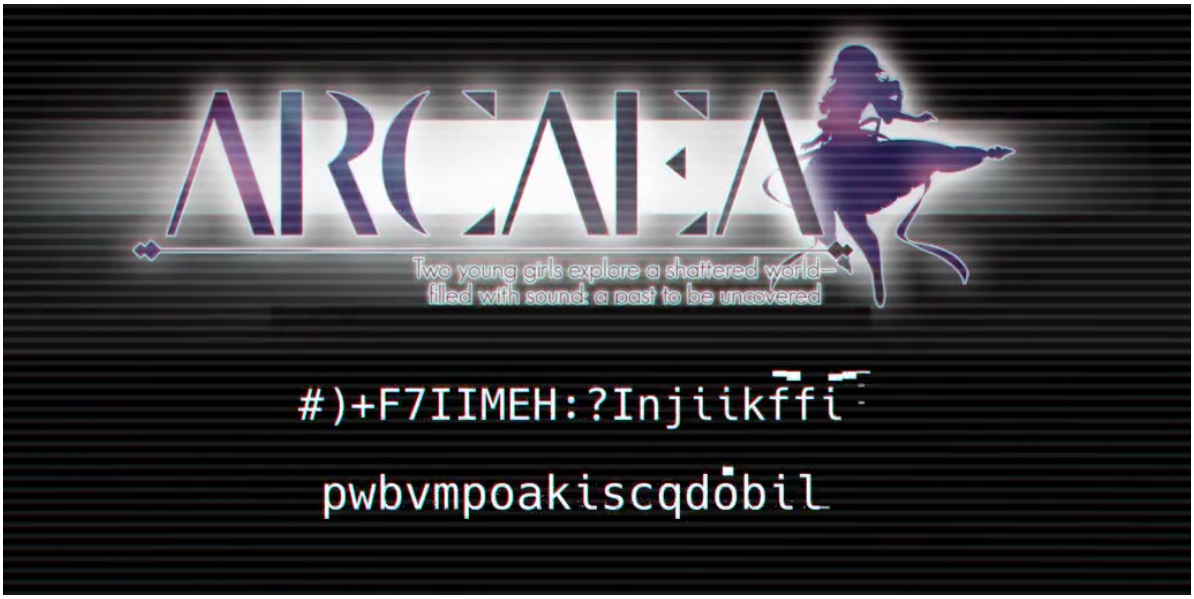
怎么实现的？查 av号怎样转 bv号后发现 https://www.zhihu.com/question/381784377/answer/1099438784 这个回答里的实现方法，根据 **hint1：8558 应该理解成85和58，BV号_是所以图片里的是__** 先对图里的密文进行 base85 解密

```
>>> import base64
>>> c = "BK0ICG]Qr*88_$gC,'-j2+KH86?Q\"%928;LG@O*!Am0+`;E7iV2agSE<c'U;6Yg^#H?!YBAQ]"
>>> m = base64.a85decode(c)
>>> print(m)
b'h8btxsWpHnJEj1aL5G3gBuMTKNPAwcF4fZodR9XQ7DSUVm2yCkr6zqiveY'
>>>
```

根据长度判断应该是 av号转 bv号算法里的table，那就换一下上面链接里代码的table，密文是压缩包名里的数字，跑一下得到压缩包密码

```
>>> table = 'h8btxsWpHnJEj1aL5G3gBuMTKNPAwcF4fZodR9XQ7DSUVm2yCkr6zqiveY'
>>> tr = {}
>>> for i in range(58):
...     tr[table[i]] = i
...
>>> s = [11, 10, 3, 8, 4, 6]
>>> xor = 177451812
>>> add = 8728348608
>>>
>>> def dec(x):
...     r = 0
...     for i in range(6):
...         r += tr[x[s[i]]] * 58 ** i
...     return (r - add) ^ xor
...
>>>
>>> def enc(x):
...     x = (x ^ xor) + add
...     r = list('BV1  4 1 7  ')
...     for i in range(6):
...         r[s[i]] = table[x // 58 ** i % 58]
...     return ''.join(r)
...
>>> print(enc(10001540))
BV17f411J77h
>>>
```

打开后得到一段 arc 视频和密文，用题目里给的软件播放视频得到

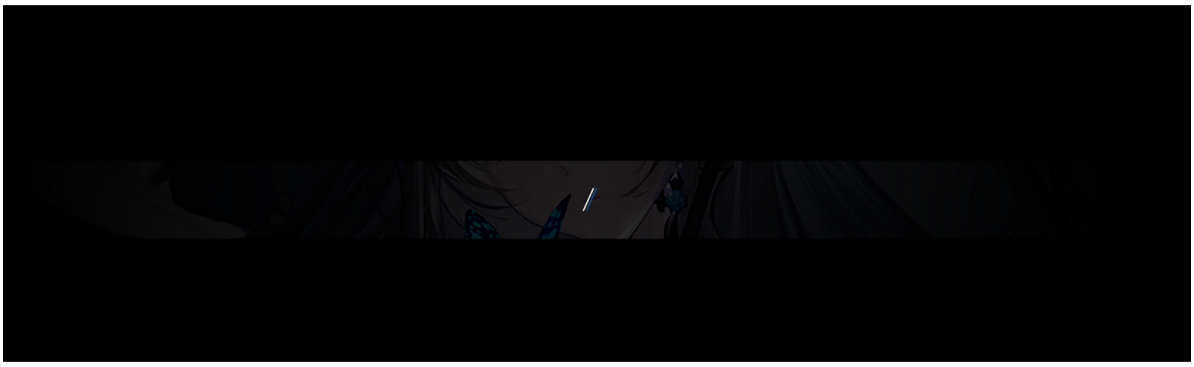


发现图中第一行文字和文本里文字相似，猜测是同种加密方式，之后又没有思路了，hint2里提到 **词频分析是个好东西，别忘了视频里的问题**，查了下视频里的问题但没什么发现。后来又放出hint **用了某种ROT的范围，但是位移不一样**，根据两个字符的应该是 is 之类的尝试将每个字符后移10位，得到了一段英文，然后又卡住了。于是又去问 Akira学长，学长说并不是后移10位提示我再想想视频里问题的答案，突然想起之前有查到 42 这个数字，尝试后移42位得到正确的文字

```
Flag is not here  but I write it because you may need more words to analysis what encoding the linel is.    For line2  L
iki has told you what it is  and Akira is necessary to do it.   Two young girls explore  shattered world  filled with s
ound: a past to be uncovered...  Each awakens in this blank  ruin-dotted world to discover that she is equally blank  re
membering nothing of what came before.  And then they make a second discovery: the Arcaea  multitudes of floating glass-
like shards containing vivid memories of the past.  MSUpasswordis:6557225  <C.B9<;-75?/=0;.58
```

视频里文字第一行也是后移 42位得到 MSU 密码，解密得到的文字里提到视频里第二行的加密方法 Liki学长提到过，根据 week1 crypto的第一题，第二行文字加密方法应该是维吉尼亚，密钥是 Akira

在 virtualDub2里安装MSUstego插件，用之前得到的密码对视频进行解密，得到网址、光和对立的名字，访问网站输入用户名和密码



根据hint5: **/ 不是可输入的意思，是网站路径**，想起之前视频解出的第二行还没用上，那就在网站路径后输入解出的明文，得到flag（pm）

这周misc虽然难但很有趣，梗挺多的

总结：

这周因为跟父母回老家拜年少了很多做题的时间，之后电脑还出了故障重装了系统，下周要专注于re了，不能再靠着 misc 和 crypto 上分了

接下来是没做出来的部分

re

Gun这道题用查壳工具查得是梆梆免费版加固，想要脱壳，用了很多网上查到的方法但都失败了，问了 Trotsky学长，推荐使用frida-dexdump脱壳，但是因为这道题需要root过的真机（模拟器不行）来进行脱壳，所以最后只能放弃

hello_re3我没太搞明白，拖进ida发现应该都是调试信息，mezone学长说要用dbgview看调试信息再定位到程序里，可惜这题是最后一天才做的，最后没时间了，打算看官方writeup再学习一下

Crypto

EncryptedChats这题尝试了各种方法去解，还是解不出 a 和 b ，需要的时间都很长，但是 g 挺大（不是2、5这种很小的） a 跟 b 应该是能解出来的...