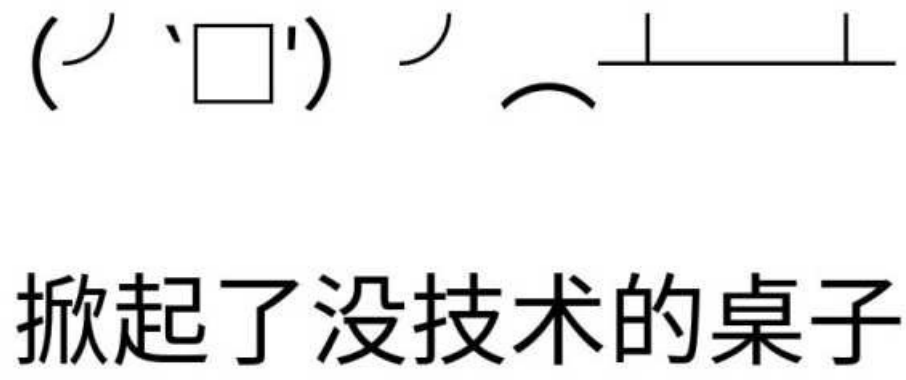
DDCTF-2018-writeup(5misc)

转载

weixin_30340819 于 2018-04-20 11:40:00 发布 166 收藏 1
文章标签： python 操作系统 密码学
原文链接：http://www.cnblogs.com/kagari/p/8889412.html

打了好几天最后也只是80多名,我好菜啊.jpg



0x00 (ノ °□°）ノ ┻━┻

题目：

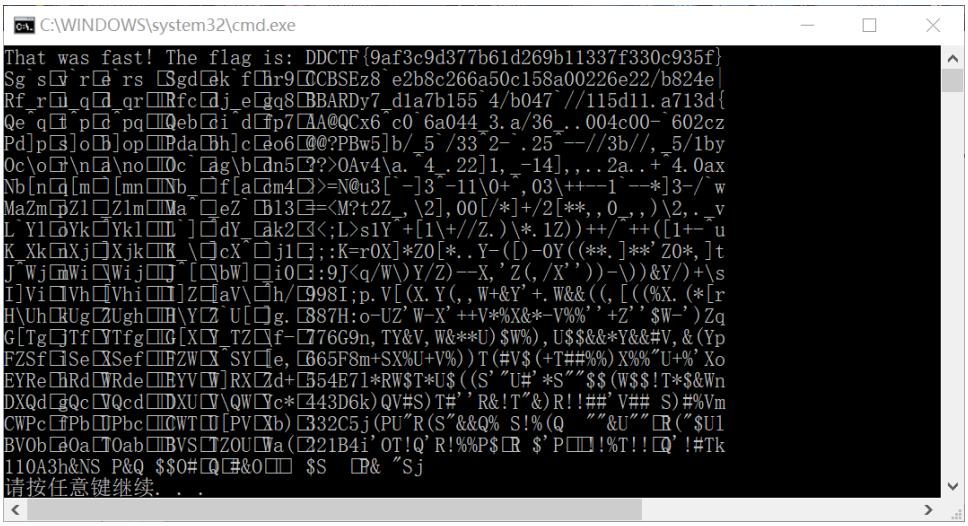
(ノ °□°）ノ ┻━┻

d4e8e1f4a0f7e1f3a0e6e1f3f4a1a0d4e8e5a0e6ece1e7a0e9f3baa0c4c4c3d4c6fbb9e1e6b3e3b9e4b3b7b7e2b6

一开始被这个(ノ °□°）ノ ┻━┻误导以为是jjencode，尝试了好久没啥结果，后面又有人说是翻转，试了很久还是没结果。然后跑了下移位密码获取flag

贴上脚本

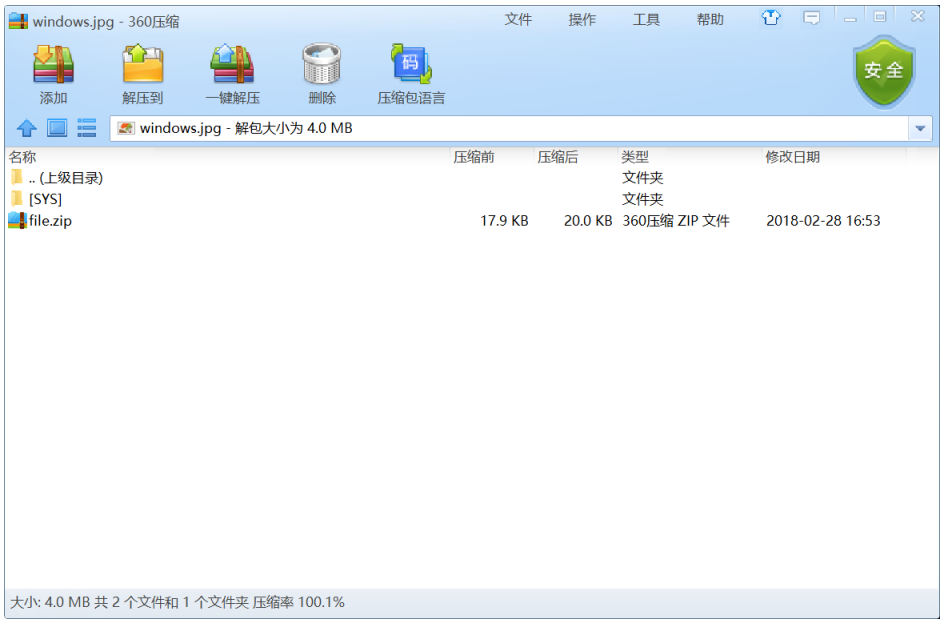
```
s='d4e8e1f4a0f7e1f3a0e6e1f3f4a1a0d4e8e5a0e6ece1e7a0e9f3baa0c4c4c3d4c6fbb9e1e6b3e3b9e4b3b7b7e2b6b1e4b2b6b9e2b1b1b3b3b7e6b3b3b0e3b9b3b5e6fd'
'''
s1=''
for x in range(len(s)/2):
    s1+=chr((int(s[x*2:x*2+2],16))%128)
print s1
'''

for j in range(20):
    s1=''
    for x in range(len(s)/2):
        s1+=chr((int(s[x*2:x*2+2],16)-j)%128)
    print s1

```

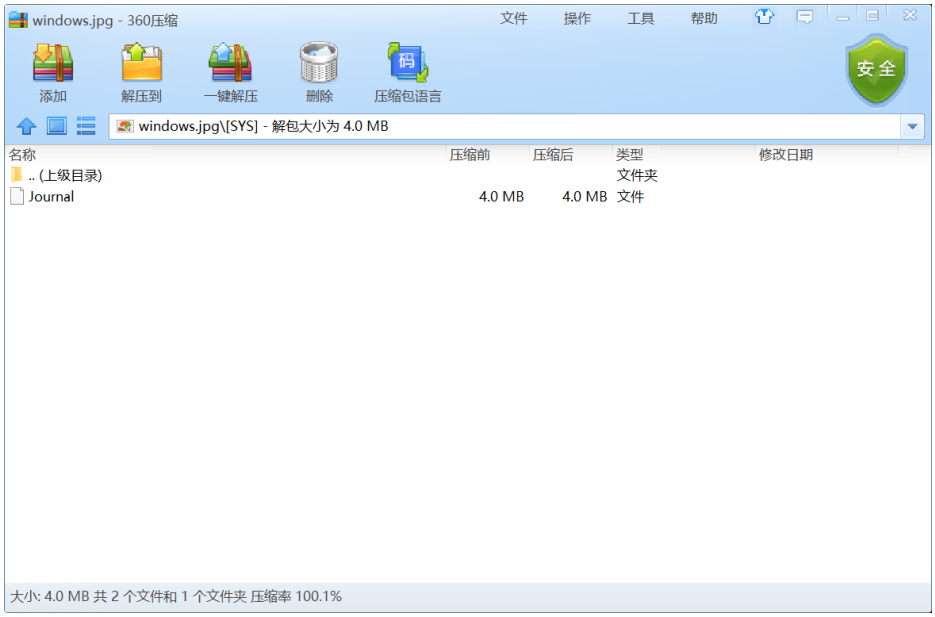


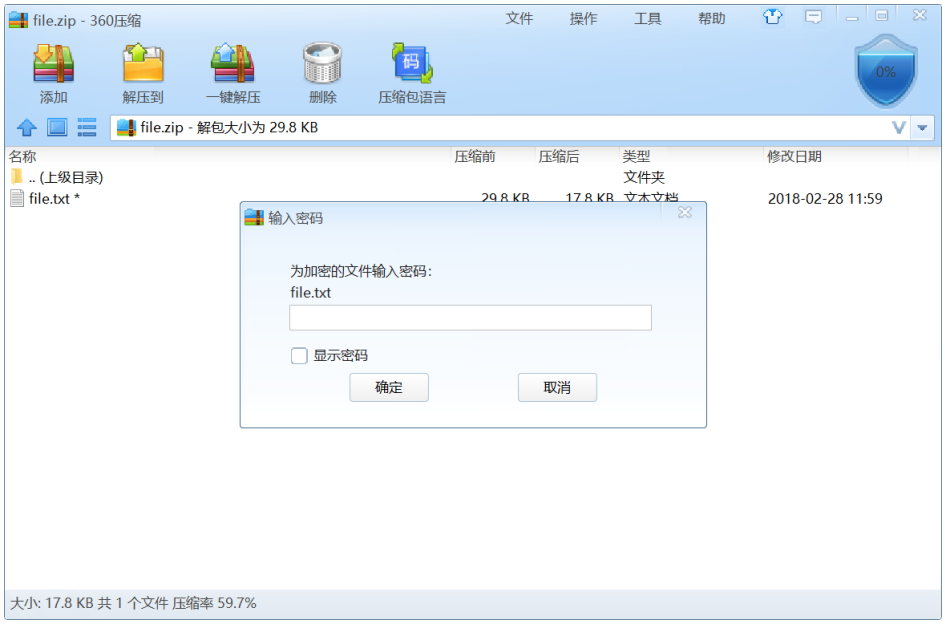
0x01 第四扩展FS

题目:D公司正在调查一起内部数据泄露事件，锁定嫌疑人小明，取证人员从小明手机中获取了一张图片引起了怀疑。这是一道送分题，提示已经在题目里，日常违规审计中频次有时候非常重要。
https://pan.baidu.com/s/1DJpMFU2lajHGTo0yfzTHVQ 密码：fpp4

这道题很多坑，拿到题目看题目名字就可以知道它考的是ext4文件，先用压缩软件打开查看了一下







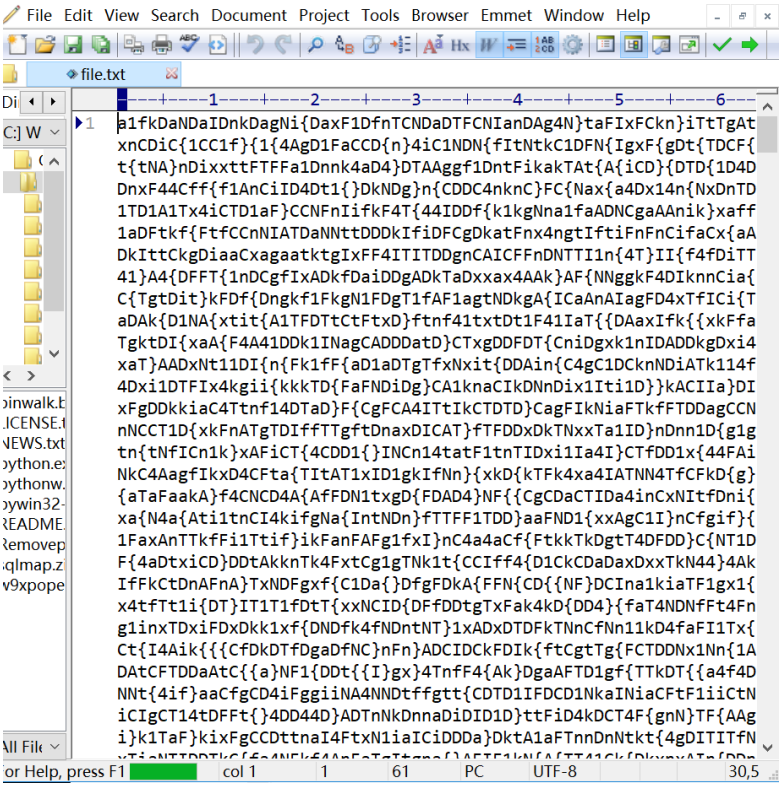
因为没怎么接触过ext4文件系统所以不知道journal文件是拿来干嘛的，百度后说是文件系统日志，所以考虑mount到linux上，但是怎么都不成功。

另外还有一个加密的file.txt，用binwalk扫了下文件，发觉就是这两个文件

```
PS C:\WINDOWS\system32> binwalk [redacted]windows.jpg
* suggest: you'd better to input the parameters enclosed in double quotes.
* ************************************************************************

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
30            0x1E            TIFF image data, big-endian, offset of first image directory: 8
2825118       0x2B1B9E        Linux EXT filesystem, rev 2.0, ext4 filesystem data, UUID=57f8f4bc-abf4-0000-675f-946fc0f9
c0f9
7236510       0x6E6B9E        Zip archive data, encrypted at least v2.0 to extract, compressed size: 18210, uncompressed
 size: 30500, name: file.txt
7254812       0x6EB31C        End of Zip archive, footer length: 22

PS C:\WINDOWS\system32>
```

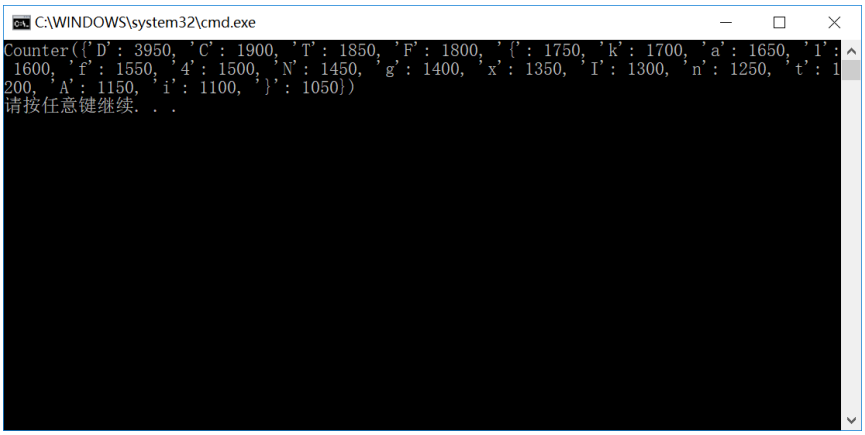
为了寻找file.txt密码，用winhex打开jpg后发现有些图片有敏感信息，然后直接查看图片属性，发现解压密码

解压打开file.txt



根据提示频率，统计一下字符出现频率，贴上脚本，得到flag

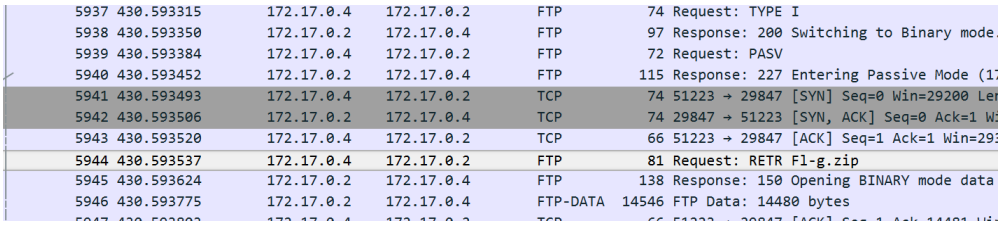
```
1 import collections
2
3 f=open('file.txt','r')
4 print collections.Counter(f.read())
```

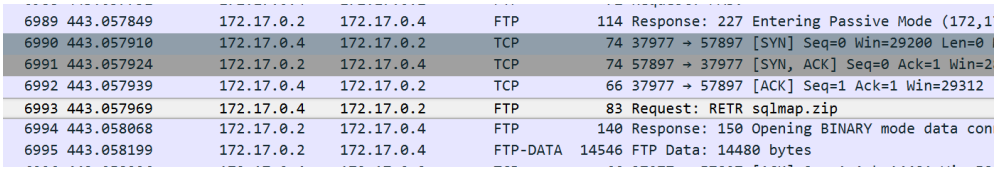


0x02 流量分析

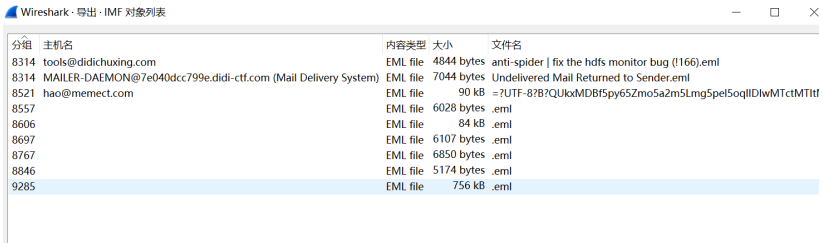
题目:提示一：若感觉在中间某个容易出错的步骤，若有需要检验是否正确时，可以比较MD5:
90c490781f9c320cd1ba671fcb112d1c
提示二：注意补齐私钥格式
-----BEGIN RSA PRIVATE KEY-----
XXXXXXX
-----END RSA PRIVATE KEY-----

这题给的数据包很大，而且有很多坑，数据包前面有两个用ftp传输加密压缩文件fl-g.zip，sqlmap-dev.zip，中间有九封邮件，只到看到最后才会发现本题关键ssl加密通信。





前面的两个压缩包可能是本题彩蛋，不过我是真的解不出来。。。所以跳过，直接导出邮件。



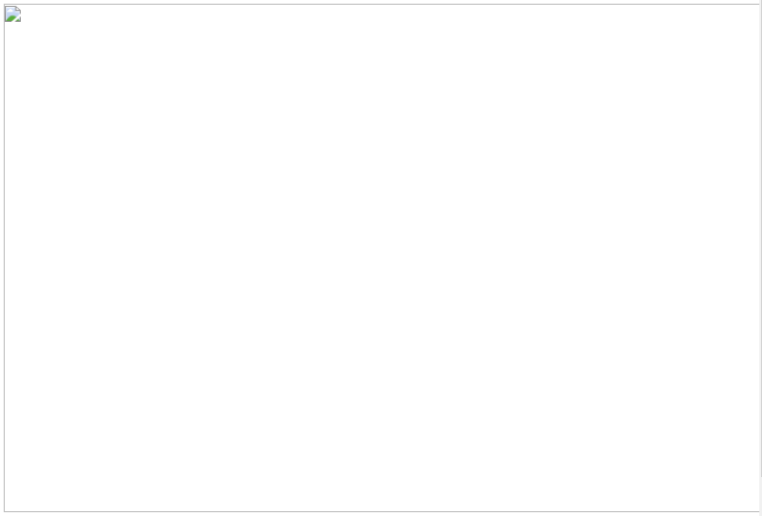
将导出的邮件导入qq邮箱中查看，可以发现这个

(5).eml

? ? ? ? ?

发件人：(杨睿信息安全部)<zhangsan@didi-ctf2.com>
时　间：2018年1月11日(星期四) 下午5:56
收件人：yangruiyangrui<zhangsan@didi-ctf2.com>

小张你好：



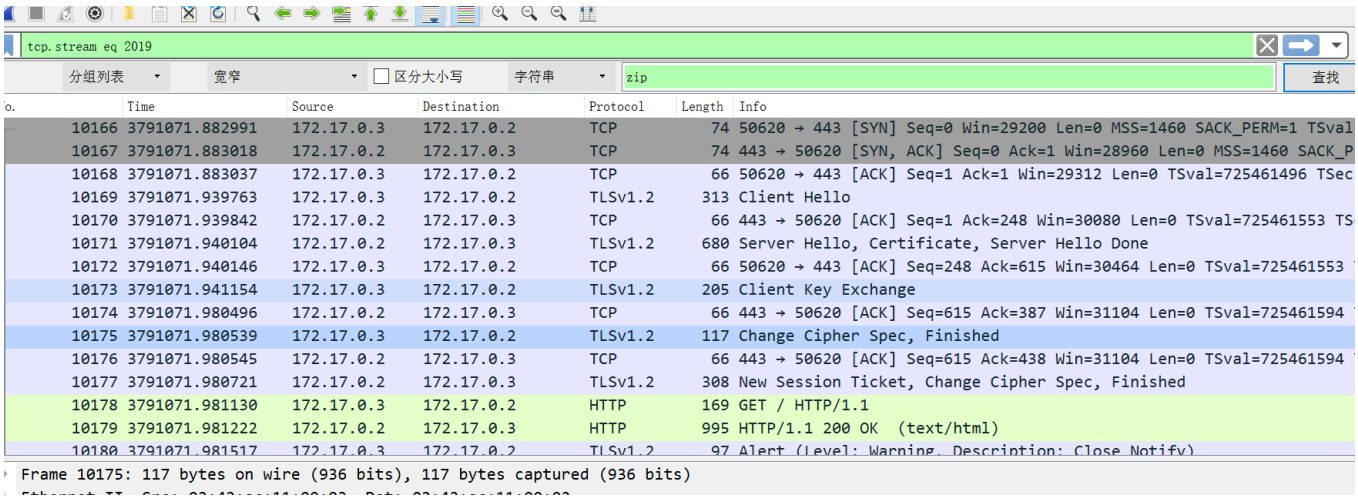
你好，请你将密钥安装到服务器上。谢谢

```
--B_3598538194_2172270964--


--B_3598538194_700708737
Content-type: image/png; name="image001.png";
 x-mac-creator="4F50494D";
 x-mac-type="504E4766"
Content-ID: <image001.png@01D38B05.8079CE40>
Content-disposition: inline;
             filename="image001.png"
Content-transfer-encoding: base64

iVBORw0KGgoAAAANSUhEUgAABIwAAAISCAYAAACu6lSmAAAMKGlDQ1BJQ0MgUHJvZmlsZQAA
SImVVwdYU8kWnluSkJDQAhGQEnoTpVepoUUQkCrYCEkgocSYEETsyKKCa0HFghVZFVFwLYAs
NixYWBTs9WFBRVkXCzZU3iQBdPV7733vfN+5979nzpz5z7kz880AoB7NEYuzUA0AskU5kpjQ
QObEpGQm6SFAAArIwBM4crhScUB0dASAMvT+p7y7Dr2hXLGXx/q5/b+KJo8v5QKAREOcypNy
syE+BADuxhVLcgAg9EC72cwcMcREyBJoSyBBiM3lOF2JPeQ4VYkjFD5xMSyIUwBQoXI4knQA
1OS8mLncdBhHbRnEDiKeUARxE8S+XAGHB/FniEdlZ0+HWN0aYuvU7+Kk/yNm6nBMDid9GCtz
UYhKkFAqzuLM+j/L8b8lO0s2NIYZVKpAEhYjz1let8zp4XJMhficKDUyCmItiK8KeQp/OX4i
kIXFD/p/4EpZsGaAAQBK5XGCwiE2gNhUlBUZMWj3TROGsCGGtUfjhDnsOGVflCeZHjMYH83j
S4NjhzBHohhL7lMsy4wPGIy5RcBnD8VszBfEJSp5opdzhQmREKtBfFeaGRs+6PM8X8CKHPKR
yGLknOE/x0CaJCRG6YOZZ0uH8sK8BEJ25CCOyBHEhSn7YlO5HAU3XYgz+NKJEUM8efygYGVe
WAFfFD/IHysV5wTGDPpXirOiB/2xJn5WqNxuCnGbNDd2qG9vDpxsynxxIM6JjlNyw7UzOOOi
lRxwWxABWCAIMIEMaiqYDjKAsK2nvgd+KVtCAAdIQDrgA/tBy1CPREWLCD5jQT74CyI+kA73
C1S08kEutH8Ztiqf9iBN0Zqr6JEJnkCcDcJBFvyWKXqJhkdLAI+hRfjT6FzINQuqvO0nG1N9
yEYMJgYRw4ghRBtcH/fFvfEI+PSH6oR74J5DvL75E54Q2gkPCdcInYRb04QFkh+YM8F40Ak5
hgxml/p9drgljOqKB+I+MD6MjTNwfWCPu8CRAnA/OLYrtH7PVTac8bdaDsYiO5BR8giyP9n6
RwZqtmquw1Hklfq+FkpeqcPVYg23/JgH67v68eA7/EdPbAl2EGvBTmLnsSasHjCx41gD1ood
lePhufFYMTeGRotR8MmEcYQ/jccZHFNeNalDtUO3w+fBNpDDz8uRLxbWdPEsiTBdkMMMgLs1
```

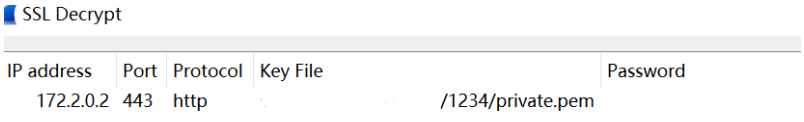
分组 9189。57 客户端 分组，8 服务器 分组，15 turn(s).点击选择。

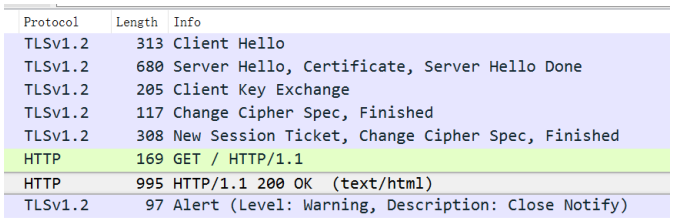
这里是本题关键，在数据包中这里是很大一串base64，解密后得到rsa私钥

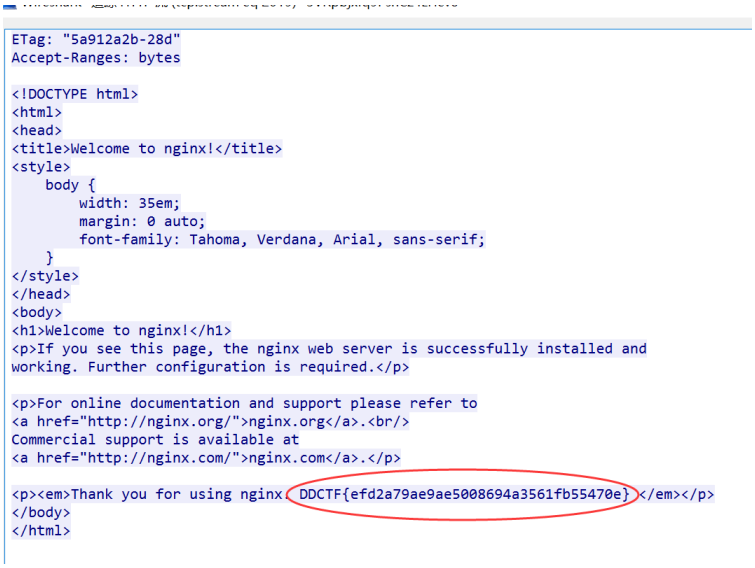
```
MIICXAIBAAKBgQDCm6vZmclJrVH1AAyGuCuSSZ8O+mIQiOUQCvN0HYbj8153JfSQ
LsJIhbRYS7+zZ1oXvPemWQDv/u/tzegt58q4ciNmcVnq1uKiygc6QOtvT7oiSTyO
vMX/q5iE2iClYUIHZEKX3BjjNDxrYvLQzPyGD1EY2DZIO6T45FNKYC2VDwIDAQAB
AoGAbtWUKUkx37lLfRq7B5sqjZVKdpBZe4tL0jg6cX5Djd3Uhk1inR9UXVNw4/y4
QGfzYqOn8+Cq7QSoBysHOeXSiPztW2cL09ktPgSlfTQyN6ELNGuiUOYnaTWYZpp/
QbRcZ/eHBulVQLlk5M6RVs9BLI9X08RAl7EcwumiRfWas6kCQQDvqC0dxl2wIjwN
czILcoWLig2c2u71Nev9DrWjWHU8eHDuzCJWvOUAHIrkexddWEK2VHd+F13GBCOQ
ZCM4prBjAkEAz+ENahsEjBE4+7H1HdIaw0+goe/45d6A2ewO/lYH6dDZTAzTW9z9
kzV8uz+Mmo5163/JtvwYQcKF39DJGGtqZQJBAKa18XR16fQ9TFL64EQwTQ+tYBzN
+04eTWQCmH3haeQ/0Cd9XyHBUveJ42Be8/jeDcIx7dGLxZKajHbEAfBFnAsCQGq1
AnbJ4Z6opJCGu+UP2c8SC8m0bhZJDelPRC8IKE28eB6SotgP61ZqaVmQ+HLJ1/wH
/5pfc3AmEyRdfyx6zwUCQCAH4SLJv/kprRz1a1gx8FR5tj4NeHEFFNEgq1gmiwmH
2STT5qZWzQFz8NRe+/otNOHBR2Xk4e8IS+ehIJ3TvyE=
```



，将私钥识别补齐格式进行ssl解密得到







0x03 安全通信

题目：请通过nc XXXX.XXXX.XXXX.XXXX XXXX答题，mission key是b9ba15b341c847c8beba85273f9b7f90，agent id随意填就可以

```python
#!/usr/bin/env python
import sys
import json
from Crypto.Cipher import AES
from Crypto import Random


def get_padding(rawstr):
    remainder = len(rawstr) % 16
    if remainder != 0:
        return '\x00' * (16 - remainder)
    return ''


def aes_encrypt(key, plaintext):
    plaintext += get_padding(plaintext)
    aes = AES.new(key, AES.MODE_ECB)
    cipher_text = aes.encrypt(plaintext).encode('hex')
    return cipher_text


def generate_hello(key, name, flag):
    message = "Connection for mission: {}, your mission's flag is: {}".format(name, flag)
    return aes_encrypt(key, message)


def get_input():
    return raw_input()


def print_output(message):
    print(message)
    sys.stdout.flush()


def handle():
    print_output("Please enter mission key:")
    mission_key = get_input().rstrip()

    print_output("Please enter your Agent ID to secure communications:")
    agentid = get_input().rstrip()
    rnd = Random.new()
    session_key = rnd.read(16)

    flag = '<secret>'
    print_output(generate_hello(session_key, agentid, flag))
    while True:
        print_output("Please send some messages to be encrypted, 'quit' to exit:")
        msg = get_input().rstrip()
        if msg == 'quit':
            print_output("Bye!")
            break
        enc = aes_encrypt(session_key, msg)
        print_output(enc)


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

分析代码得知，该题为aes ecb加密，这几天密码学课刚好讲到这，ecb是一种非常不安全的加密，wiki上有很好的举例。脚本放后面了。

1.本题通过输入agent_id,构造形如

Connection for mission: {agent_id}, your mission's flag is: DDCTF{32位}的字符串，

用随机生成的16位密钥进行加密，在一次连接中密钥不会变换

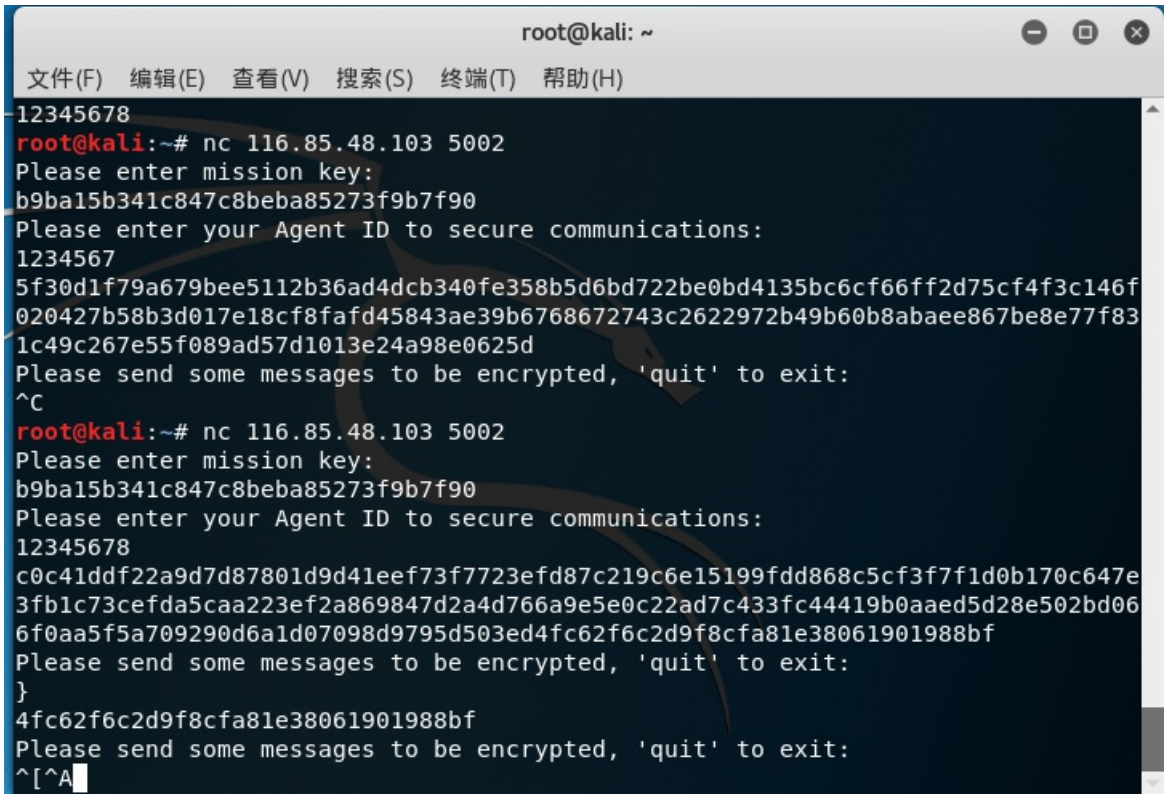
2.因为aes的特性通过改变输入agent_id长度，可以确定flag长度为32位，如下图

3.因为aes特性将字符串16位一组分组，当输入12345678时，字符串多出一位}，分析脚本得知会用0补齐。

```
Connection for m
ission: 1234567,
 your mission's
flag is: Edctf{8
9012345678901234
561234567890123}
```

```
Connection for m
ission: 12345678
, your mission's
 flag is: Edctf{
8901234567890123
4561234567890123
}
```

4.aes_ecb的特性，最后一组的}和单独输入}加密后一样，得知可以进行爆破，通过改变agent_id长度进行爆破。得到flag



```python
#coding=utf-8
import socket
import string
mission_key = '''xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx'''
a='''12345678901234567890123456789012345678901234567890'''#最初的agent_id
flag='DDCTF{'
msg="flag is: DDCTF{"
for x in range(1,33):#长度32位
    agent=a[:40-x]+'\n'#每次变换agent
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect(('116.85.48.103',5002))
    s.recv(1024)
    s.send(mission_key+'\n')
    s.recv(1024)
    s.send(agent)
    c=s.recv(1024).rstrip()
    for i in string.printable:
        s.recv(1024)
        s.send(msg+i+'\n')
        ss=s.recv(1024).rstrip()[:32]
        if(ss==c[160:192]):
            msg=msg[1:]+i
            flag+=i
            print flag
            break
    s.close()
print flag+'}'
```

0x04 complex stego

题目:

不算提示的提示3：这道题一共有两个解法，之前只提示了一种，另外一种比较简单

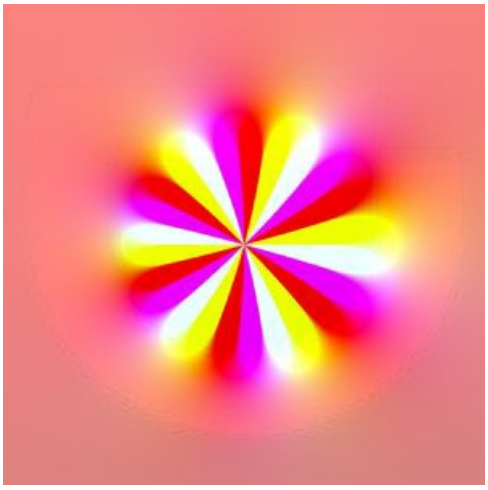
不算提示的提示2：https://www.bilibili.com/video/av19141078
https://www.bilibili.com/video/av21411551
P.S. 出题时间在出视频时间之前，从这个意义上说跟题目没有关系，因此这是不算提示的提示

提示：请善用搜索引擎来学习，重点是z

=====

警方正在利用某黑客使用过的设备，来追查其行踪，并发现了下面附件中的文件。

该黑客利用附件中的encrypt.py，将其密钥转换成encrypted.bmp保存。警方在测试时，发现利用encrypt.py字符串“DDCTF{}”能被转换成下图的美丽图样。



你能帮助警方从encrypted.bmp恢复出该黑客的密钥么？（为便于修正误差，本题flag后面重复出现了一次flag本体）

题解:根据两张图片和一些我自定义的字符串进行加密，得到字符串长7位时每种颜色出现6次，8位时出现7次，数出加密图片颜色次数位37次可以确定加密的字符串长38位。根据题意flag本体重复了一次所以加密字符串格式为DDCTF{15位} 15位，还有一位一直不知道是啥，现在知道是空格了

直接引用出题人的解答吧，毕竟我连傅里叶变换都没学过。只知道解题脚本是用那个微分方程写出来的。

https://github.com/garzon/DDCTF_2018/tree/master/complex_stego

根据函数名z_encrypt、题目名complex、提示（傅里叶变换视频）、搜索引擎或者数学功底，了解到这是一个z变换，encrypted.png为其函数图象，参照https://en.wikipedia.org/wiki/Z-transform#Inverse_Z-transform 中的路径积分公式

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{+\pi} X(e^{j\omega}) e^{j\omega n} d\omega.$$

计算即可。在单位圆上面选取1000个点计算数值积分即可。

贴一下出题人的解题脚本

```python
from PIL import Image
import cmath, math, random, string

maxLen = 45
img_size = 800
abs_cor_size = 1.05
maxValue = 2200.0

def linear_map(v, old_dbound, old_ubound, new_dbound, new_ubound):
    return (v-old_dbound)*1.0/(old_ubound-old_dbound)*(new_ubound-new_dbound) + new_dbound

def read_from_img(img):
    def mapping(r, i):
        x = int(round(linear_map(r, -abs_cor_size, abs_cor_size, 0, img_size)))
        y = int(round(linear_map(i, -abs_cor_size, abs_cor_size, img_size, 0)))
        return (x, y)
    def X(z):
        pix_pos = mapping(z.real, z.imag)
        t = img.getpixel(pix_pos)
        return complex(linear_map(t[1], 0, 255, -maxValue, maxValue), linear_map(t[2], 0, 255, -maxValue, maxValue))
    return X

def z_decrypt(X):
    real_max_val = 0.0
    decrypted = []
    for n in xrange(maxLen):
        res = 0.0
        dw = 0.001
        w = -math.pi
        while w < math.pi:
            v = X(pow(math.e, 1j * w))
            if v.real > real_max_val: real_max_val = v.real
            if v.imag > real_max_val: real_max_val = v.imag
            res += v * pow(math.e, 1j * w * n) * dw
            w += dw
        decrypted.append(res/math.pi/2)
    print 'realmax', real_max_val
    err = [math.fabs(j.imag) for j in decrypted]
    res = ''.join([string.printable[j-1] if j <= len(string.printable) and j >= 1 else '?' for i in decrypted for j in [int(round(i.real))]]).strip('?')
    return res

img = Image.open('encrypted.bmp')
X = read_from_img(img)
decrypted = z_decrypt(X)
print len(decrypted), decrypted
print '--------- decrypted! ----------------'
```

```
C:\WINDOWS\system32\cmd.exe
realmax 1475.29411765
38 DDCTF{ZtR@n5FOrm_be7a} ZtR@n5FOrm_be7a
--------- decrypted! ----------------
请按任意键继续. . .
```