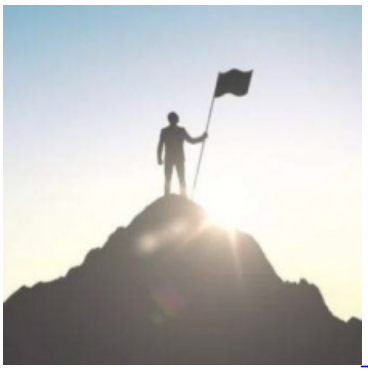
XCTF MISC 几道进阶题

原创

A_dmins 于 2019-12-11 13:50:13 发布 814 收藏 2

分类专栏： CTF题 XCTF

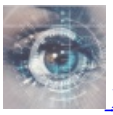
本文链接：https://blog.csdn.net/qq_42967398/article/details/102784401



CTF题 同时被 2 个专栏收录

115 篇文章 11 订阅

订阅专栏



XCTF

24 篇文章 0 订阅

订阅专栏

XCTF MISC 几道进阶题

好久没有打CTF了，做几道misc玩玩吧

小小的PDF

直接binwalk看一下，发现存在三张图片，，，

```
root@kali:~# binwalk  7e5ab2e7587d4a4abf9c705dfb935a92.pdf

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PDF document, version: "1.4"
452           0x1C4           JPEG image data, JFIF standard 1.01
73254         0x11E26         JPEG image data, JFIF standard 1.01
81606         0x13EC6         Zlib compressed data, default compression
82150         0x140E6         JPEG image data, JFIF standard 1.01
104469        0x19815         Zlib compressed data, default compression
105134        0x19AAE         Zlib compressed data, default compression
```


binwalk -e一下没发现有什么东西，，，，

直接使用dd命令吧：`dd if=7e5ab2e7587d4a4abf9c705dfb935a92.pdf of=1 skip=82150 bs=1`

```
root@kali:~# dd if=7e5ab2e7587d4a4abf9c705dfb935a92.pdf of=1 skip=82150 bs=1
26695+0 records in
26695+0 records out
26695 bytes (27 kB, 26 KiB) copied, 0.157018 s, 170 kB/s
root@kali:~#
```

得到flag：

SYC{so_so_so_easy}

不明白以前为什么不会写，，，

Cephalopod

下载下来是一个pcap文件，用strings大法先看看：

```
root@kali:~# strings 1.pcap | grep "flag"
flag.png
flag.png
flag.png
root@kali:~#
```

有图片？？？binwalk一下：

```
root@kali:~# binwalk 1.pcap

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             Libpcap capture file, little-endian, version 2.4,
Ethernet, snaplen: 262144
26441         0x6749          PNG image, 1754 x 2480, 8-bit/color RGBA, non-inte
rlaced
26577         0x67D1          Zlib compressed data, best compression
```

binwalk -e无用，foremost也无用，难受
试试dd命令：`dd if=1.pcap of=1 skip=82150 bs=1`
emmmm，图片出来了但是看不了，无用，，，，，，
在windows下wireshark也无用，没看见能提取的文件，，，
最后查阅资料得知tcpxtract可以从网络流量中提取文件，然而kali上没有，需要安装，，，
安装直接百度其他人的吧，装好直接利用命令：`tcpxtract -f 1.pcap`
能够得到flag图片，，，，，

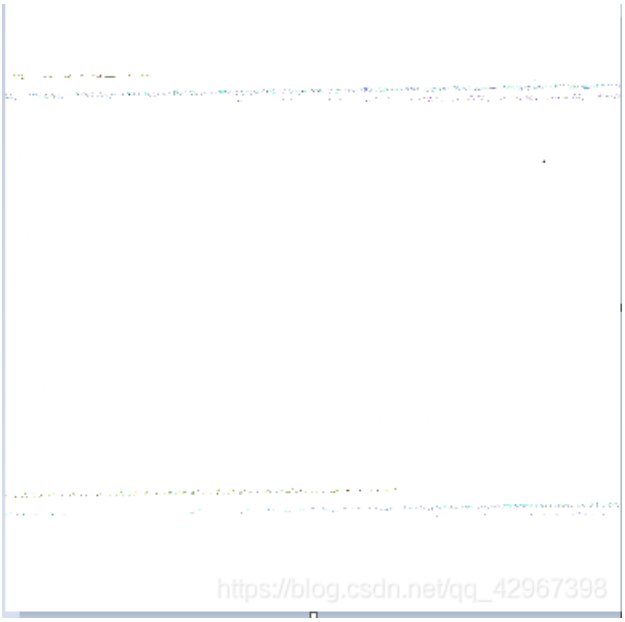

```
HITB{95700d8aefdc1648b90a92f3a8460a2c}
```

misc 2 - 1

下载文件发现图片打不开，拖入winhex，发现文件头不对，修改文件头：

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ANSI ASCII
00000000	80	59	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	€YNG IHDR
00000016	00	00	00	00	00	00	02	F8	08	06	00	00	00	93	2F	8A	ø "/Š
00000032	6B	00	00	00	04	67	41	4D	41	00	00	9C	40	20	0D	E4	k gAMA œ@ ä
00000048	CB	00	00	00	20	63	48	52	4D	00	00	87	0F	00	00	8C	Ë cHRM ‡ Œ
00000064	0F	00	00	FD	52	00	00	81	40	00	00	7D	79	00	00	E9	ýR @ }y é
00000080	8B	00	00	3C	E5	00	00	19	CC	73	3C	85	77	00	00	0A	‹ <å Ìs<…w
00000096	39	69	43	43	50	50	68	6F	74	6F	73	68	6F	70	20	49	9iCCPPhotoshop I
00000112	43	43	20	70	72	6F	66	69	6C	65	00	00	48	C7	9D	96	CC profile HÇ –
00000128	77	54	54	D7	16	87	CF	BD	77	7A	A1	CD	30	D2	19	7A	wTT× ‡Ï½wz¡Í0Ò z

修改之后还是打不开，突然发现表示宽度的地方显示为0？？？
怪不得打不开，，，随便修改一下宽度，然后发现：



没什么用，估计要crc爆破宽度了，，，

```
import struct
import binascii
import os

m = open("1.png","rb").read()

for i in range(0,65535):
    c = m[12:16] + struct.pack('>i', i) + m[20:29]
    crc = binascii.crc32(c) & 0xffffffff
    if crc == 0x932f8a6b:
        print(hex(i))
```

```
python 1.py
0x2c5
```

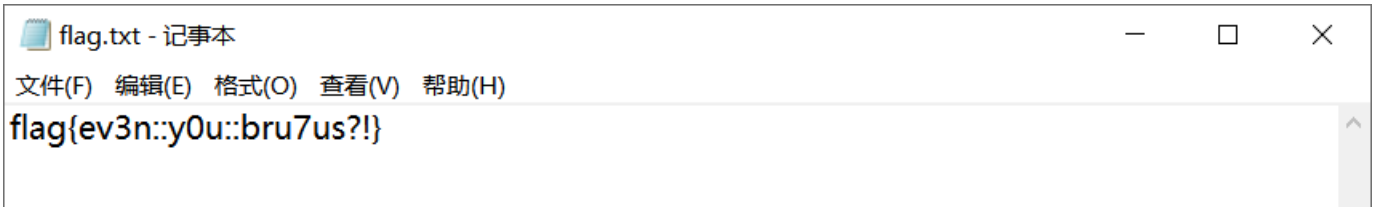
得到flag图片：

János-the-Ripper

下载一个压缩包解压之后一个未命名的文件，查看一下发现是PK：

```
PKETXEOTDC4NULETXNULBSNULSO  DD瞥w'NU
```

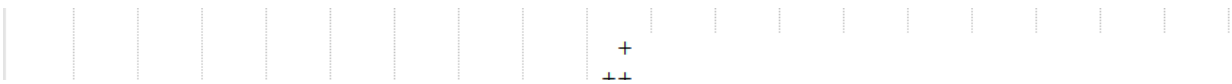
改名为.zip，解压需要密码，，，发现不是伪加密
直接使用工具爆破，得到密码：fish



彻底怀疑自己以前是不是没带脑子，，，，

can_has_stdio?

下载解压得到一个文件，发现：



```
                                          ++
                                          +++
                                         ++[>
                                         +>++>
                                        +++>++
                                        ++>++++
                                       +>++++++
                                       >+++++++>
                                      ++++++++>+
                                      ++++++++>++
                                     +++++++>+++
                                     +++++++>++++
++++++++>++++++++++++++>++++++++++++++>+++++++++++++++>++
  ++++++++++++++<<<<<<<<<<<<<<<<-]>>>>>>>>>>>>>>>--.++<<
    <<<<<<<<<<<>>>>>>>>>>>>>>>----.++++<<<<<<<<<<<<<<<<
      >>>>>>>>>>>>>+.-<<<<<<<<<<<>>>>>>>>>>>>>>-.+<
        <<<<<<<<<<<>>>>>>>>>>>>>>>+++.---<<<<<<
          <<<<<<<<<>>>>>>>>>>>>>>---.+++<<<<<<<<
            <<<<<<>>>>>>>>>>>>>>>+++.---<<<<<<
              <<<<<<<<<>>>>>>>>>>>>>>-.+<<
                <<<<<<<<<<<>>>>>>>>>>>>
                >>----.++++<<<<<<<<<<<<<<
                <>>>>>>>>>>>>+.-<<<<<<<<<
               <<<<>>>>>>>>>>>>>>>--.++<<<
               <<<<<<<<<<<>>>>>>>>>>>>>>-.
              +<<<<<<<<<<<<<>>>>>>>>>>>>>>
              +++.---<<<<<<    <<<<<<<<>>>>
             >>>>>>>>>-.+<        <<<<<<<<<<
             >>>>>>>>>>             >>>--.++<
            <<<<<<<<<                 <<<>>>>>
            >>>>>>                      >>>-.+
           <<<<<                          <<<<<
           <<<                               <>>
          >>                                   >>
>>>>>>>>>++.--<<<<<<<<<<<<<<<>>>>>>>>>>>>>>-.+<<<<<<<<<<<<<<>>>>>>>>>>>>>>--.++<<<<<<<<<<<<<<>>>>>
```

trainfuck编码，，，在线解密：

```
                    <<<                          <>>
                   >>                              >>


>>>>>>>>>++.--<<<<<<<<<<<<<<<>>>>>>>>>>>>>>-.+<<<<<<<<<<<<<<>>>>>>>>>>>>>>--.++<<<<<<<<<<<<<<>>>>>
<<<<<<<<<<<<<<<>>>>>>>>>>>>>>-.+<<<<<<<<<<<<<<>>>>>>>>>>>>>>+.-<<<<<<<<<<<<<<>>>>>>>>>>>>>>--.++<<<<<<<<<<<<<<>>>>>
<<<<<<<<<<<<<<>>>>>>>>>>>>>>>.<<<<<<<<<<<<<<>>>>>>>>>>>>>>>++.--<<<<<<<<<<<<<<>>>>>>>>>>>>>>>-.+<<<<<<<<<<<<<<<>
<<<<<<<<<<<<<<>>>>>>>>>>>>>>>---.++++<<<<<<<<<<<<<<>>>>>>>>>>>>>>>---.+++<<<<<<<<<<<<<<<.
```

字符集 utf8(unicode编码)

加 密　　解 密

```
flag{esolangs_for_fun_and_profit}
```


再次怀疑，，，，

MISCall

下载文件放到kali中使用file查看一下文件：

```
root@kali:~# file 123
123: bzip2 compressed data, block size = 900k
```

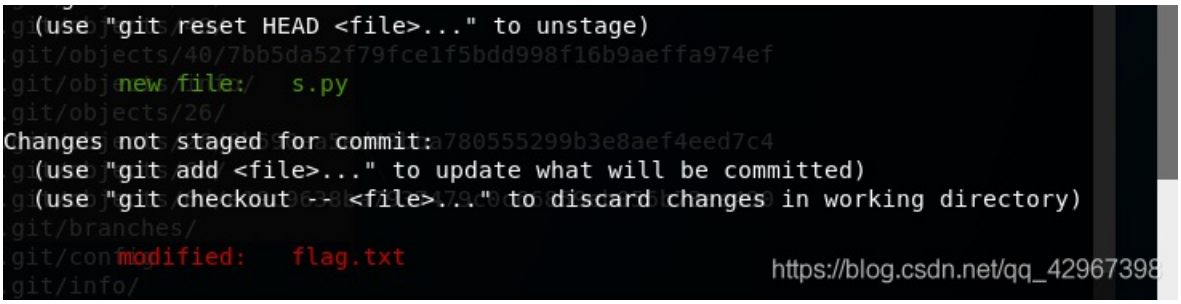
是一个bzip2的压缩文件，直接使用命令：`tar -xvjf 123`的到文件：

```
root@kali:~# tar -xvjf 123
ctf/
ctf/flag.txt
ctf/.git/
ctf/.git/description
ctf/.git/refs/
ctf/.git/refs/heads/
ctf/.git/refs/heads/master
ctf/.git/refs/stash
ctf/.git/refs/tags/
ctf/.git/ORIG_HEAD
ctf/.git/logs/
ctf/.git/logs/refs/
ctf/.git/logs/refs/heads/
ctf/.git/logs/refs/heads/master
ctf/.git/logs/refs/stash
ctf/.git/logs/HEAD
ctf/.git/HEAD
ctf/.git/COMMIT_EDITMSG
ctf/.git/hooks/
ctf/.git/hooks/pre-commit.sample
ctf/.git/hooks/update.sample
ctf/.git/hooks/applypatch-msg.sample
ctf/.git/hooks/pre-applypatch.sample
```


好像是.git目录？？？进入目录看一下：

好像“.”开头的文件都被隐藏了？？？flag.txt中没有flag，估摸着.git中隐藏了什么东西。。。。
查看了一下.git列表好像没有什么特别的地方，，，，
最后得知一个git stash命令，
git stash会把所有未提交的修改（包括暂存的和非暂存的）都保存起来，用于后续恢复当前工作目录
查看现有stash：git stash list
查看列表：git stash show
复原文件：git stash apply

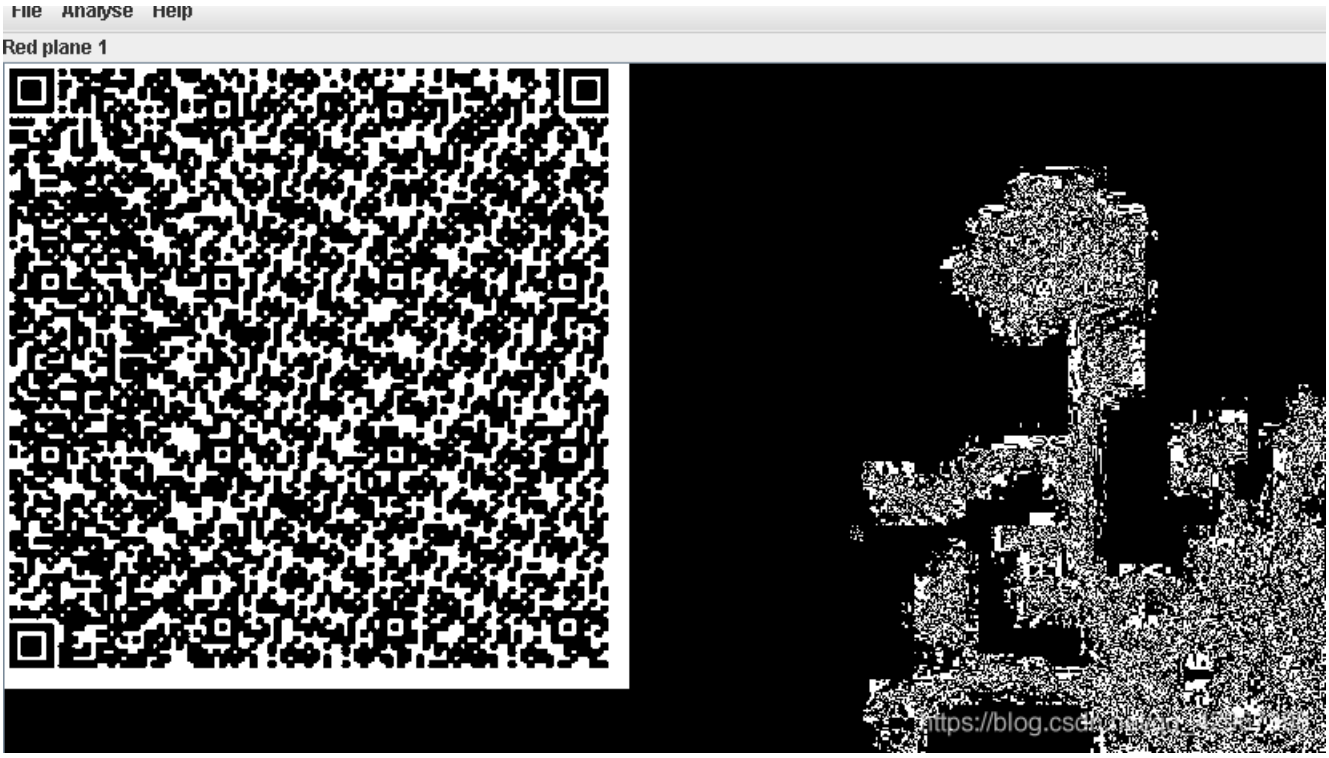
```
root@kali:~/ctf# git stash list
stash@{0}: WIP on master: bea99b9 Initial commit
root@kali:~/ctf# git stash show
 flag.txt | 25 ++++++++++++++++++++++++
 s.py     |  4 ++++
 2 files changed, 28 insertions(+), 1 deletion(-)
root@kali:~/ctf# git stash apply
On branch master
Changes to be committed:
```



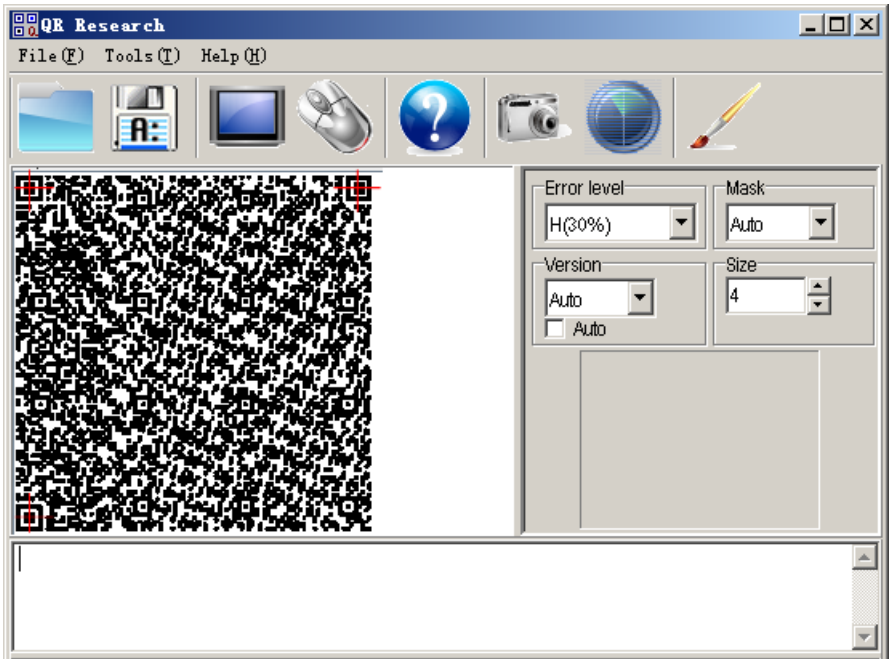
出现一个s.py文件，运行得到flag，，，，，

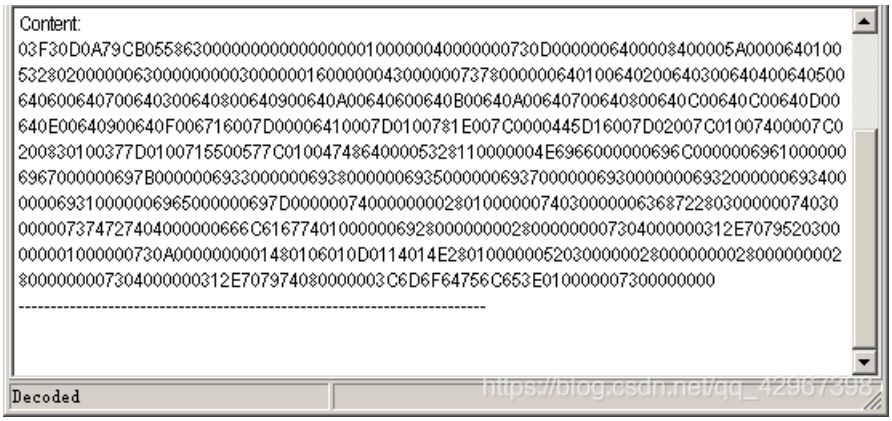
适合作为桌面

下载文件解压发现是一张图片！！放入stego中查看一下，发现存在二维码：



利用二维码扫描工具扫到内容：





扣下来：

```
03F30D0A79CB055863000000000000000001000000400000000730D00000064000084000005A000064010053280200000063000000000300000016000000430000007378000000640100640200640300640400640500640600640700640300640800640900640A00640600640B00640A00640700640800640C00640C00640D00640E00640900640F006716007D00006410007D0100781E007C0000445D16007D02007C01007400007C0200830100377D0100715500577C0100474864000053281100000004E6966000000696C0000006961000000696700000069 7B0000006933000000693800000069350000006937000000693000000069320000006934000000693100000069650000006 97D0000007400000000 2801000000740300000006368722803000000074030000007374727404000000666C6167740100000069280000000028000000007304000000312E707952030000000100000073 0A0000000001480106010D0114014E2801000000520300000002800000000280000000028000000007304000000312E707974080000003C6D6F64756C653E01000000730000000 0
```

怀疑是一个文件之类的，查找一下是否为文件头：

原来是pyc的文件头，直接用winhex保存为pyc文件
利用在线反编译工具的到源码：

```python
def flag():
    str = [
        102,
        108,
        97,
        103,
        123,
        51,
        56,
        97,
        53,
        55,
        48,
        51,
        50,
        48,
        56,
        53,
        52,
        52,
        49,
        101,
        55,
        125]
    flag = ''
    for i in str:
        flag += chr(i)

    print flag
```

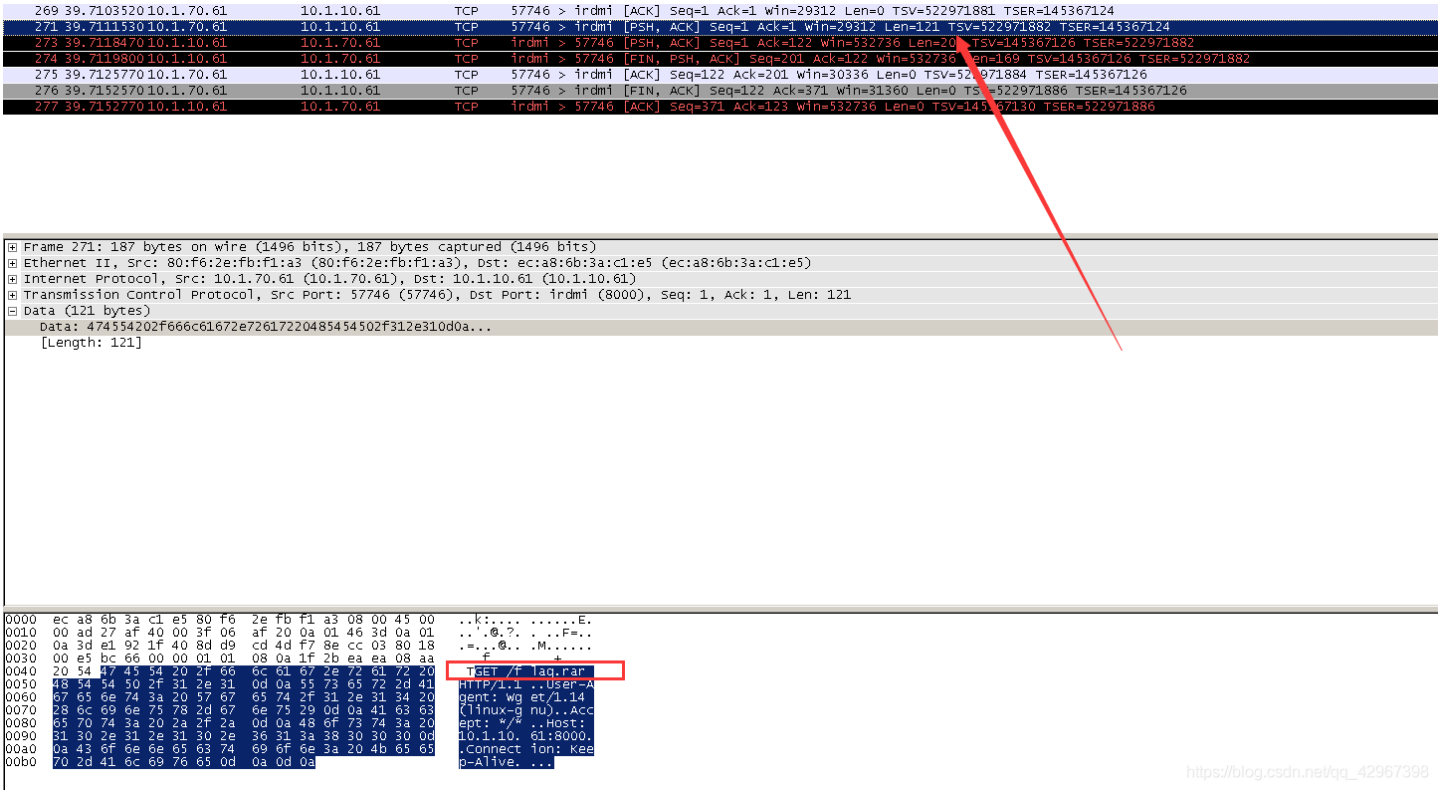
直接运行一下得到flag！

```
(base) C:\Users\Administrator\Desktop>python 1.py
flag{38a57032085441e7}
```

misc 3 - 1

下载文件下来，发现是一个rar压缩包，解压得到一个文件，发现是一个pcap文件

wireshark打开文件来，没找到什么，直接搜索字符串flag找到一个flag.rar文件？？



save下来，发现需要密码？？？难受了，返回去继续看wireshark，emmmm

在tcp.stream eq 6中发现可疑内容：

```
[root@localhost wireshark]# llss

1  2  3  test
[root@localhost wireshark]# ccaatt  11

Rar!....3...
.............TU..<..... .+......flag.txt0.....n.Kr..z....uEo.Bn&=i.S..>....4.B..~...xj.".
...u......3.....jWj..%m..!.+h...+s..q#.]...3Ks.y.....r.2...wVQ....[root@localhost wireshark]# ccaatt  22

19aaFYsQQKr+hVX6hl2smAUQ5a767TsULEUebWSajEo=[root@localhost wireshark]# ppiinngg  bbaaiidduu..ccoomm

PING baidu.com (111.13.101.208) 56(84) bytes of data.
64 bytes from 111.13.101.208 (111.13.101.208): icmp_seq=1 ttl=48 time=33.4 ms
64 bytes from 111.13.101.208 (111.13.101.208): icmp_seq=2 ttl=48 time=32.1 ms
64 bytes from 111.13.101.208 (111.13.101.208): icmp_seq=3 ttl=48 time=34.7 ms
64 bytes from 111.13.101.208 (111.13.101.208): icmp_seq=4 ttl=48 time=31.9 ms
...^C
--- baidu.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3037ms
rtt min/avg/max/mdev = 31.921/33.067/34.784/1.155 ms
[root@localhost wireshark]# ccaatt  33

# coding:utf-8
.


.
__author__ = 'YFP'
.
```

```
from Crypto import Random

from Crypto.Cipher import AES


import sys

import base64


IV = 'QWERTYUIOPASDFGH'


def decrypt(encrypted):

  aes = AES.new(IV, AES.MODE_CBC, IV)

  return aes.decrypt(encrypted)


def encrypt(message):

  length = 16

  count = len(message)

  padding = length - (count % length)

  message = message + '\0' * padding

  aes = AES.new(IV, AES.MODE_CBC, IV)

  return aes.encrypt(message)


str = 'this is a test'


example = encrypt(str)


print(decrypt(example))
```

一串字符串：19aaFYsQQKr+hVX6hl2smAUQ5a767TsULEUebWSajEo=
还有一个python脚本，，，，emmmm，估摸着要我们解密！！
直接修改一下脚本：

```python
# coding:utf-8
__author__ = 'YFP'
from Crypto import Random
from Crypto.Cipher import AES
import sys
import base64
IV = 'QWERTYUIOPASDFGH'

def decrypt(encrypted):
 aes = AES.new(IV, AES.MODE_CBC, IV)
 return aes.decrypt(encrypted)

def encrypt(message):
 length = 16
 count = len(message)
 padding = length - (count % length)
 message = message + '\0' * padding
 aes = AES.new(IV, AES.MODE_CBC, IV)
 return aes.encrypt(message)

example = base64.b64decode("19aaFYsQQKr+hVX6hl2smAUQ5a767TsULEUebWSajEo=")
print(decrypt(example))
```

运行得到：

```
(base) C:\Users\Administrator\Desktop>python 1.py
b'passwd={No_One_Can_Decrypt_Me}\x00\x00'
```

得到解压密码：`No_One_Can_Decrypt_Me`
解压得到flag，，，