picoCTF2018 Writeup之Forensics篇

Stefano / 2018-10-31 06:52:00 / 浏览数 3260 安全技术 CTF 顶(0) 踩(0)

前言

本篇是Forensic，即取证类的题解。包括有一些常规的入门misc题目，比如流量分析，图片、文件隐写之类的。部分题目附件已打包。链接: https://pan.baidu.com/s/1kfx8GEnio7V1Z5Fa4TrfMQ 提取码: 5a9u

（拖了半个月终于把这篇补完了，还好题目还开着，拖延症彻底没救了-0-……）

Forensics Warmup 1

Question

Can you unzip this file for me and retreive the flag?

Hint

Make sure to submit the flag as picoCTF{XXXXX}

Solution

签到题，解压zip，得到`flag.jpg`

输入图上的字符就行了。

flag:`picoCTF{welcone_to_forensics}`

Forensics Warmup 2

Question

Hmm for some reason I can't open this PNG? Any ideas?

Hint

How do operating systems know what kind of file it is? (It's not just the ending!

Make sure to submit the flag as picoCTF{XXXXX}

Solution

使用`file`命令查看图片的文件格式。

```
› file flag.png
flag.png: JPEG image data, JFIF standard 1.01, resolution (DPI), density 75x75, segment length 16, baseline, precision 8, 909x
```

本质是jpg文件，修改后缀为.jpg就可以打开了。（其实大多的图片浏览器都可以直接打开这种单纯修改一下后缀的图片）

flag:`picoCTF{extensions_are_a_lie}`

Desrouleaux

Question

Our network administrator is having some trouble handling the tickets for all of of our incidents. Can you help him out by answering all the questions? Connect with `nc 2018shell1.picoctf.com 54782`. incidents.json

Hint

If you need to code, python has some good libraries for it.

Solution

文件里面是一段json数据。

```
...
{
            "ticket_id": 0,
            "timestamp": "2015/05/09 22:28:20",
            "file_hash": "b807c12fc3e10ba3",
            "src_ip": "248.63.150.241",
            "dst_ip": "251.0.92.254"
        },
        {
            "ticket_id": 1,
            "timestamp": "2016/12/27 04:01:52",
            "file_hash": "1698b8b87f51ce8e",
            "src_ip": "248.63.150.241",
            "dst_ip": "116.196.246.151"
        },
}
...
```

nc到问题服务器，一步步处理后发现有三个问题：

1. What is the most common source IP address?
2. How many unique destination IP addresses were targeted by the source IP address 236.232.221.165?
3. What is the average number of unique destination IP addresses that were sent a file with the same hash? Your answer needs to be correct to 2 decimal places.

第三个有点拗口，大概意思就是要算出每个单独的文件发送到的ip地址个数的平均数是多少。使用python的list和dict对象可以很方便的处理这些数据。

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-


import json
import re
from pwn import *
with open('./incidents.json') as f:
   text = f.read()
text = json.loads(text)

r = remote('2018shell2.picoctf.com', 54782)

src_ip_count = {}
dst_ip_count = []
unique_dst_ip = 0
hash_dst_ip = {}

r.recvuntil('common ones.')
for i in text['tickets']:

   # Question 1
   if i['src_ip'] not in src_ip_count.keys():
       src_ip_count[i['src_ip']] = 1
   else:
       src_ip_count[i['src_ip']] += 1


r.sendline(max(src_ip_count))
content = r.recvuntil('?\n')
content = re.findall(r'address \d{1,3}.\d{1,3}.\d{1,3}.\d{1,3} ?',content)
content = ''.join(content).split(' ')[1]
```

```
for i in text['tickets']:

   # Question 2
   if i['dst_ip'] not in dst_ip_count and i['src_ip'] == content:
       dst_ip_count.append(i['dst_ip'])
       unique_dst_ip += 1

   # Question 3
   if i['file_hash'] not in hash_dst_ip.keys() and i['file_hash'] != []:
       hash_dst_ip[i['file_hash']] = []
       hash_dst_ip[i['file_hash']].append(i['dst_ip'])
   elif i['file_hash'] in hash_dst_ip.keys() and i['file_hash'] != []:
       hash_dst_ip[i['file_hash']].append(i['dst_ip'])


avg = 0
for i in hash_dst_ip:
   avg += len(hash_dst_ip[i])
avg = avg * 1.0 / len(hash_dst_ip)


# print unique_dst_ip
# print round(avg, 2)

r.sendline(str(unique_dst_ip))
r.sendline(str(round(avg, 2)))
# r.interactive()
print r.recvuntil('}\n')

r.close()
```

运行脚本得到flag。

flag:`picoCTF{J4y_s0n_d3rUUUULo_c74e3495}`

Reading Between the Eyes

Question

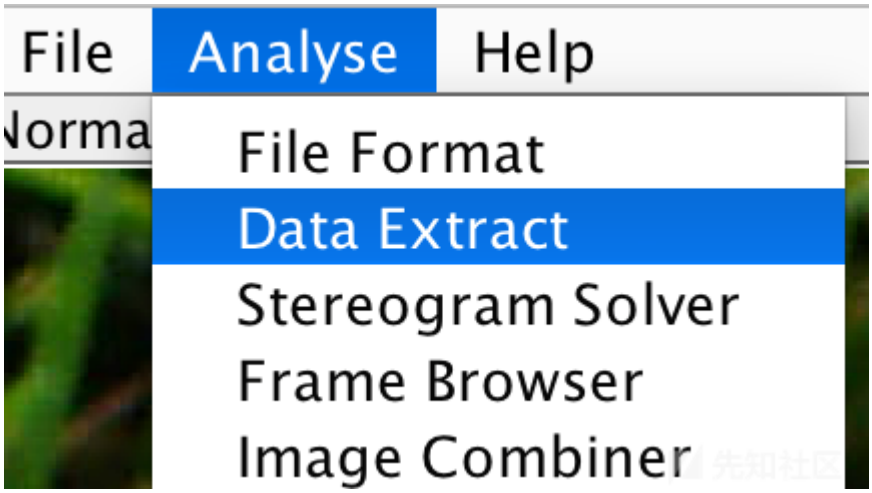
Stego-Saurus hid a message for you in this image, can you retreive it?

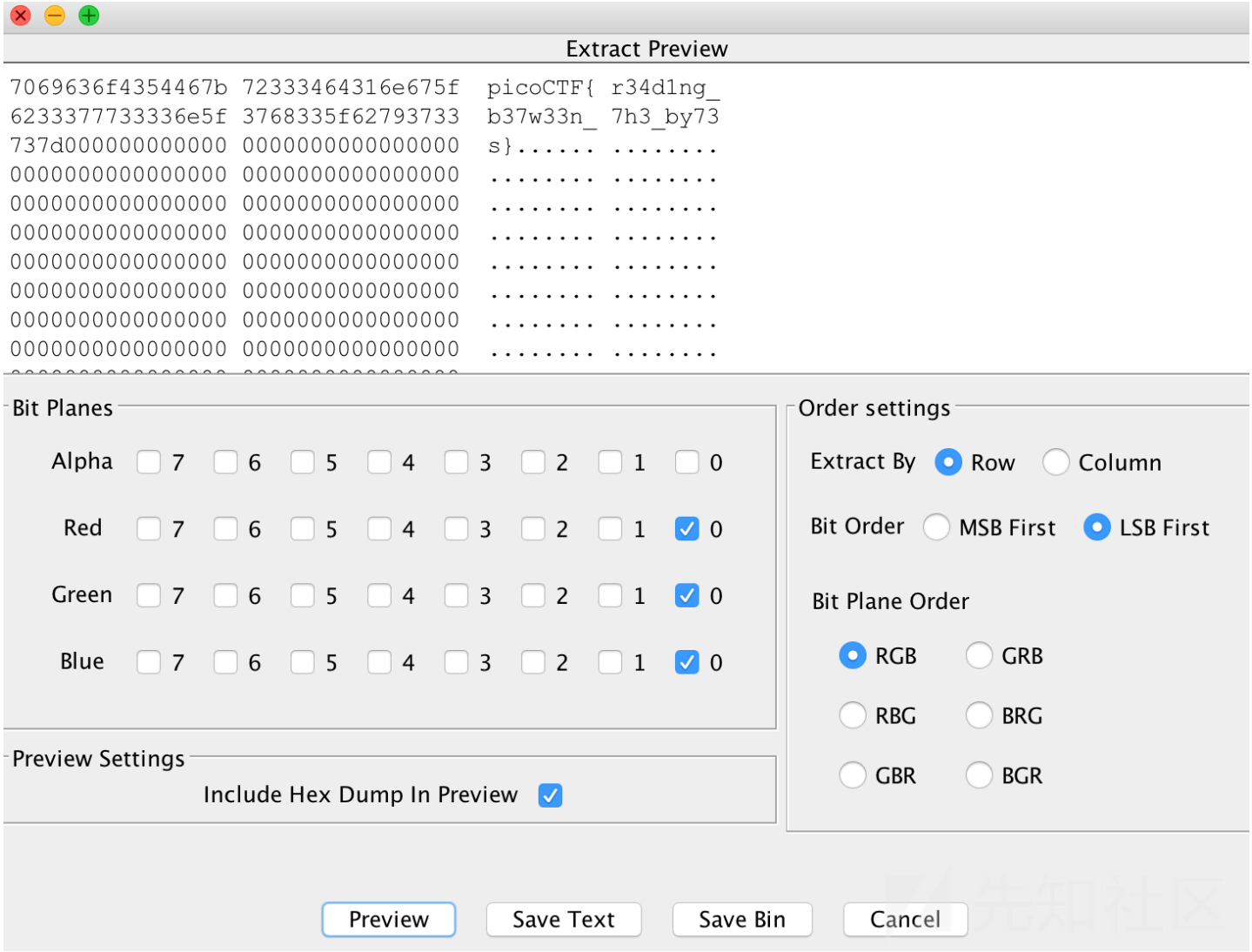
Hint

Maybe you can find an online decoder?

Solution

LSB隐写，使用`stegsolve`提取最低位。





勾选RGB信道的最低位0，顺序是LSB First，就可以看到flag。

flag:`picoCTF{r34d1ng_b37w33n_7h3_by73s}`

Recovering From the Snap

Question

There used to be a bunch of animals here, what did Dr. Xernon do to them?

Hint

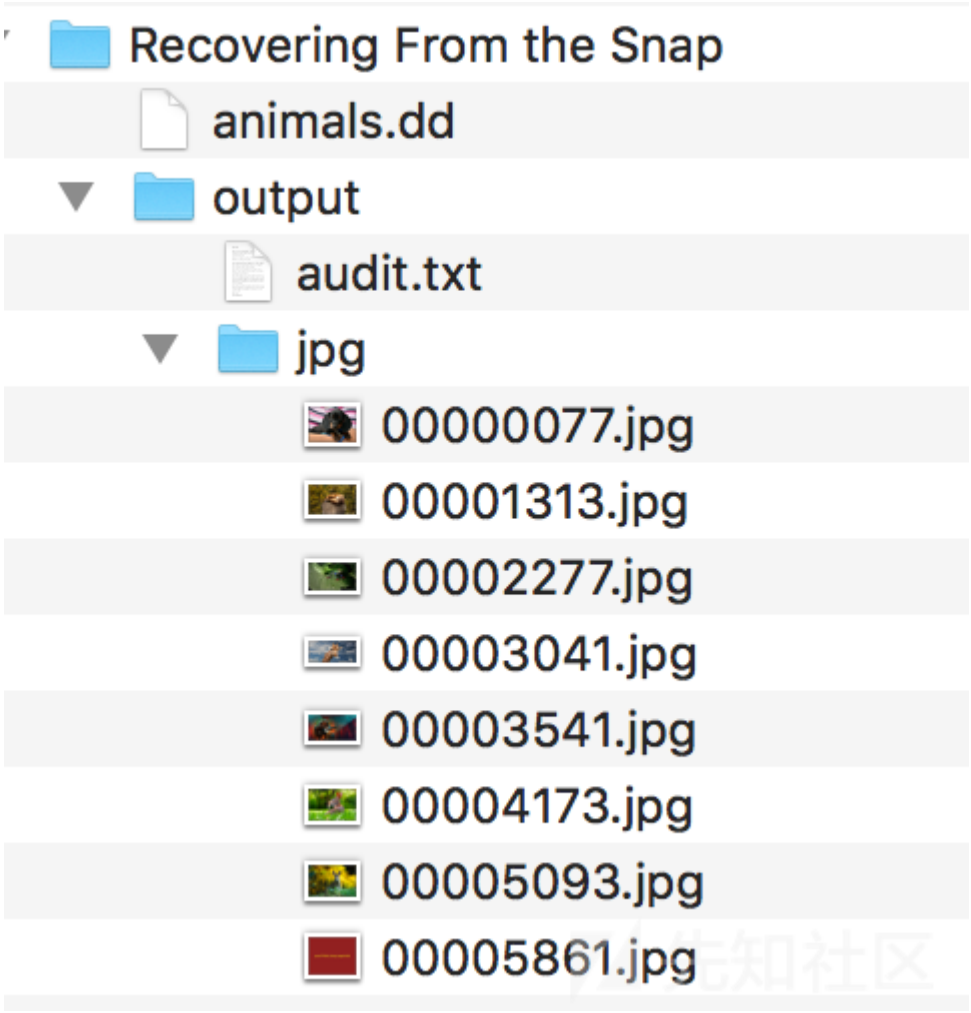
Some files have been deleted from the disk image, but are they really gone?.

Solution

使用`binwalk`查看文件，发现有很多隐写进去的jpg文件，用`foremost`提取出来。

```
› foremost animals.dd
foremost: /usr/local/etc/foremost.conf: No such file or directory
Processing: animals.dd
|*|
```

提取完成查看图片，可以看到最后一张就是flag。



flag:`picoCTF{th3_5n4p_happ3n3d}`

admin panel

Question

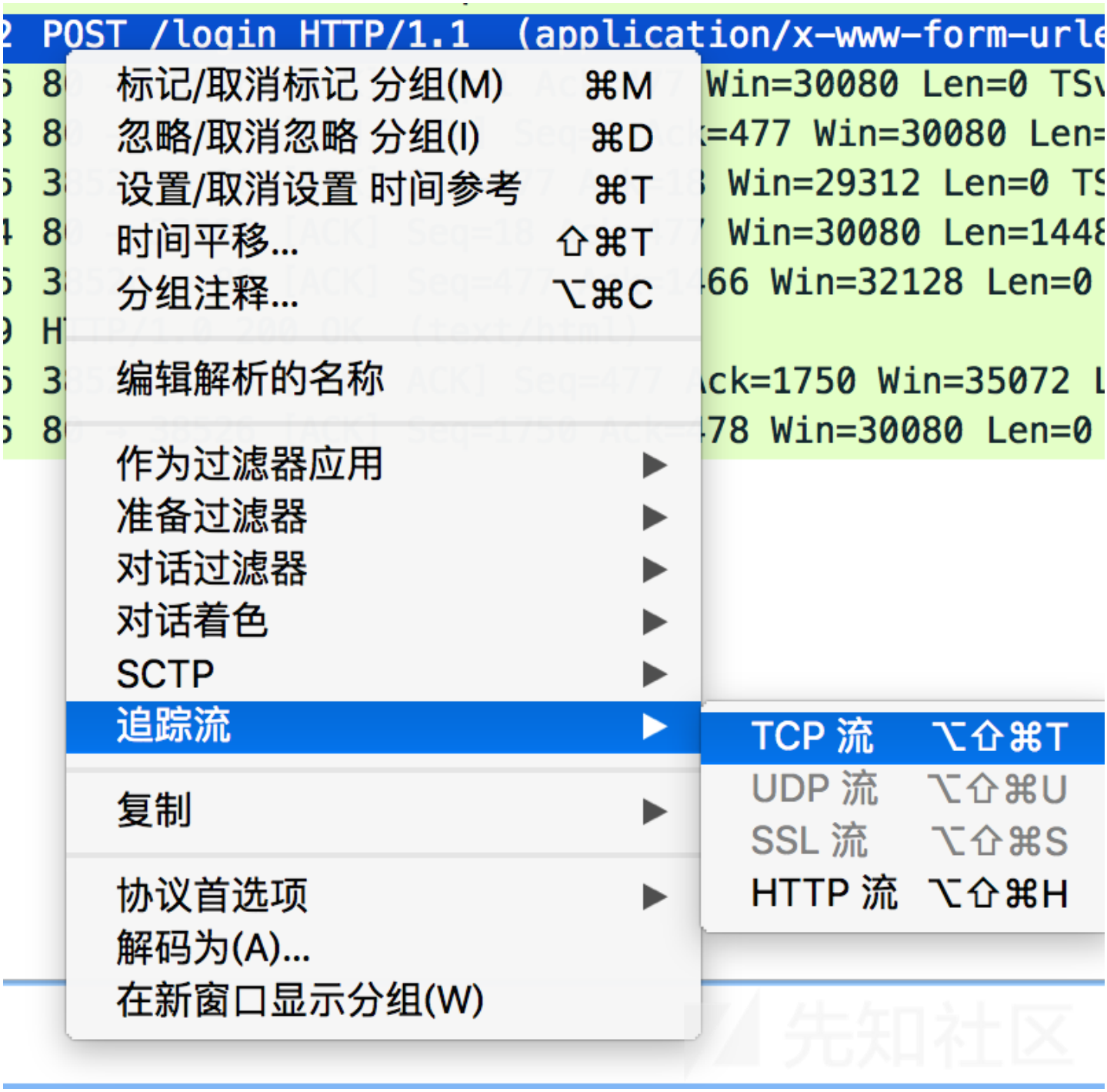
We captured some traffic logging into the admin panel, can you find the password?

Hint

Tools like wireshark are pretty good for analyzing pcap files.

Solution

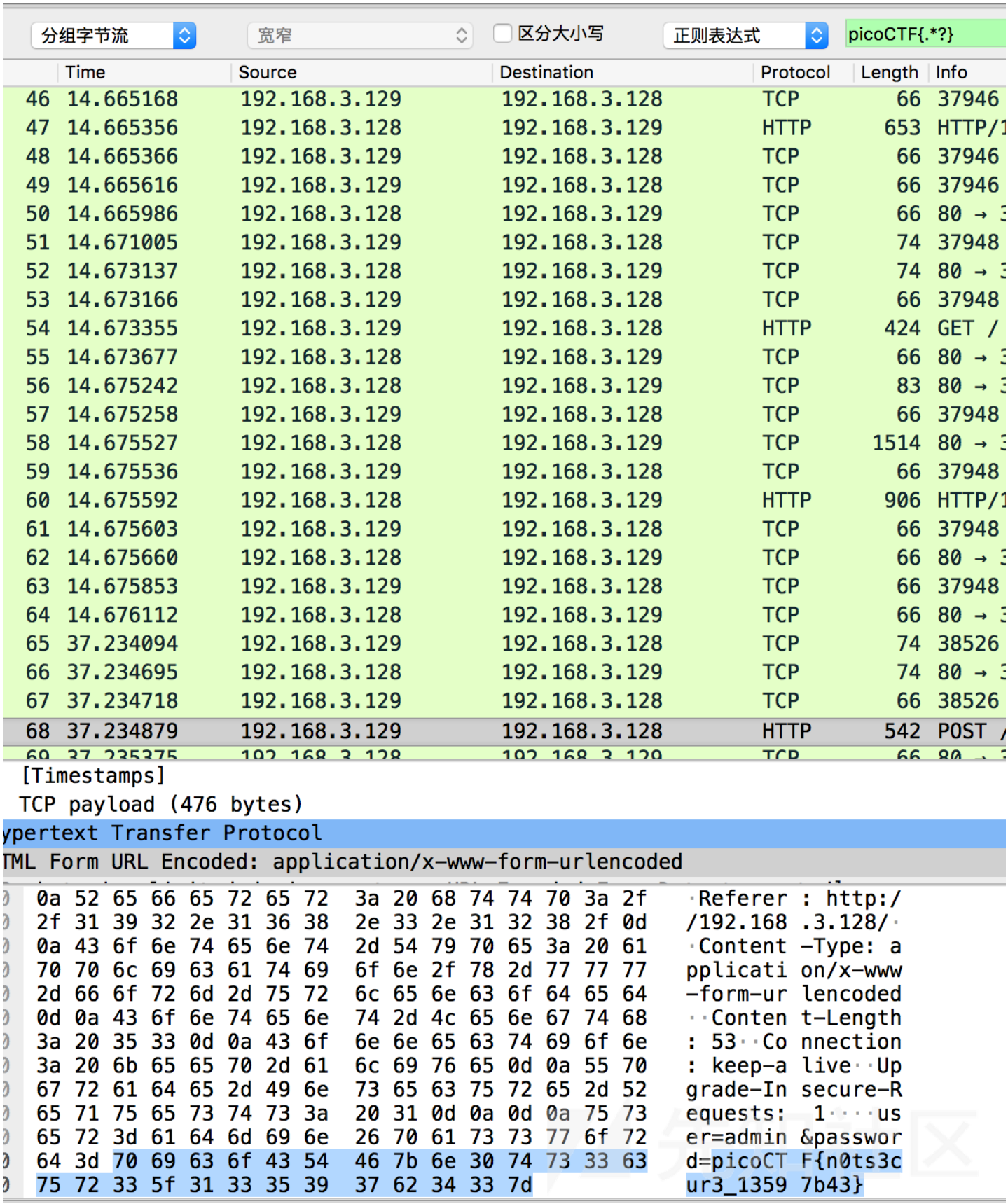
使用wireshrak打开pcap流量包，追踪tcp流，在第5个流可以看到flag。




```
POST /login HTTP/1.1
Host: 192.168.3.128
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:59.0) Gecko/20100101 F
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.3.128/
Content-Type: application/x-www-form-urlencoded
Content-Length: 53
Connection: keep-alive
Upgrade-Insecure-Requests: 1

user=admin&password=picoCTF{n0ts3cur3_13597b43}HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
```

当然这里我们知道flag的形式是picoCTF{***}，可以直接在分组字节流中搜索相关字符串，同样也可以找到flag。



flag:picoCTF{n0ts3cur3_13597b43}

hex editor

Question

This cat has a secret to teach you. You can also find the file in /problems/hex-editor_2_c1a99aee8d919f6e42697662d798f0ff on the shell server.

Hint

What is a hex editor?

Maybe google knows.

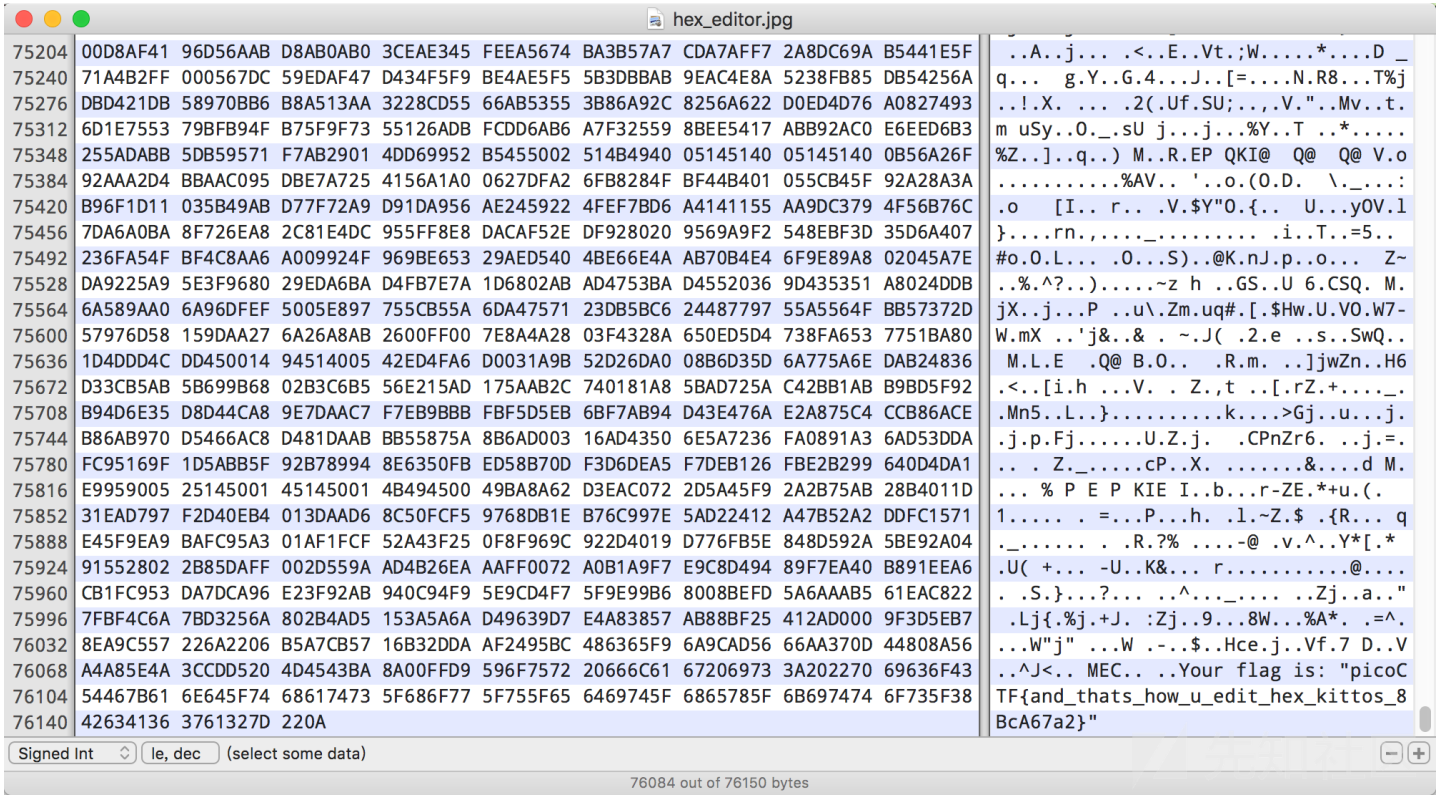
xxd

hexedit

bvi

Solution

使用16进制编辑器查看文件16进制内容，windows环境下可以使用winhex、010editor，mac环境下可以使用hex friend。



可以看到到jpg文件格式的结尾FF D9之后还有一串字符，就是我们要的flag。

flag:picoCTF{and_thats_how_u_edit_hex_kittos_8BcA67a2}

Truly an Artist

Question

Can you help us find the flag in this Meta-Material? You can also find the file in /problems/truly-an-artist_3_066d6319e350c1d579e5cf32e326ba02.

Hint

Try looking beyond the image.

Who created this?

Solution

文件属性隐藏了信息，windows下直接右键查看文件属性就可以看到，mac可以使用exiftool查看。

```
› exiftool 2018.png
ExifTool Version Number         : 11.11
File Name                       : 2018.png
Directory                       : .
File Size                       : 13 kB
File Modification Date/Time     : 2018:10:28 01:06:38+08:00
File Access Date/Time           : 2018:10:28 01:06:50+08:00
```

```
File Inode Change Date/Time     : 2018:10:28 01:06:51+08:00
File Permissions                : rw-r--r--
File Type                       : PNG
File Type Extension             : png
MIME Type                       : image/png
Image Width                     : 1200
Image Height                    : 630
Bit Depth                       : 8
Color Type                      : RGB
Compression                     : Deflate/Inflate
Filter                          : Adaptive
Interlace                       : Noninterlaced
Artist                          : picoCTF{look_in_image_7e31505f}
Image Size                      : 1200x630
Megapixels                      : 0.756
```

flag:`picoCTF{look_in_image_7e31505f}`

now you don't

Question

We heard that there is something hidden in this picture. Can you find it?

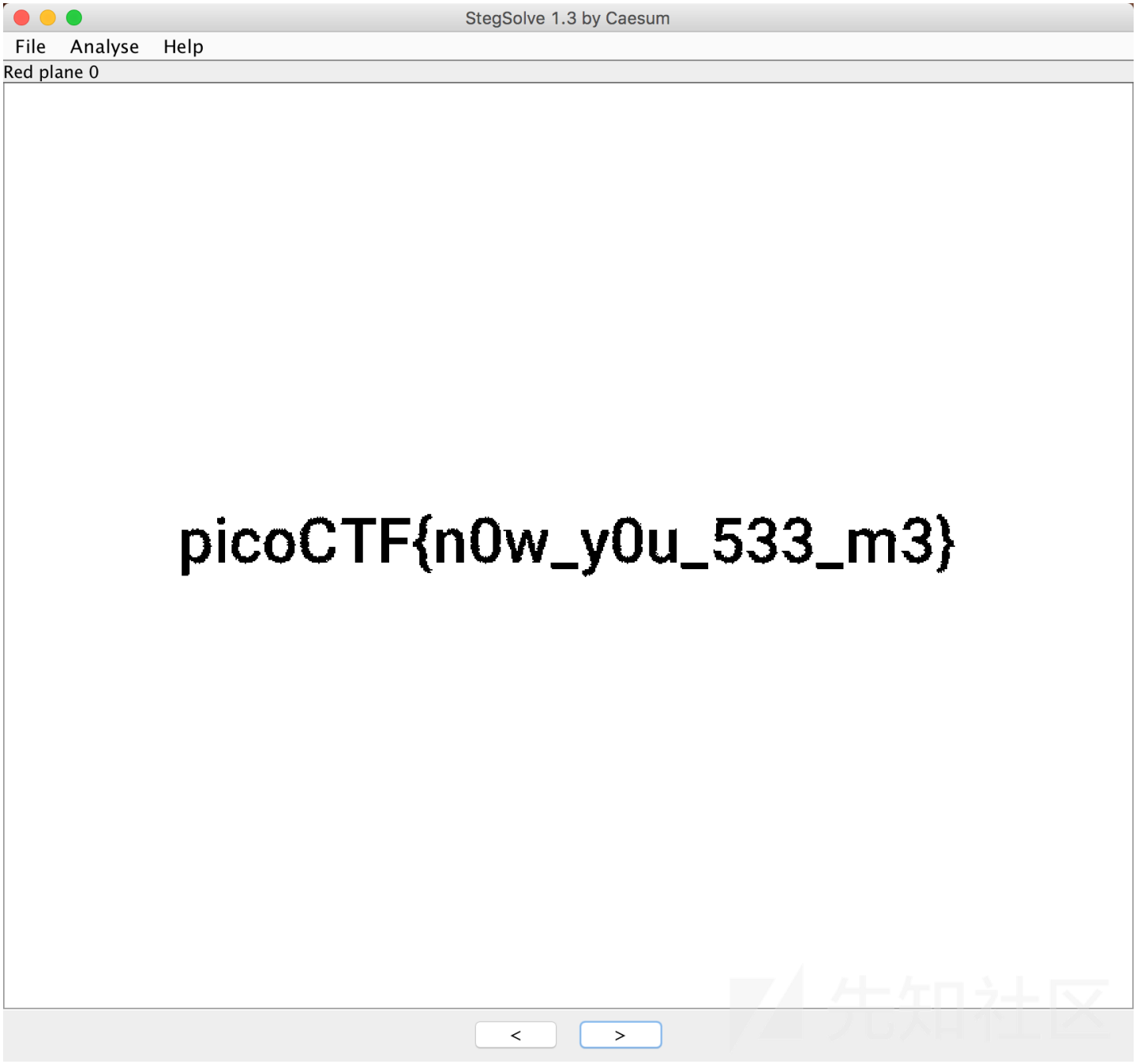
Hint

There is an old saying: if you want to hide the treasure, put it in plain sight. Then no one will see it.

Is it really all one shade of red?

Solution

还是用`stegsolve`打开文件，查看不同信道下的图片，发现flag在隐藏在红色0信道中。



flag:`picoCTF{n0w_y0u533_m3}`

Ext Super Magic

Question

> We salvaged a ruined Ext SuperMagic II-class mech recently and pulled the filesystem out of the black box. It looks a bit corrupted, but maybe there's something interesting in there. You can also find it in /problems/ext-super-magic_4_f196e59a80c3fdac37cc2f331692ef13 on the shell server.

Hint

> Are there any tools for diagnosing corrupted filesystems? What do they say if you run them on this one?
>
> How does a linux machine know what type of file a file is?
>
> You might find this doc helpful.
>
> Be careful with endianness when making edits.
>
> Once you've fixed the corruption, you can use /sbin/debugfs to pull the flag file out.

Solution

给了一个镜像文件，file命令查看一下，发现不能识别。debugfs可以识别出镜像损坏的部分。

```
› file ext-super-magic.img
ext-super-magic.img: data

› debugfs ext-super-magic.img
debugfs 1.44.1 (24-Mar-2018)
Checksum errors in superblock!  Retrying...
ext-super-magic.img: Bad magic number in super-block while opening filesystem
```

依据提示查看ext2镜像格式的文档，我们需要在superblock的第56和57字节之间插入magic number，即0xef53。

```
› file fixed.img
fixed.img: Linux rev 1.0 ext2 filesystem data, UUID=fad5d44e-2bb4-4c22-9410-79b020de84dd (large files)
```

修复成功，挂载打开，发现有很多的jpg文件，其中有一张flag.jpg里面就有flag。

flag:picoCTF{a7DB29eCf7dB9960f0A19Fdde9d00Af0}

Lying Out

Question

> Some odd traffic has been detected on the network, can you identify it? More info here. Connect with `nc 2018shell1.picoctf.com 50875` to help us answer some questions.

Hint

No Hints.

Solution

需要分析异常访问流量，对照图片一个个看就可以了。

```
You'll need to consult the file `traffic.png` to answer the following questions.
Which of these logs have significantly higher traffic than is usual for their time of day? You can see usual traffic on the at
   log_ID      time  num_IPs
0       0  00:00:00     9552
1       1  02:30:00    11573
2       2  06:00:00    10381
3       3  07:00:00    11674
4       4  07:00:00    10224
5       5  07:30:00    10966
6       6  16:00:00     9685
7       7  17:45:00    15875
8       8  18:00:00    11889
9       9  19:15:00    11935
10     10  19:30:00    11191
11     11  20:30:00     9952
12     12  20:45:00     9898
13     13  22:45:00    11609
1 3 7 13
Correct!
Great job. You've earned the flag: picoCTF{w4y_0ut_ff5bd19c}
```

flag:picoCTF{w4y_0ut_ff5bd19c}

What's My Name?

Question

Say my name, say my name.

Hint

If you visited a website at an IP address, how does it know the name of the domain?

Solution

提示给的很明显了，我们需要查找和DNS有关的信息流，在wireshark过滤器中搜寻dns流量，就可以看到flag了。

dns

No.	Time	Source	Destination	Protocol	Length
43	1418.341495	192.168.2.1	192.168.2.12	DNS	8
55	1418.342859	192.168.2.12	192.168.2.1	DNS	31

```
        Time to live: 300
        Data length: 55
        TXT Length: 54
        TXT: picoCTF{w4lt3r_wh1t3_33ddc9bcc77f22a319515c59736f64a2}
```

```
0050  00 01 00 00 01 2c 00 04  c0 a8 02 0d c0 0c 00 05   ·····,·· ········
0060  00 01 00 00 01 2c 00 09  06 6d 79 6e 61 6d 65 c0   ·····,·· ·myname·
0070  19 c0 0c 00 0f 00 01 00  00 01 2c 00 04 00 05 c0   ········ ··,·····
0080  3e c0 0c 00 0f 00 01 00  00 01 2c 00 08 00 0a 03   >······ ··,·····
0090  6d 78 32 c0 3e c0 0c 00  0f 00 01 00 00 01 2c 00   mx2·>··· ·····,·
00a0  08 00 14 03 6d 78 33 c0  3e c0 0c 00 02 00 01 00   ····mx3· >·····
00b0  01 51 80 00 06 03 6e 73  31 c0 3e c0 0c 00 02 00   ·Q····ns 1·>·····
00c0  01 00 01 51 80 00 06 03  6e 73 32 c0 3e c0 0c 00   ···Q···· ns2·>···
00d0  10 00 01 00 00 01 2c 00  37 36 70 69 63 6f 43 54   ·····,· 76picoCT
00e0  46 7b 77 34 6c 74 33 72  5f 77 68 31 74 33 5f 33   F{w4lt3r _wh1t3_3
00f0  33 64 64 63 39 62 63 63  37 37 66 32 32 61 33 31   3ddc9bcc 77f22a31
0100  39 35 31 35 63 35 39 37  33 36 66 36 34 61 32 7d   9515c597 36f64a2}
0110  c0 0c 00 06 00 01 00 01  51 80 00 20 03 6e 73 31   ········ Q·· ·ns1
```

同样也可以通过字符串搜索的方式找到flag，方法和`admin panel`一样。

Malware Shopscore

There has been some malware detected, can you help with the analysis? More info here. Connect with `nc 2018shell1.picoctf.com 18874`.

Hint

No Hints.

Solution

还是看图识别数据，和`Lying Out`差不大多，第一个问题看图片找特征，第二个问题看哪个文件与题目问的文件的`jmp_count`和`add_count`的数值相近就选哪个。

```
You'll need to consult the file `clusters.png` to answer the following questions.


How many attackers created the malware in this dataset?
5
Correct!


In the following sample of files from the larger dataset, which file was made by the same attacker who made the file 628e79cf?
      hash  jmp_count  add_count
0  628e79cf       17.0       18.0
1  1f2c7915       18.0       60.0
2  6e7d554a       10.0       42.0
3  a55f572c       30.0       37.0
4  f118fcd7       36.0       13.0
5  97b1425e       35.0       30.0
6  a163e543       18.0       71.0
7  ebaf5ccd       11.0       18.0
8  9059414f       38.0       13.0
9  c30ea3fe       18.0       37.0
ebaf5ccd
Correct!


Great job. You've earned the flag: picoCTF{w4y_0ut_dea1794b}
```