领航杯部分Writeup

原创

LetheSec 于 2019-11-25 09:08:47 发布 2284 收藏 4

分类专栏： wp 文章标签： wp

编码解码

```
V2VsY29tZXRvdGhlUGlsb3RDdXA=
```

签到题，base64解码一下得到：

```
WelcometothePilotCup
```

取证分析

给了一个pacp.pcap，CTF Wiki里面收录过这一道例题…

wireshark 打开发现全部为DNS协议,查询名为大量字符串 `xxxxx.skullseclabs.org`

好像是BSides San Francisco CTF 2017原题，参考这篇文章：https://volatilevirus.home.blog/2018/12/30/bsidessf17-ctf-dnscap-write-up/

先用下面命令进行提取：

```
tshark -r pacp.pcap -T fields -e dns.qry.name > hex
```

然后用如下脚本得到flag.png:

```
from scapy.all import *

r = rdpcap("pacp.pcap")

a = ""

b = ""
c = ""
new = ""

f = open("flag.png","w")

for i in range (0,len(r)):
    if r[i].haslayer(DNSQR) and not r[i].haslayer(DNSRR):
        a = r[i][DNSQR].qname
        b = a.replace(".skullseclabs.org.","")
        b = b.replace(".","").decode("hex")[9:]

        if b == c:
            continue

        c = b

        if 6 < i <365:
            new = new + b

f.write(new)

f.close()
```

直接打不开，用010 Editor看一下：

编辑方式: 十六进制(H) 运行脚本 运行模板

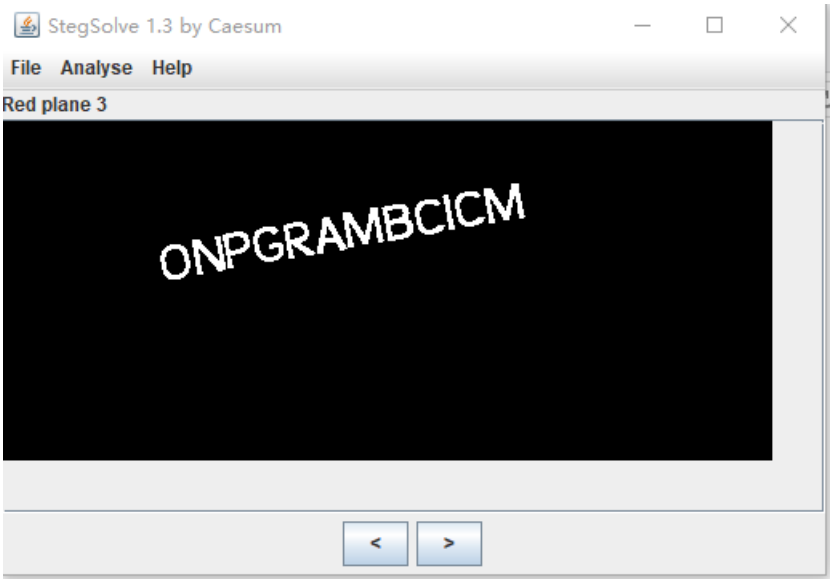
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	57	65	6C	63	6F	6D	65	20	74	6F	20	64	6E	73	63	61	Welcome to dnsca
0010h:	70	21	20	54	68	65	20	66	6C	61	67	20	69	73	20	62	p! The flag is b
0020h:	65	6C	6F	77	2C	20	68	61	76	65	20	66	75	6E	21	21	elow, have fun!!
0030h:	0A	63	6F	6D	6D	61	6E	64	20	28	73	69	72	76	69	6D	.command (sirvim
0040h:	65	73	29	00	00	00	2C	ED	80	01	00	03	89	50	4E	47	es)...,í€...‰PNG
0050h:	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	00	00	01	00	........IHDR....
0060h:	00	00	01	00	08	04	00	00	00	F6	7B	60	ED	00	00	00	.........ö{`í...
0070h:	04	67	41	4D	41	00	01	86	A0	31	E8	96	5F	00	00	00	.gAMA..† 1è–_...
0080h:	02	62	4B	47	44	00	FF	87	8F	CC	BF	00	00	00	09	70	.bKGD.ÿ‡.Ì¿....p
0090h:	48	59	73	00	00	0B	13	00	00	0B	13	01	00	9A	9C	18	HYs.........šœ.
00A0h:	00	00	00	07	74	49	4D	45	07	E1	02	02	05	0D	35	24	....tIME.á....5$
00B0h:	D3	81	E9	00	00	2C	08	49	44	41	54	78	DA	ED	9D	77	Ó.é..,.IDATxÚí.w
00C0h:	9C	1B	D5	D5	F7	BF	A3	AE	95	56	DB	AB	D7	DE	5D	7B	œ.ÕÕ÷¿£®•VÛ«×Þ]{
00D0h:	77	ED	75	B7	71	C1	D8	98	66	C0	38	26	C6	80	C1	0F	wíu·qÁØ˜fÀ8&Æ€Á.
00E0h:	09	84	04	48	42	42	48	F2	52	9E	87	84	27	A4	40	EA	.„.HBBHòRž‡„'¤@ê
00F0h:	43	78	52	48	21	8D	BC	40	48	42	89	43	31	25	74	B0	CxRH!.¼@HB‰C1%t°
0100h:	81	60	03	2E	B8	E0	EE	B5	D7	F6	F6	5E	D4	E6	F9	63	.`..¸àîµ×öö^Ôæùc
0110h:	65	79	46	1A	69	34	5A	49	33	32	FC	E6	83	B1	2C	8D	eyF.i4ZI32üæƒ±,.
0120h:	74	E7	DE	DF	3D	E7	DC	73	CE	3D	57	E0	E4	82	05	7B	tçÞß=çÜsÎ=Wàä‚.{
0130h:	F8	72	50	40	31	F9	14	90	4F	3F	F9	14	F0	C1	8F	03	ørP@1ù..O?ù.ðÁ..

删掉上面选中的部分，再打开，得到flag：b91011fc

lsb

给了下面这张图片：

根据题目名字lsb，用Stegsolve打开，移位到下面位置得到flag：ONPGRAMBCICM



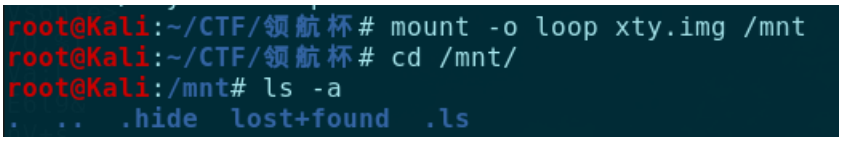
恢复与解密

题目描述：公安人员在犯罪分子的电脑中发现一些磁盘文件，但是发现关键信息已经被删除，现需要你对磁盘进行恢复，并对恢复出来的一些秘密文件里面的加密信息进行解密。 注意：通过strings获取到的yc4pl0fvjs2k1t7T为假flag，请尝试使用其他正确的做题方式获取flag。

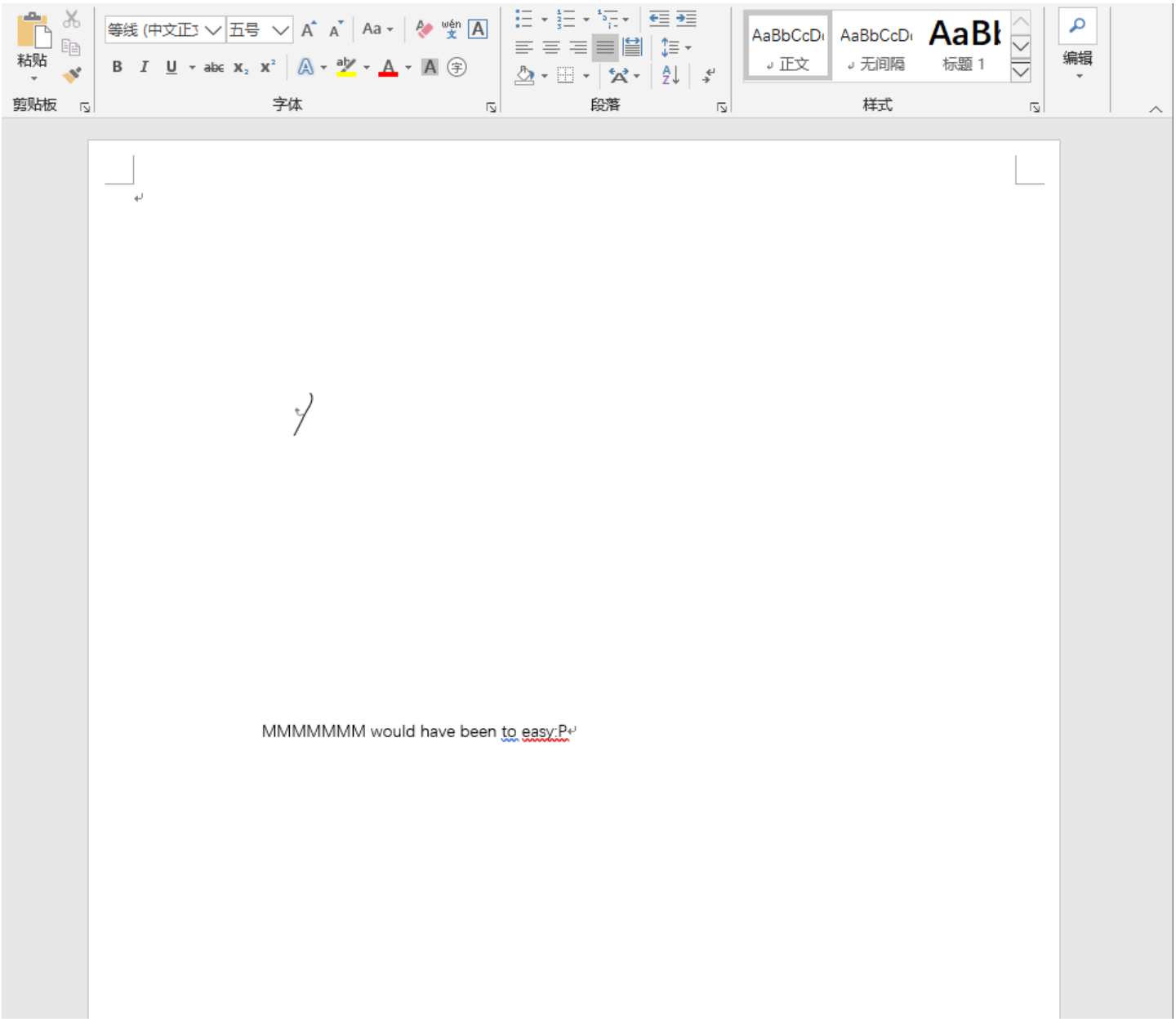
所以可以判断这题为磁盘恢复，并且有假flag。

附件给的是一个xty.img，在kali下面命令进行挂载：

```
mount -o loop xty.img /mnt
```

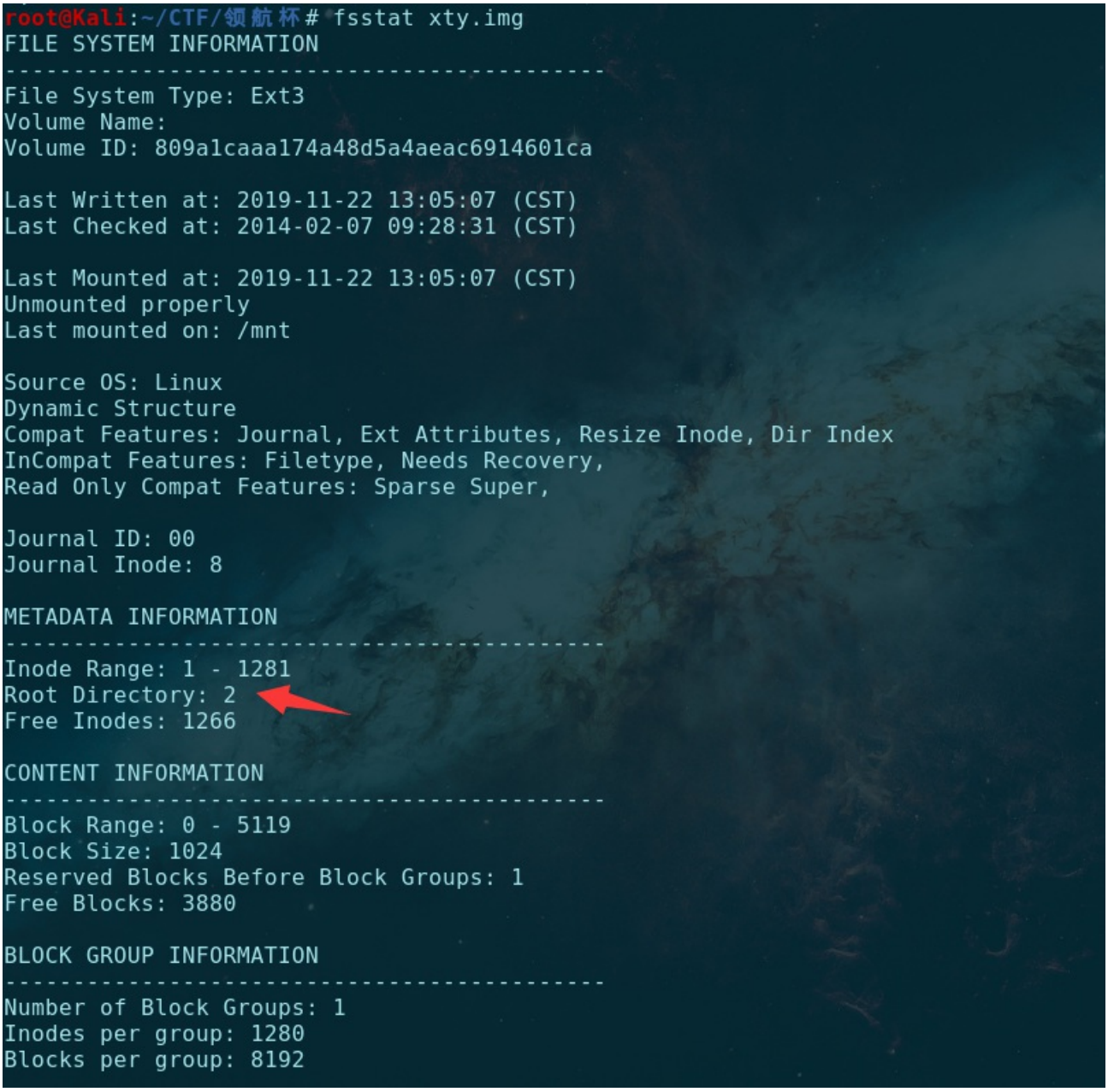


发一下一个隐藏文件夹`.hide`，里面有一个`secret.odg`，打开得到如下内容：



好像没什么用，想到题目描述中的磁盘恢复，所以需要用到sstat 和 ext3grep这两个工具，先用下面命令查看相关信息：

```
fsstat xty.img
```



注意这里`Root Directory: 2`中的2是我们要用到的。

再用ext3grep工具，查看目录如下：

```
ext3grep --inode 2 xty.img
```

```
Writing analysis so far to 'xty.img.ext3grep.stage2'. Delete that file if you want to do this
stage again.
The first block of the directory is 184.
Inode 2 is directory "".
Directory block 184:
         .-- File type in dir_entry (r=regular file, d=directory, l=symlink)
         |          .-- D: Deleted ; R: Reallocated
Indx Next |  Inode   | Deletion time                       Mode        File name
==========+==========+----------------data-from-inode------+-----------+=========
   0    1 d       2                                       drwxr-xr-x  .
   1    2 d       2                                       drwxr-xr-x  ..
   2    5 d      11                                       drwx------  lost+found
   3    5 r      12  D 1558974332 Tue May 28 00:25:32 2019  rrw-r--r--  0secret
   4    5 r      12  D 1558974332 Tue May 28 00:25:32 2019  rrw-r--r--  secret
   5    6 d      13                                       drwxr-xr-x  .hide
   6  end d      17                                       drwxr-xr-x  .ls
   7    8 r      16  D 1558974209 Tue May 28 00:23:29 2019  rrw-r--r--  secret.1
   8    9 r      18  D 1558974209 Tue May 28 00:23:29 2019  rrw-r--r--  secret.2
   9   10 r      19  D 1558974209 Tue May 28 00:23:29 2019  rrw-r--r--  secret.3
  10   11 r      20  D 1558974209 Tue May 28 00:23:29 2019  rrw-r--r--  secret.4
  11   12 r      21  D 1558974209 Tue May 28 00:23:29 2019  rrw-r--r--  secret.5
  12   13 r      22  D 1558974209 Tue May 28 00:23:29 2019  rrw-r--r--  secret.6
  13   14 r      23  D 1558974209 Tue May 28 00:23:29 2019  rrw-r--r--  secret.7
  14   15 r      24  D 1558974209 Tue May 28 00:23:29 2019  rrw-r--r--  secret.8
  15  end r      25  D 1558974209 Tue May 28 00:23:29 2019  rrw-r--r--  secret.9
  16  end r      26  D 1558974209 Tue May 28 00:23:29 2019  rrw-r--r--  secret.10
```

可以看到很多secret文件，实际上内容都一样，用下面命令将其恢复出来：

```
ext3grep --restore-file secret xty.img
```

```
root@Kali:~/CTF/锁航杯# ext3grep --restore-file secret xty.img
Running ext3grep version 0.10.2
WARNING: I don't know what EXT3_FEATURE_COMPAT_EXT_ATTR is.
WARNING: EXT3_FEATURE_INCOMPAT_RECOVER is set. This either means that your partition is st
ill mounted, and/or the file system is in an unclean state.
Number of groups: 1
Minimum / maximum journal block: 198 / 1227
Loading journal descriptors... sorting... done
The oldest inode block that is still in the journal, appears to be from 1391737510 = Fri F
eb  7 09:45:10 2014
Number of descriptors in journal: 246; min / max sequence numbers: 23 / 121
Loading xty.img.ext3grep.stage2... done
Restoring secret
```

打开该文件发现下面一串字符：

```
aWdxNDs3NDFSOzFpa1I1MWliT08waWdx
```

aWdxNDs3NDFSOzFpa1I1MWliT08waWdx

base64一下得到：

```
igq4;741R;1ikR51ibOO0igq
```

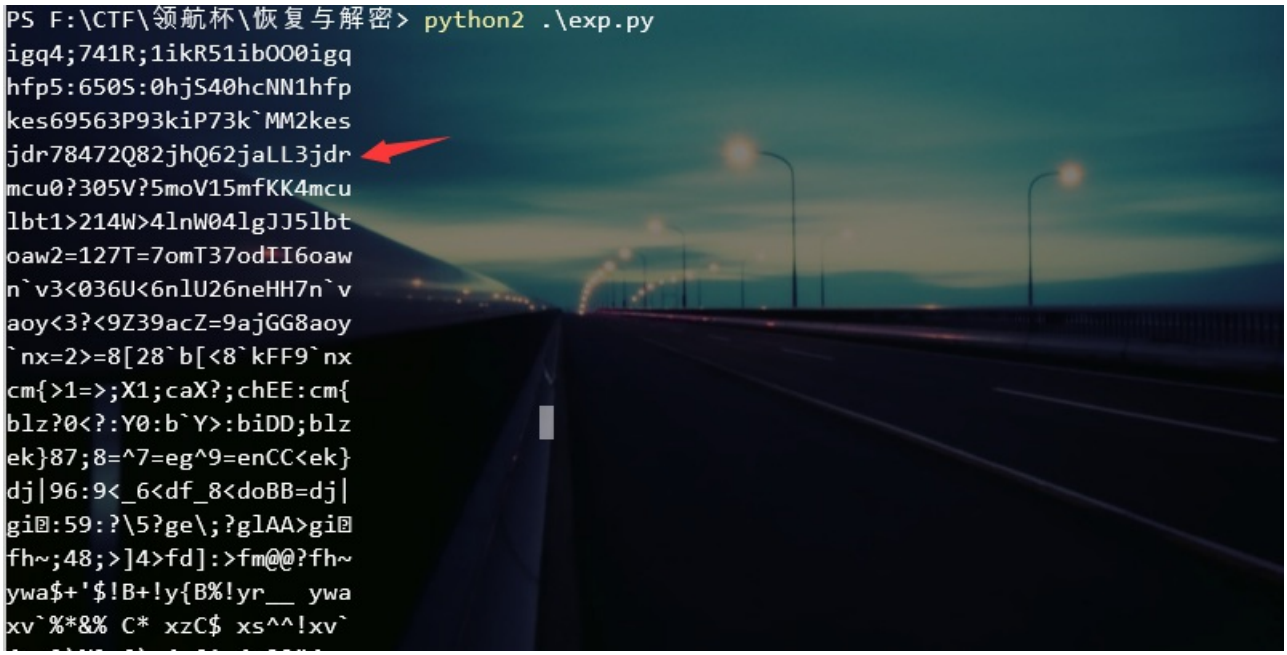
异或解密脚本如下：

```python
import string

c = "igq4;741R;1ikR51ibOO0igq"

for i in range(0,200):
    p = ""
    for j in range(len(c)):
        p += chr(ord(c[j])^i)
    print (p)
```

得到的唯一一个无特殊字符的即为flag：jdr78472Q82jhQ62jaLL3jdr



文件提取

给了一个flag.exe，strings一下可以看到是图片转base64后的字符串：

```
root@Kali:~/CTF/领航杯/文件提取# strings flag.exe

qPDU/Pjs1OjlDS2BRQ0daSDk6U3FUWmNma2xrQFB2fnRofWBpa2f/2wBDARITExkWGTEbGzFnRTpFZ2dnZ2dnZ2dnZ
2dnZ2dnZ2dnZ2dnZ2dnZ2dnZ2dnZ2dnZ2dnZ2dnZ2dnZ2dnZ2dnZ2f/wAARCADaAYYDASIAAhEBAxEB/8QAGwAAAQU
BAQAAAAAAAAAAAAAAAAECBAUGAwf/xABIEAABAwIEAwQFBgwFAwUAAAABAAIDBBEFEiExBkFREyJhgTJxkaGxBxQXQ
lPRFSMzQ1JUYnKSssHwFiQ24fFzk6ImNERVY//EABUBAQEAAAAAAAAAAAAAAAAAAAB/8QAFREBAQAAAAAAAAAAAAA
AAAAAAAH/2gAMAwEAAhEDEQA/ANyhCFECFlKnjqkpquWIwSvaxxGYdQbexc/pDoL/APtqj/x+9BqamoipoY+aaoX6
T3BoSxSNmiD43Nc07OacwKwHEfFFDjNA2OOCZsrXhzXPas3rsdU/h/jGHDMOjpKinkcI7hrmka6/FB6CEiyX0i0H6r
Uf+P3ob8odA7T5rP67t+9VWuTXODGkkgADcmwTIJWzwxyM9F7Q4eohcMY1wWt5f5eT+UqI6wVMNS0ugljlaObXBy7j
QLI/Jvc4XUk7mW3uWtzAb6IOMlVDFKyOSZjXv0a1zspd6gnySsijL5HBjRu5xssVxWWjjHDQD9nmaP3ja/krrjgf+m
5SDYh7fjZBdxPZJGHxuDmnZzdbroqDgcl3DVOSbkl3uNlfgoEzeBSgosOgVDiHFVHh2Jijkjkc4kAuA0bf+wgvkIQg
EIQgEIQgEIQgEIQgELmZ2hJ84ag6oXJs7U8PaeaBxAGqaHEpC0m5uEgcG7nzQOt3lCrcQjpL5zyPTklxLE
W0NKZCRt19681xjHZK6ZxDu76I9SC5xDjV7JXCEZuR10CqpuLa2R12vA9qok3fyQavD+MKlrLSAOV1hvFLJXWeQNRb
Vedh3Ic+aVsjmbG2t9Cg9qglZPG1wN9F02K8+4Z4nkjeIqh7jsM263VLOyojDgRsgkN5pya3mnIC4br5ISFCBUIXOe
UQQSSO1axpcbeG6DAcM0tNVcW1sdTCydre0c3O24uHW1B52utt+BcNLSPmFLb/pN+5ZWn4wwqnqXztw50Ur9HOZlu7
nrt/ftXf6QqLI4ilmJA0GlvNA3H8QwvC66OjiwilqJTZzhka3LfloN9NvFaGLCMPdG0uoKZpIBI7Jtl57heMQMx44l
iTXSvJLmBoGjuR9Vvf6lomfKFQkWfTTg2+rlOvtCDR/gTDP8A6+l/7TfuWT4/o6SlpqT5vTRRPc92rGBtwBt6lK+kS
j/U5x63NXKo44wmqYG1FBJIAbgPa1wug1eFaYZSj/8AFvwWTxx/Er5K5rG2o7OF7N9AX169VpsErosRw6KohjMUZ0D
SF0xc2wetPSB/8pQefcKy40Ipm4TG10ZcM7nNbYHz529a0VbwvX4hI2abFC2TL3mtacoO/dXH5Oy1mE1byQG9rq4nT
Qe4K3xPiWgwuRjKmRznObmGRuYW239aDG4jgxwfiDDo5JzUPe5r3E9c3w0Wt44/0zPy7zeX7QWUxnGIMX4loJ6QyZY
3Mbdwy65lq+OP9MVH77P5ggzXD/DFTiGGR1LcSfBG4nutBdsbdbLU8OYLLhEUrJqp1Q6QtOYg6Wv1VFwzxThuG4LFT
```

随便找个在线网站转回图片如下：

然后strings一下这个图片得到flag：flag{068EEF6A7BAD3FDF}

```
%H\ouGj
%GNB
NMo4
8;Ki
# >Oi
flag{068EEF6A7BAD3FDF}
```

forensics

给了一个pngdecode.docx：

朋友，密码是 8 位

你自己猜哦？

一开始分析了好久这个图片，但是无果…然后对整个docx文档foremost提取一下，得到flag：360HA360

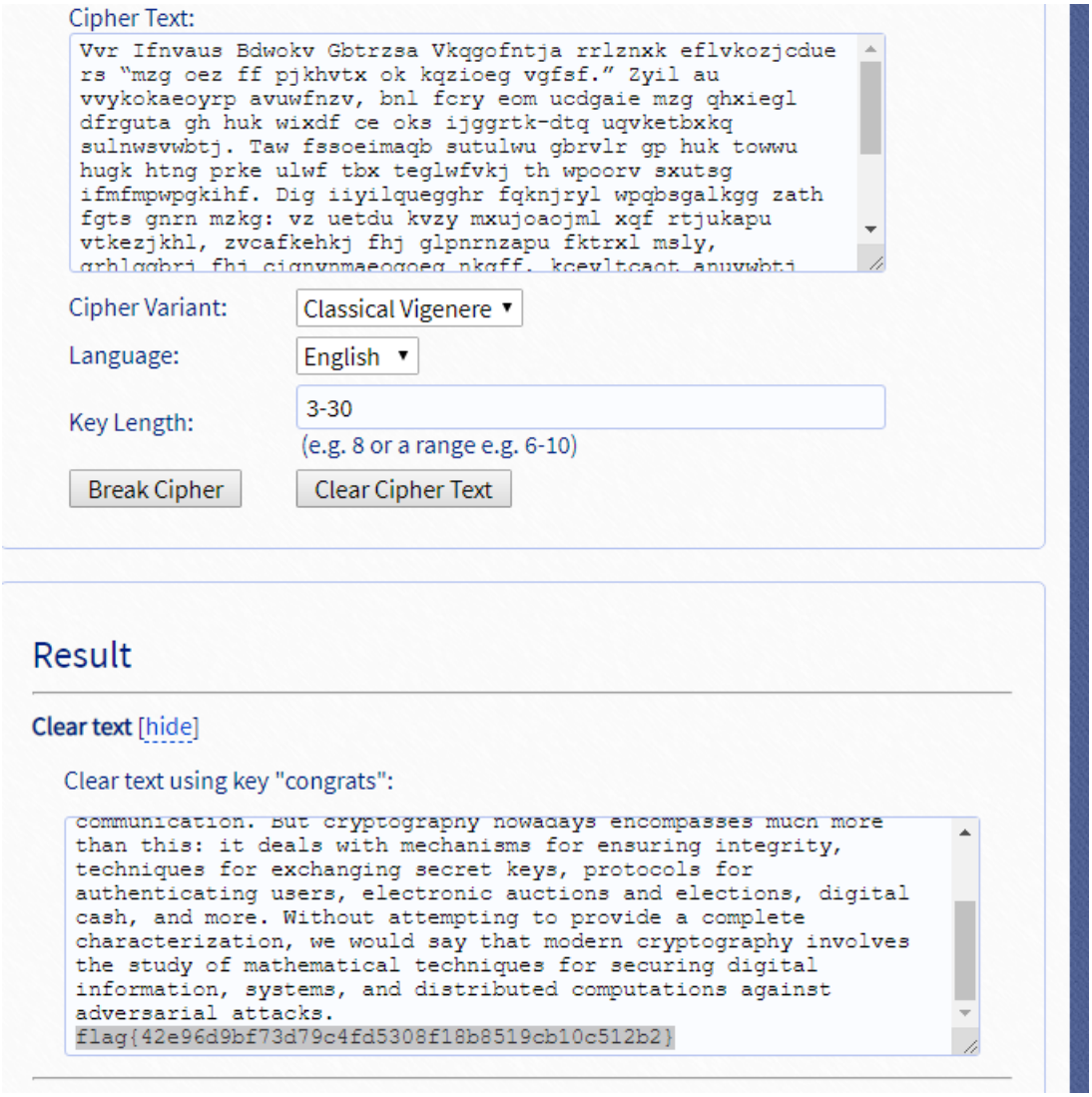
OldCypher-Easy

给了密文如下：

```
Vvr Ifnvaus Bdwokv Gbtrzsa Vkqgofntja rrlznxk eflvkozjcdue rs “mzg oez ff pjkhvtx ok kqzioeg vgfsf.” Zyil au vvy kokaeoyrp avuwfnzv, bnl fcry eom ucdgaie mzg qhxiegl dfrguta gh huk wixdf ce oks ijggrtk-dtq uqvketbxkq sulnwsvw btj. Taw fssoeimaqb sutulwu gbrvlr gp huk towwu hugk htng prke ulwf tbx teglwfvkj th wpoorv sxutsg ifmfmpwpgkihf. Dig iiyilquegghr fqknjryl wpqbsgalkgg zath fgts gnrn mzkg: vz uetdu kvzy mxujoaojml xqf rtjukapu vtkezjkhl, zv cafkehkj fhj glpnrnzapu fktrxl msly, grhlqqbrj fhj cignvnmaeogoeg nkgff, kcevltcaot anuvwbtj agv gzrikihfu, rvmz ttd eofn, rnw eqfr. Cztagwh nzkefhvwam ko ijqjvjv a vgodykke vzcfnikekabogofn, pw ychru stq vvnz dowwtb pxppmgif nvyy bfxcybvs mzg ggauy hx oognvmtlkqnr kevzpwdavs ygt grilrbfi rvmzttd kbsuimtlkca, ypsmwog, ntu dbkvfvhltxv ec zvlttlkcay rgtapgg guvxjuoeorl tlvopqj. fesi{42s96q9hw73d79v4xf5308t18o8519is10c512u2}
```

最后一串明显像flag的格式，在这个网站进行维吉尼亚解密：https://www.guballa.de/vigenere-solver

得到flag：flag{42e96d9bf73d79c4fd5308f18b8519cb10c512b2}



MasterofZip-Middle

level1

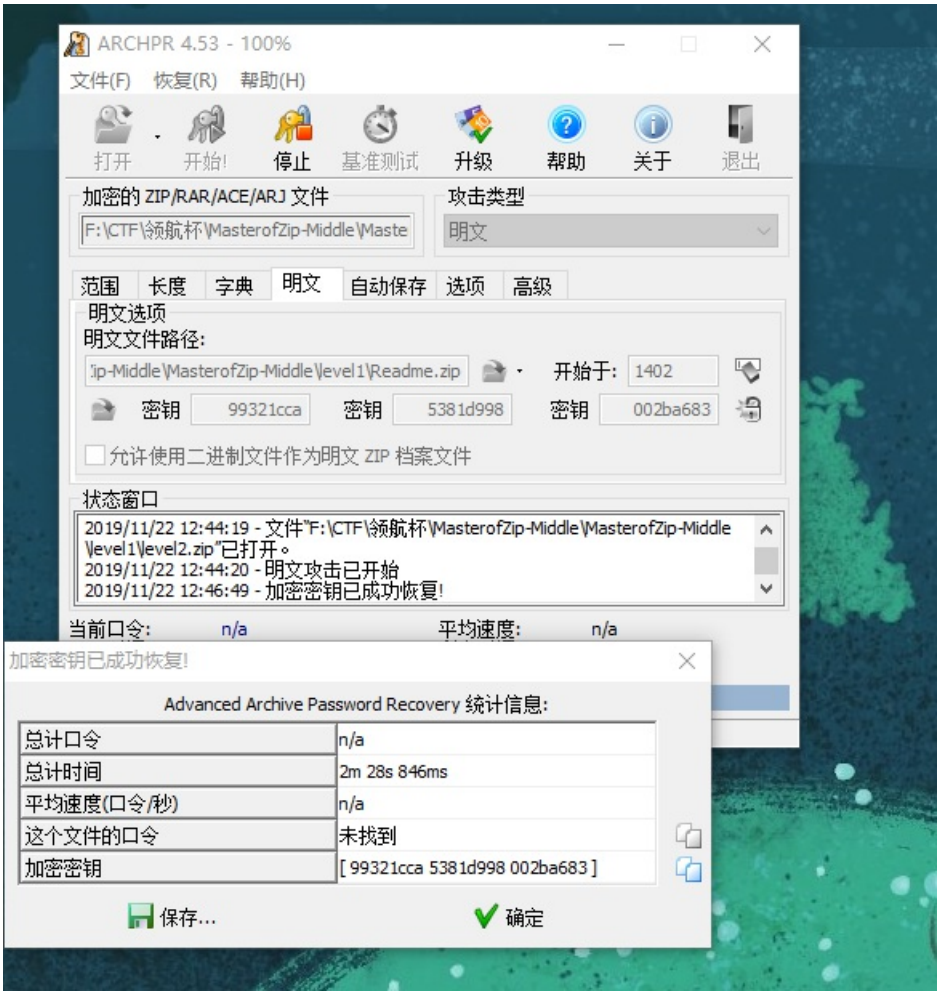
给了一个压缩包，和一个Readme.md内容如下：

小明喜欢用自己的手机号来加密zip，我们通过社工只查到了他手机号的前三位为199，你能解开这个压缩包吗？(手机号为伪造手机号，请不要试图拨打或社工)

于是爆破199开头的手机号，得到密码：19950453796

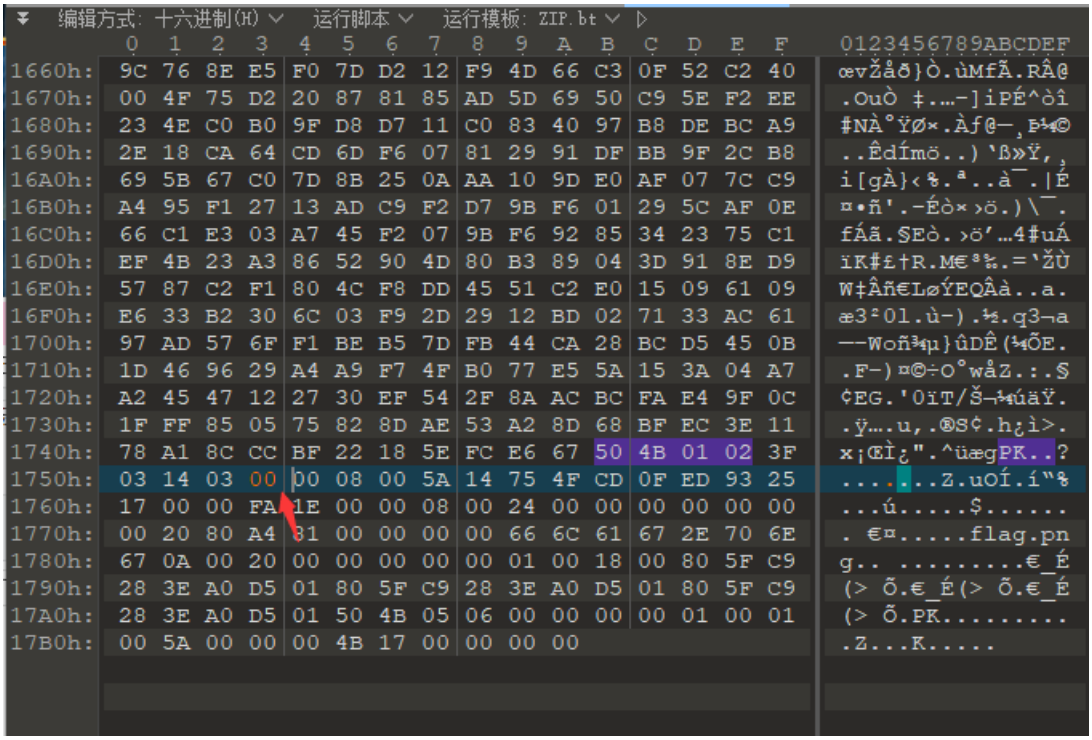
level2

第一层解压后得到一个压缩包和一个Readme.txt，内容不重要，但是看到level2.zip里面也有一个Readme.txt，所以尝试明文攻击，成功恢复密钥并解压：



level3

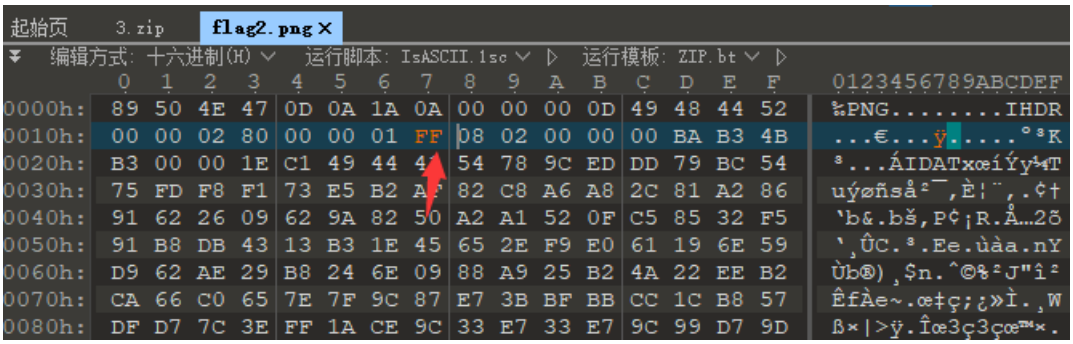
第二层解压后，只给了一个压缩包，于是尝试伪加密：



将箭头位置修该为0，成功解压，得到下面这张图片：

???

然后修改发图片高度：



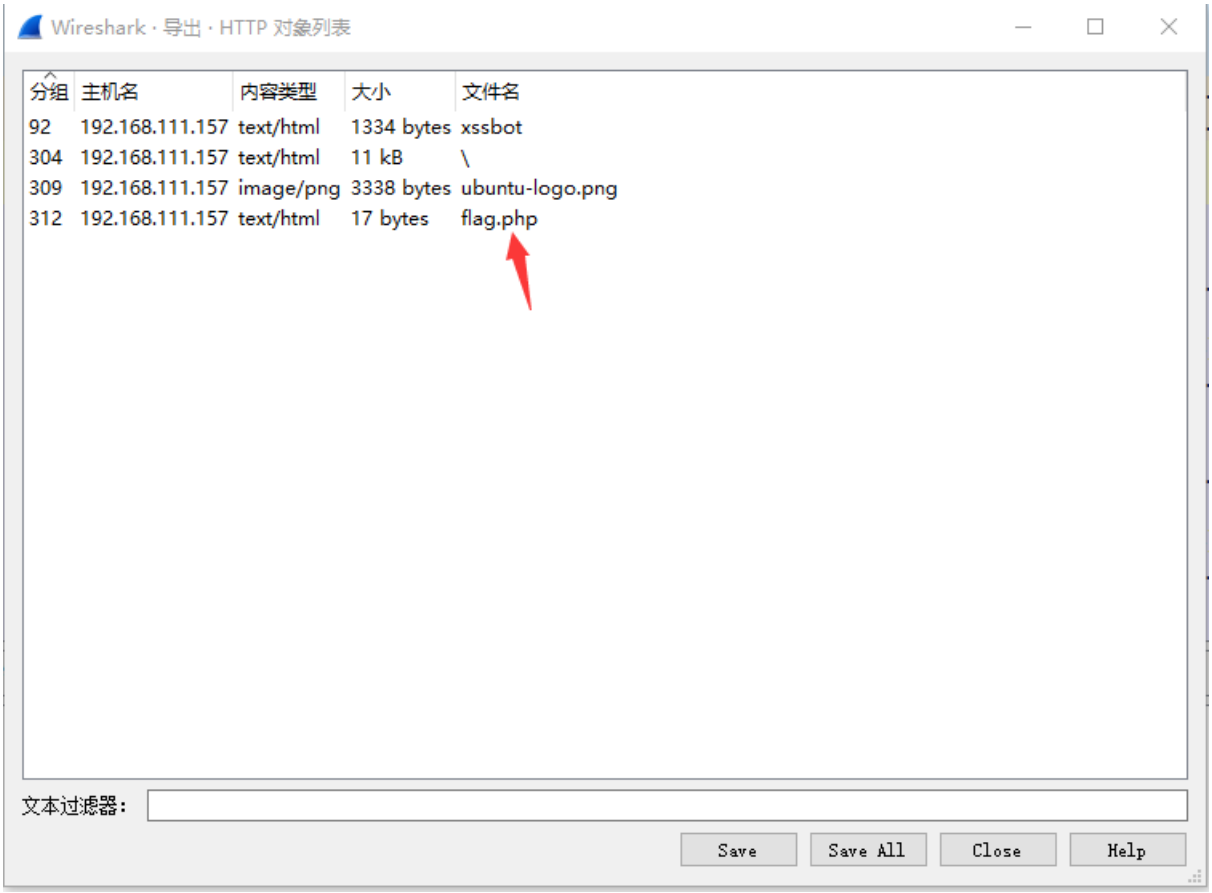
得到flag：flag{4537ec3bd52ba2b41c4a780db841efc3ddccc4a4}

???

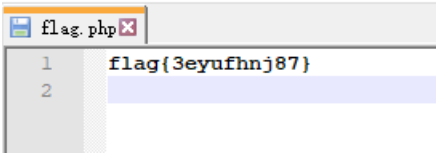
flag{4537ec3bd52ba2b41c4a780db841efc3ddccc4a4}

数据包分析-Easy

题目给了一个http3.pcap，进行流量分析，根据题目描述，直接导出HTTP对象列表，发现flag.php:



导出后，即可在里面看到flag：flag{3eyufhnj87}



EasyRSA

题目给了加密脚本如下：

```
from Crypto.Util.number import getPrime, inverse
flag = 'flag{a-z0-9}'
nbits = 2048
p = getPrime(nbits / 2)
q = getPrime(nbits / 2)
assert p != q
N = p * q
e = 0x10001
phiN = (p - 1) * (q - 1)
d = inverse(e, phiN)
phint = d % (p - 1)
qhint = q % (p - 1)


def str2int(s):
    return int(s.encode('hex'), 16)


with open('pubkey.txt', 'w') as f:
    f.write(str(e) + '\n')
    f.write(str(N) + '\n')
    f.write(str(phint) + '\n')
    f.write(str(qhint) + '\n')

plain = str2int(flag)

c = pow(plain, e, N)
with open('cipher.txt', 'w') as f:
    f.write(hex(c))
```

但实际上没什么用，只是告诉了我们pubkey.txt里面分别对应的内容，即：

```
e = 65537
N = 16969752165509132627630266968748854330340701692125427619559836488350298234735571480353078614975580378467355952333755313935516513773552163392952656321490268452556604858966899956242107008410558657924344295651939297328007932245741660910510032969527598266270511004857674534802203387399678231880894252328431133224653544948661283777645985028207609526654816645155558915197745062569124587412378716049814040670665079480055644873470756602993387261939566958806296599782943460141582045150971031211218617091283284118573714029266331227327398724265170352646794068702789645980810005549376399535110820052472419846801809110186557162127
phint = 17816257752910288702696852575211080903295430127287054677825469139515376426236217692464411221899486713749909464051644598674106468255913106226183791162842937940909702921652633347493930099993354130899037966243261680396182870781926464904885340628039604187908748904355293930473892287188352443706452151873580818 05
qhint = 10450978538445496868237560522155647855468044038637044818500580506745232482415364474390893285539835615564332910332708101539048528242011762736342572650763270501265440674647489375438852377280494520168349154604800218665628586180057648386859933274414030182106920793492451577530884172876623074281199949317487086975
```

而这里的phint和qhint实际上就是常说的dp和dq。

还给了密文如下：

```
0x7b5d1ea2d92df27239817ce8d885e1f66569dd41e075efc13d09dd1df673a8fba68ec7487c1552028e9eb9ba6663983f96d01925bbdfd18398e44f970257fa0f96b6ec915d05d637ebb4c8f4c56c44b2bb46bd1afe5a67acd640585dccf1681155308c0663cb57fccdc10c097c454dafdd2a96ccd08e9a2a8b0a9727bbe9945e579b0652d1c1d826305f0dd716cfb647cadb8eca1a0286dfb938b60b89981403d4faa6df54cfac0fa4699c97aeba6e82ab575cd6aa4421018cf9b404836c02b5301dbc475a0bcc5eef86bcbeb89a73355dbeb80e7b4d23c7a39f32c6b6138125c73892633f46b0bf1114aa67f09e1d394dfa4020e318f7d8004b84fc835b1ee9L
```

所以这题我们有c、e、n、dp、dq，足够解出明文了。

参考这篇文章：https://skysec.top/2018/08/24/RSA%E4%B9%8B%E6%8B%92%E7%BB%9D%E5%A5%97%E8%B7%AF(1)/

最终解密脚本如下：

```
import gmpy2
import libnum
e = 65537
n = 16969752165509132627630266968748854330340701692125427619559836488350298234735571480353078614975580378467355952333755313935516513773552163392952656321490268452556604858966899956242107008410558657924344295651939297328007932245741660910510032969527598266270511004857674534802203387399678231880894252328431133224653544948661283777645985028207609526654816645155558915197745062569124587412378716049814040670665079480055644873470756602993387261939566958806296599782943460141582045150971031211218617091283284118573714029266331227327398724265170352646794068702789645980810005549376399535110820052472419846801809110186557162127
dp = 17816257752910288702696852575211080903295430127287054677825469139515376426236217692464411221899486713749909464051644598674106468255913106226183791162842937940909702921652633347493930099993354130899037966243261680396182870781926464904885340628039604187908748904355293930473892287188352443706452151873580818 05
c = "0x7b5d1ea2d92df27239817ce8d885e1f66569dd41e075efc13d09dd1df673a8fba68ec7487c1552028e9eb9ba6663983f96d01925bbdfd18398e44f970257fa0f96b6ec915d05d637ebb4c8f4c56c44b2bb46bd1afe5a67acd640585dccf1681155308c0663cb57fccdc10c097c454dafdd2a96ccd08e9a2a8b0a9727bbe9945e579b0652d1c1d826305f0dd716cfb647cadb8eca1a0286dfb938b60b89981403d4faa6df54cfac0fa4699c97aeba6e82ab575cd6aa4421018cf9b404836c02b5301dbc475a0bcc5eef86bcbeb89a73355dbeb80e7b4d23c7a39f32c6b6138125c73892633f46b0bf1114aa67f09e1d394dfa4020e318f7d8004b84fc835b1ee9L"
c = int(c[:-1], 16)
for i in range(1,65538):
    if (dp*e-1)%i == 0:
        if n%(((dp*e-1)/i)+1)==0:
            p=((dp*e-1)/i)+1
            q=n/(((dp*e-1)/i)+1)
            phi = (p-1)*(q-1)
            d = gmpy2.invert(e,phi)%phi
            print libnum.n2s(pow(c,d,n))
```

运行得到flag：flag{6b85823e6f121a7bb3407ff2e9f5f2f27efcc5a6}