【转载】安恒八月月赛流量分析writeup

转载

gclome 于 2020-08-25 11:06:04 发布 1767 收藏 8

分类专栏： # CTF

原文链接：https://www.cnblogs.com/sn1per/p/12553064.html

本文转载自：https://www.cnblogs.com/sn1per/p/12553064.html

这篇文章我是看了上面的链接，自己操作了一遍，有些地方或许会有点不同·

一、题目背景

某公司内网网络被黑客渗透，简单了解，黑客首先攻击了一台web服务器，破解了后台的账户密码，随之利用破解的账号密码登陆了mail系统，然后获取了vpn的申请方式，然后登陆了vpn，在内网pwn掉了一台打印机，请根据提供的流量包回答下面有关问题

二、关卡列表

1 某公司内网网络被黑客渗透，请分析流量，给出黑客使用的扫描器

2 某公司内网网络被黑客渗透，请分析流量，得到黑客扫描到的登陆后台是(相对路径即可)

3 某公司内网网络被黑客渗透，请分析流量，得到黑客使用了什么账号密码登陆了web后台(形式:username/password)

4 某公司内网网络被黑客渗透，请分析流量，得到黑客上传的webshell文件名是，内容是什么,提交webshell内容的base编码

5 某公司内网网络被黑客渗透，请分析流量，黑客在robots.txt中找到的flag是什么

6 某公司内网网络被黑客渗透，请分析流量，黑客找到的数据库密码是多少

7 某公司内网网络被黑客渗透，请分析流量，黑客在数据库中找到的hash_code是什么

8 某公司内网网络被黑客渗透，请分析流量，黑客破解了账号ijnu@test.com得到的密码是什么

9 某公司内网网络被黑客渗透，请分析流量，被黑客攻击的web服务器，网卡配置是是什么，提交网卡内网ip

10 某公司内网网络被黑客渗透，请分析流量，黑客使用了什么账号登陆了mail系统（形式: username/password）

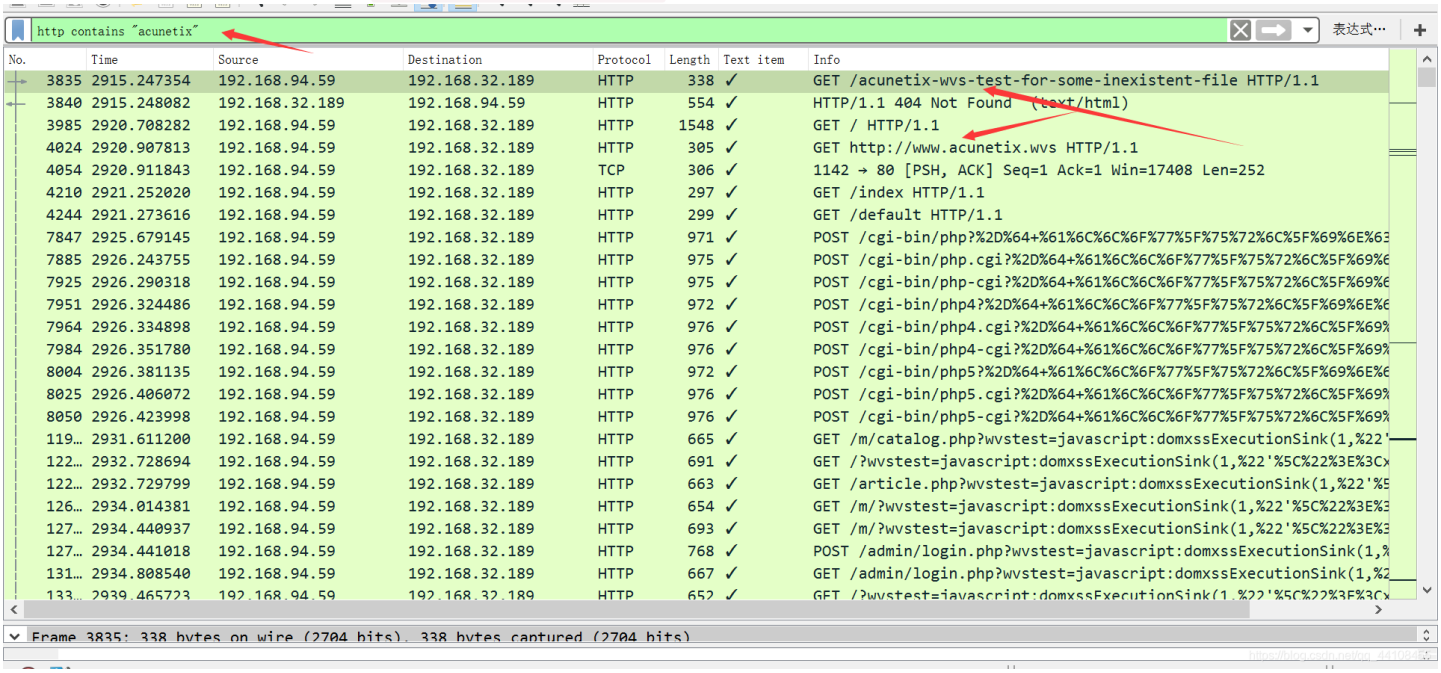
11某公司内网网络被黑客渗透，请分析流量，黑客获得的vpn的ip是多少

三、解题过程

1.黑客使用的扫描器

打开webone.pcap数据包，按照协议类型排序一下，看到http协议的时候，发现了明显的awvs的特征

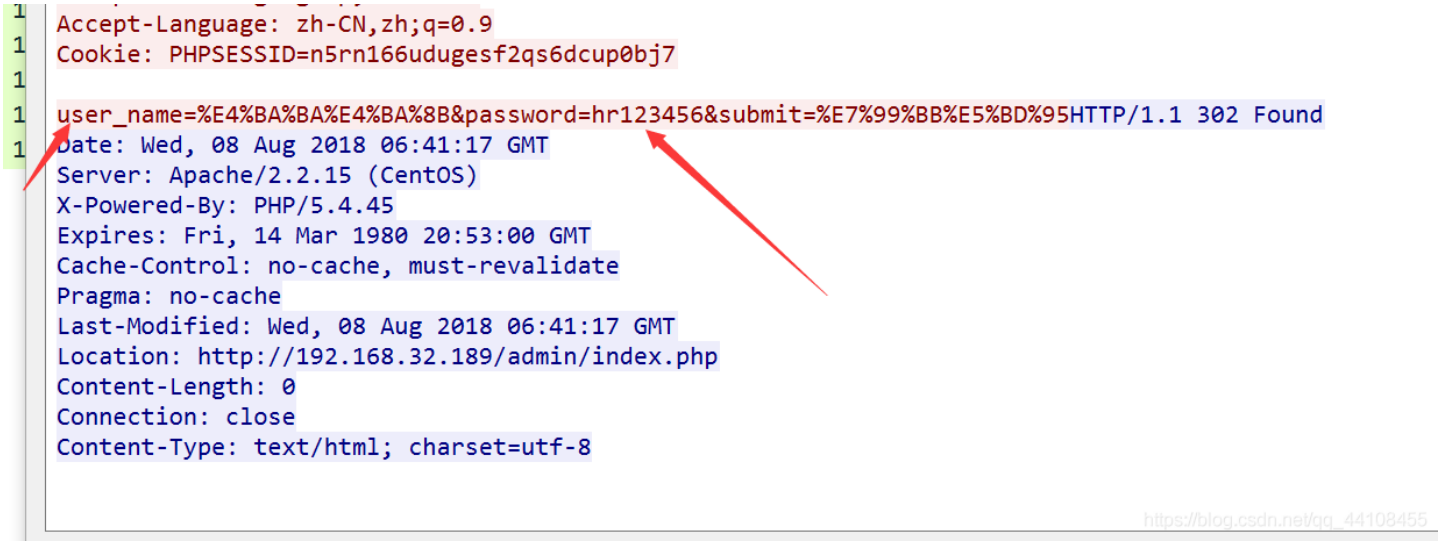
打开webone.pcap数据包，使用 `http contains acunetix` 发现了很多awvs的特征，说明是用awvs进行扫描的



2、黑客扫描到的登陆后台

登陆后台99%使用的是POST方法，直接使用过滤器过滤一下，然后追踪TCP流，看到302重定向，基本就是登陆成功了

```
http.request.method=="POST"
```

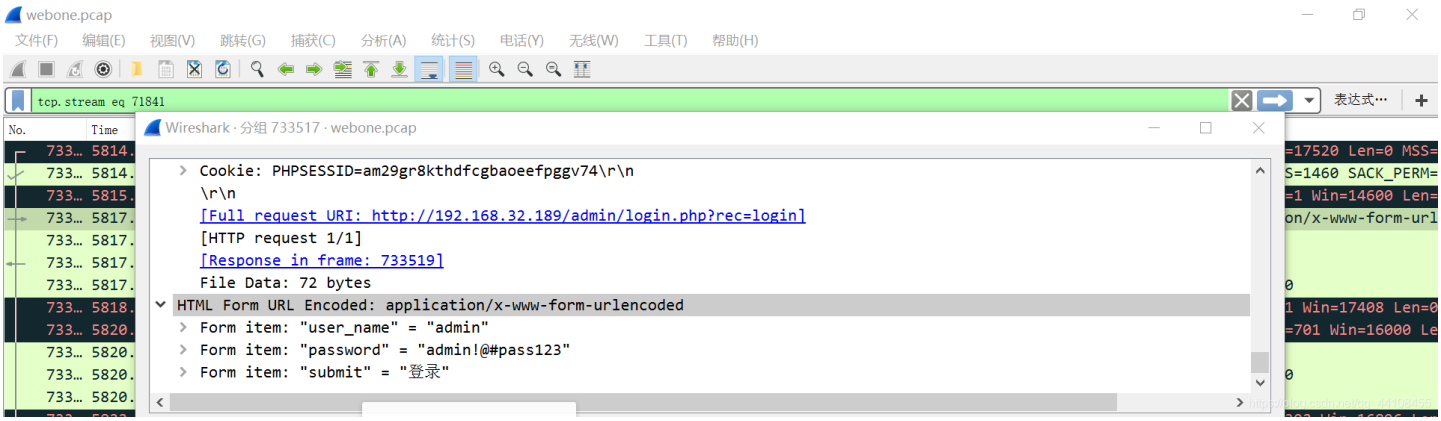


根据上一步我们看到的后台登陆302重定向结果可以判断已经登陆成功了，直接查看数据即可获得账号密码

3 黑客登录使用的账号密码

但是查看过后我发现有很多302重定向登陆成功的结果，发现了很多账号密码，为了确定黑客所使用的，我找了一下黑客的ip地址，就是刚刚看到使用awvs进行扫描的源地址一定就是黑客的ip。然后使用过滤器再次过滤一下。

```
http.request.method=="POST" and ip.src==192.168.94.59 and http contains "rec=login"
```

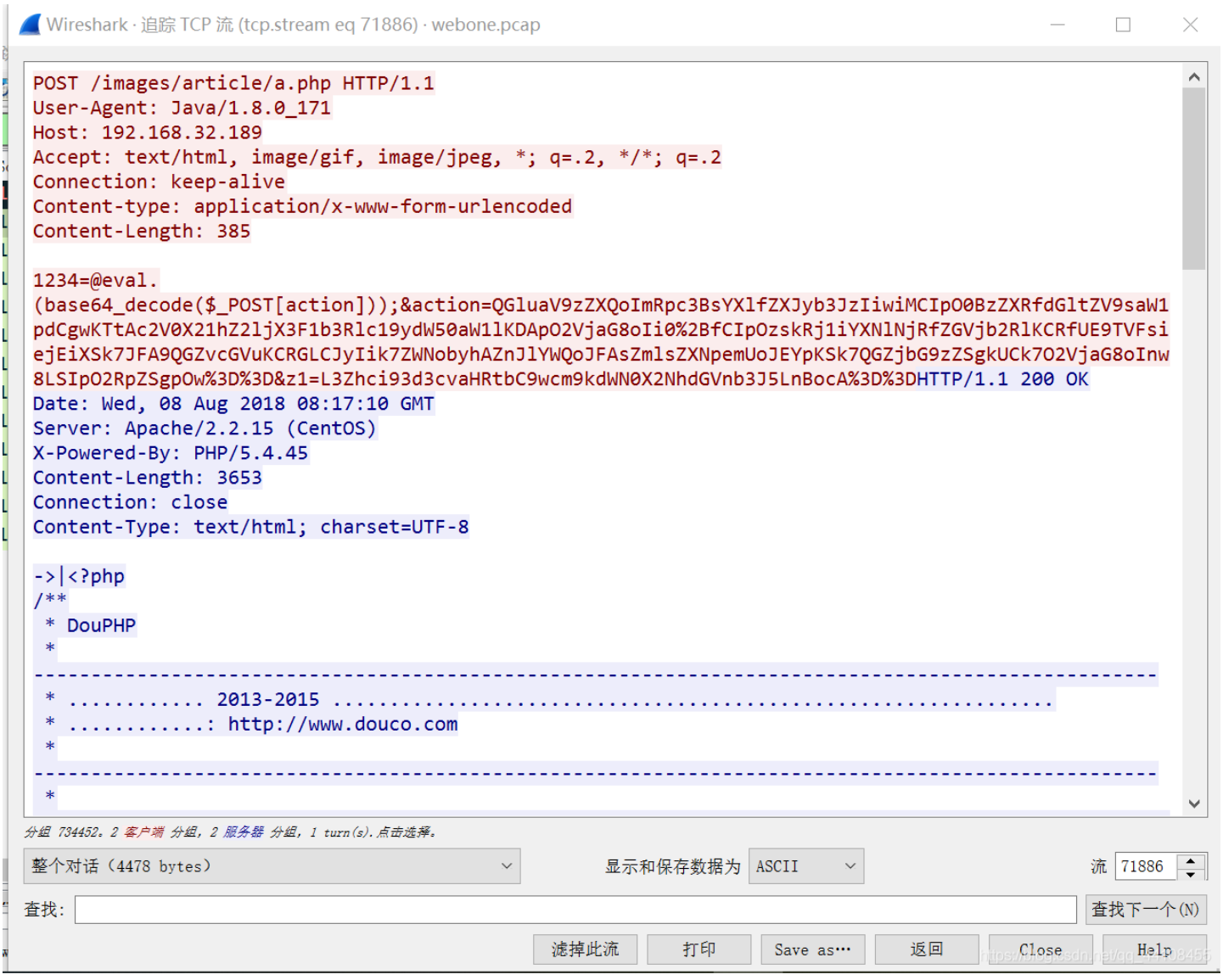


于是乎，找到了，找到了黑客登陆的账号密码！

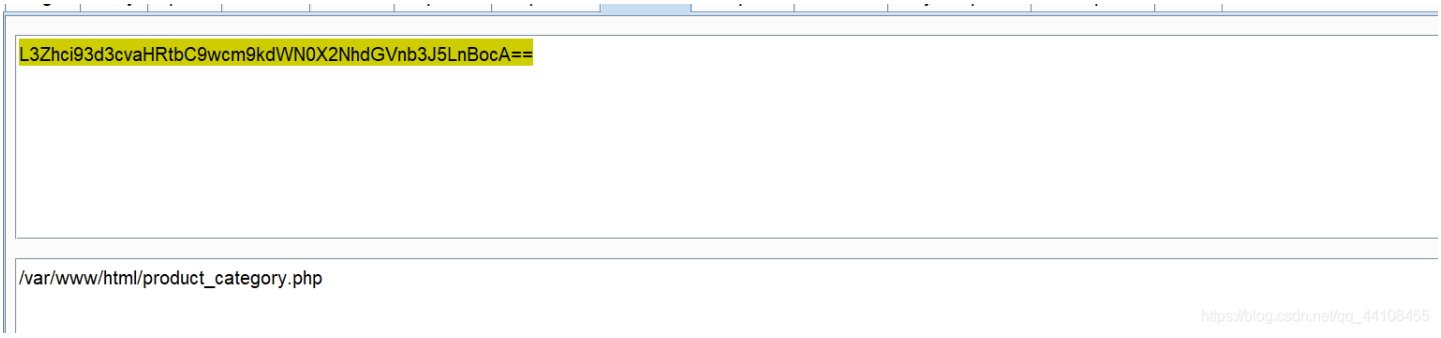
4 webshell文件名和内容

通过下面的语句过滤一下数据，翻阅数据包后发现了一个a.php可能有点蹊跷，但是没有发现他是如何上传的，不过追踪一下TCP流，发现1234为传递值，并有base64加密过的内容，解密一下发现是php代码，以z1为传递值，z1也是使用了base64加密过的内容，再次解密一下得到了一个目录。总结上面的东西发现好像并没有什么作用。。。

```
http.request.method=="POST" and ip.src==192.168.94.59 and http
```

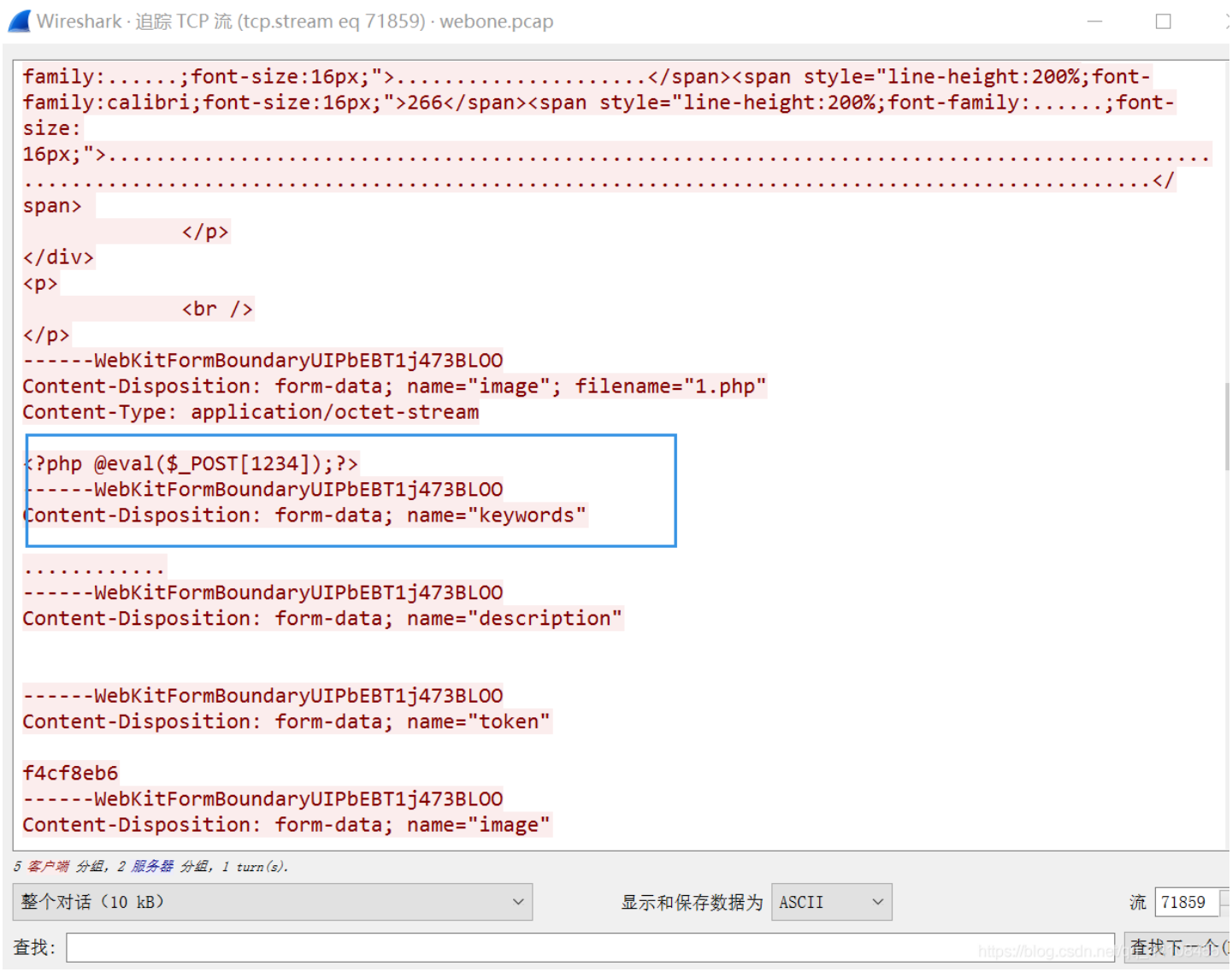


```
@ini_set("display_errors","0");
@set_time_limit(0);
@set_magic_quotes_runtime(0);
echo("->|");;
$F=base64_decode($_POST["z1"]);
$P=@fopen($F,"r");
echo(@fread($P,filesize($F)));
@fclose($P);;
echo("|<-");
die();
```



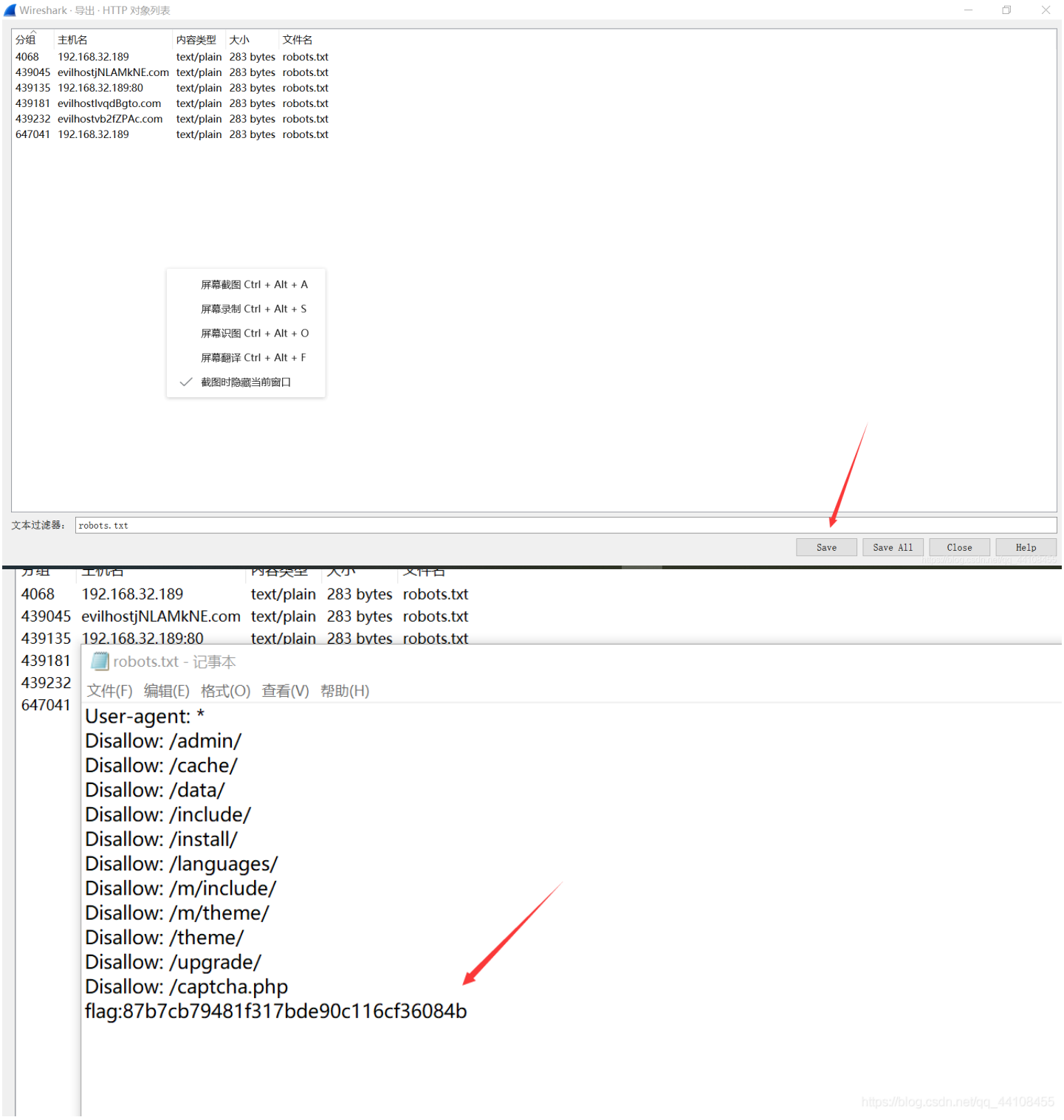
从上面的发现基本可以断定webshell是php写的，盲猜一下是php一句话木马，使用下面的语句过滤一下，没有发现数据，考虑到可能是tcp重传的原因，导致http中没追踪到，把http换成tcp再次过滤一下查看，最终找到了webshell的内容

```
http contains "<?php @eval"
tcp contains "<?php @eval"
```



5 robots.txt中的flag

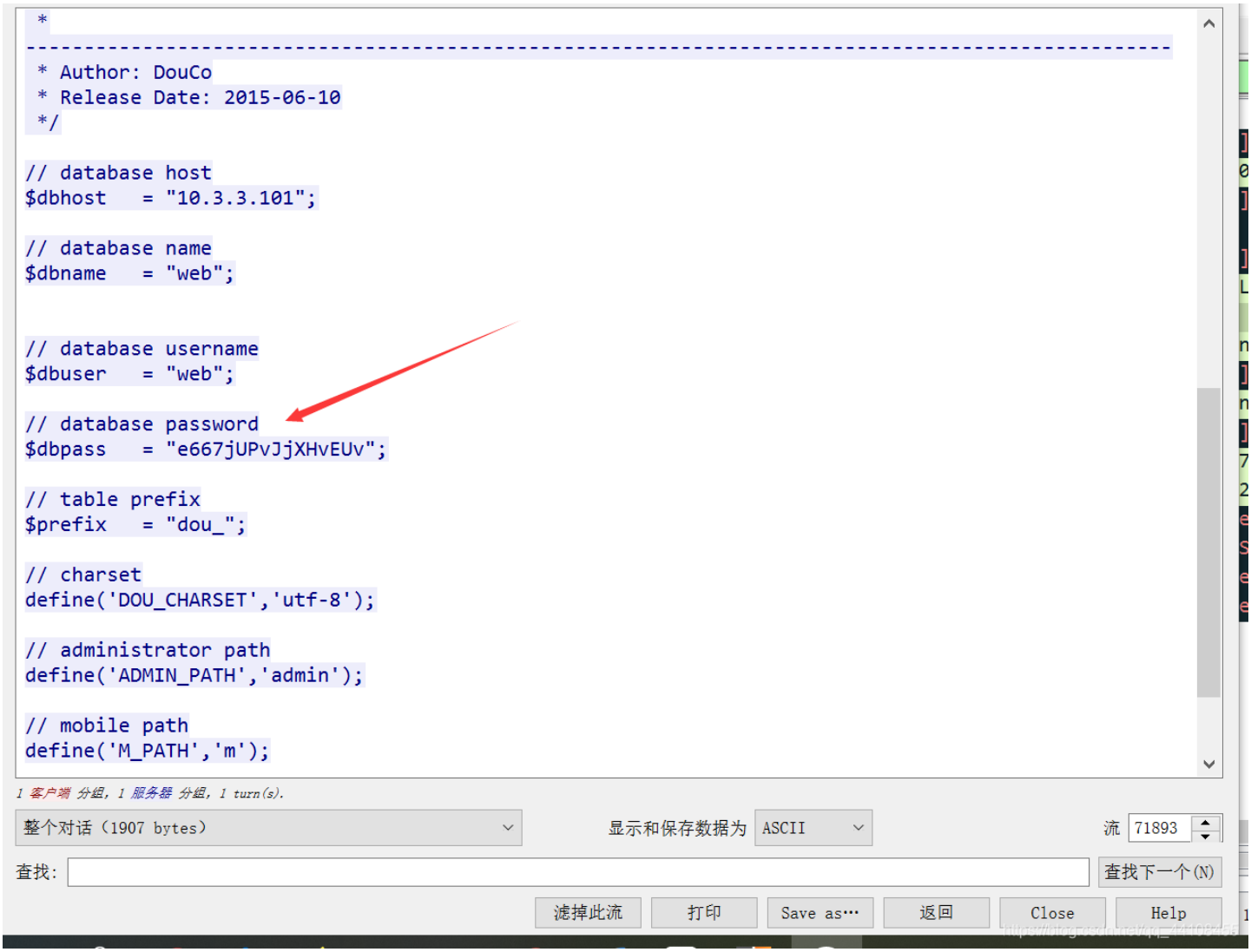
直接导出http对象，在文本过滤器中选择robots.txt，将文件保存下来，即可获得flag



6 数据库密码

直接过滤http数据包，查看数据包的末尾，如果数据库登陆成功，那么http响应码应该为200,并且一般会包含database，逐一查看响应码为200的数据包，即可找到数据库密码

```
http.response.code==200 and http contains "database"
```



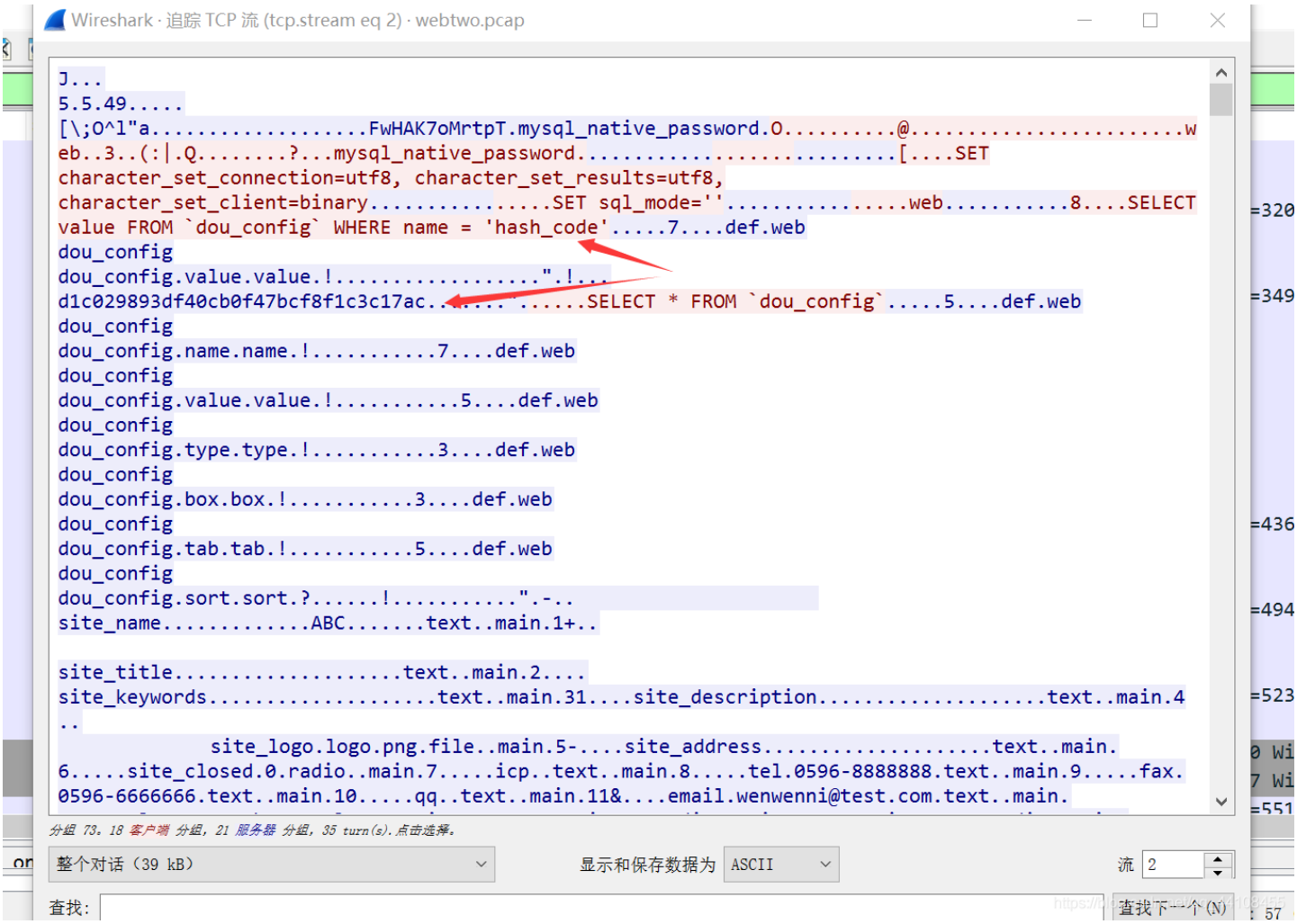
7 hash_code

打开webtwo流量包，

可以先利用这个关键字查找一下，但是没有发现什么，既然还是关于数据库的，在上面我们已经知道数据库的主机是10.3.3.101，可以先查这个ip有什么数据。

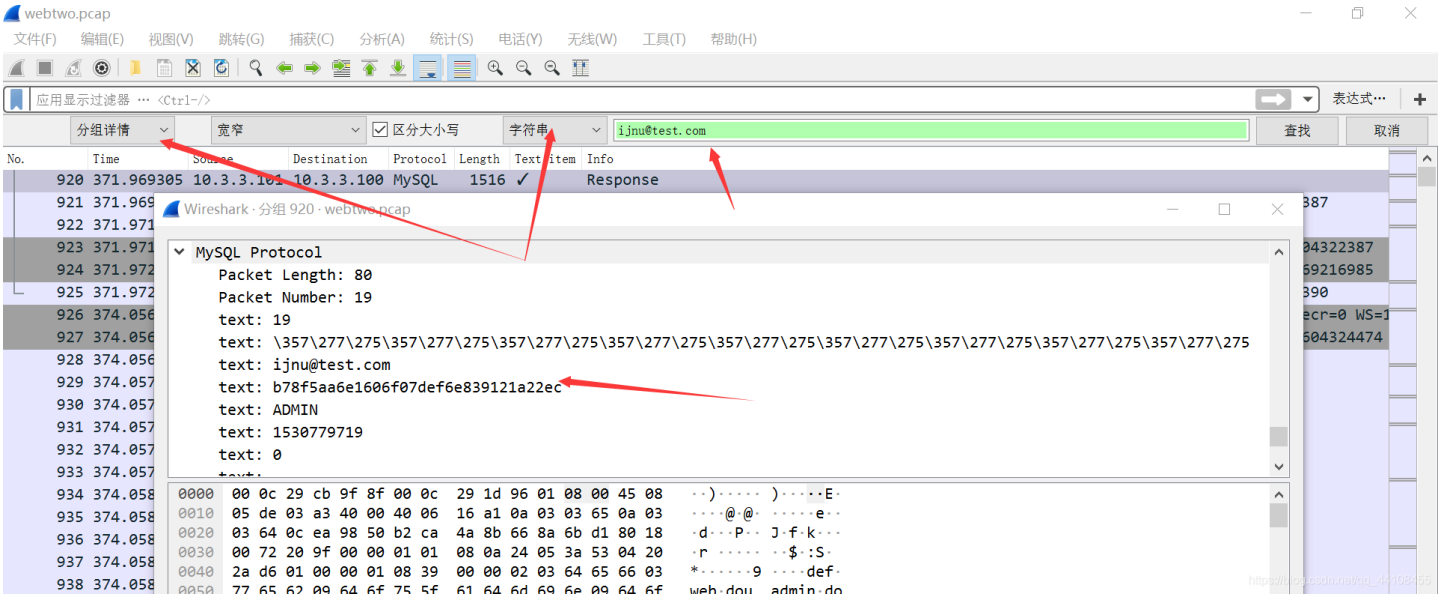
```
ip.src==10.3.3.101 and tcp contains "hash_code"
```

追踪tcp流，



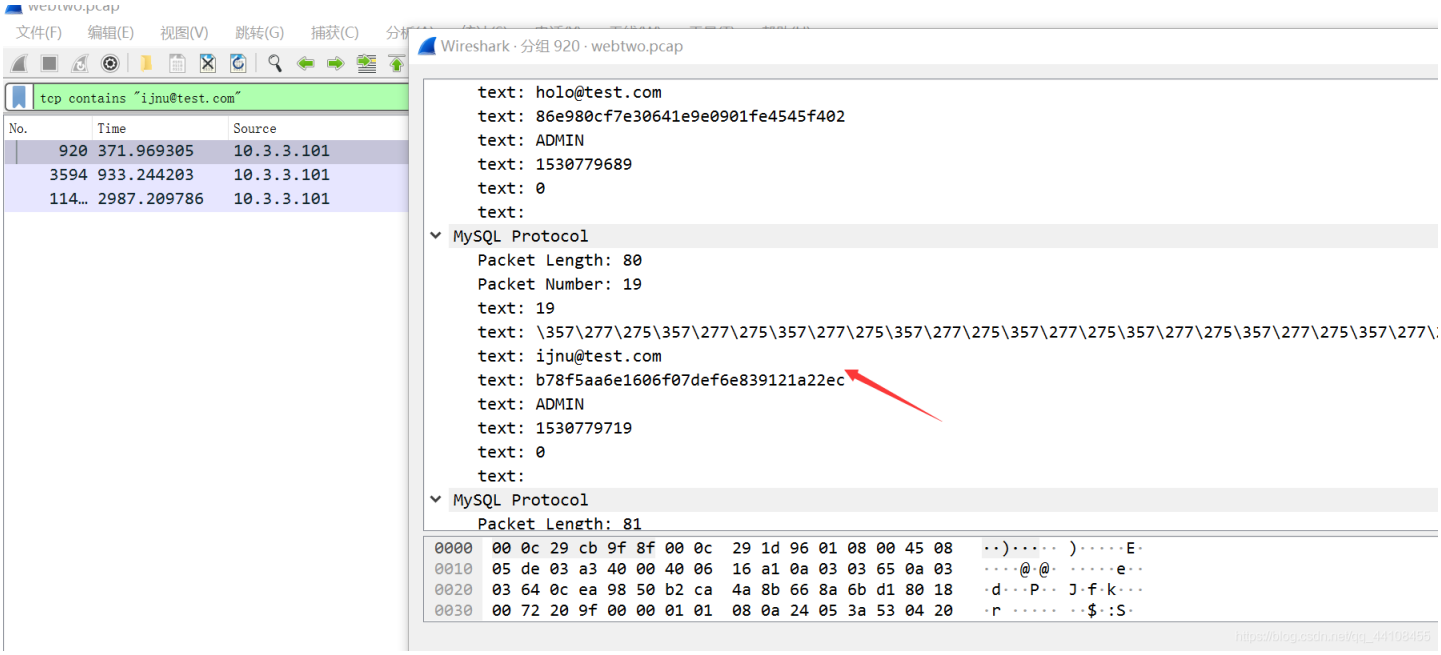
8 黑客破解了账号ijnu@test.com得到的密码是什么

在webtwo.pcap这个流量包中，使用分组详情查询，即可查到密码

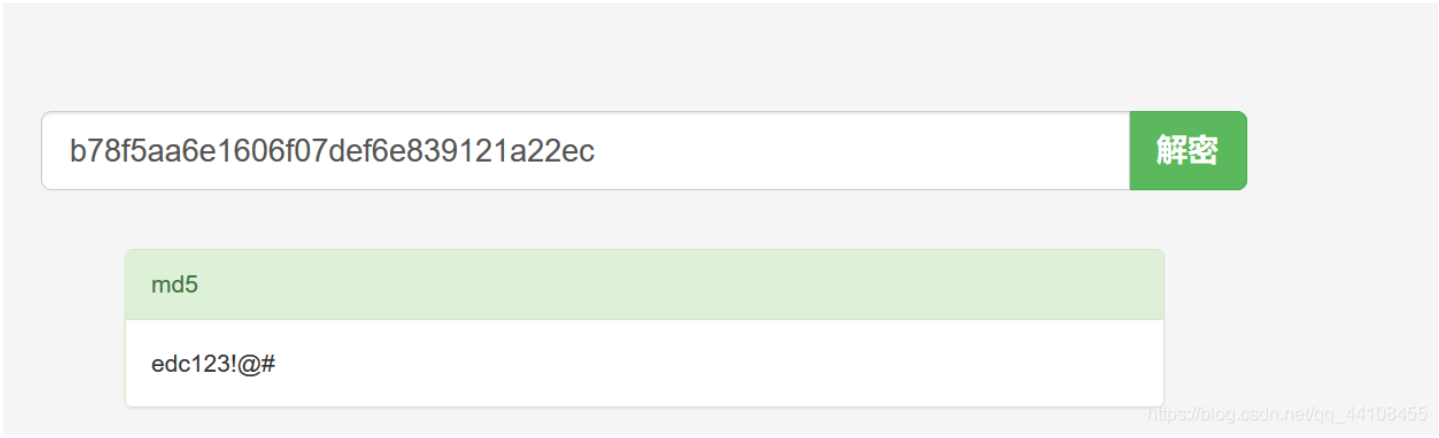


也可以直接用下面的语法进行过滤：

```
tcp contains "ijnu@test.com"
```



md5解密：

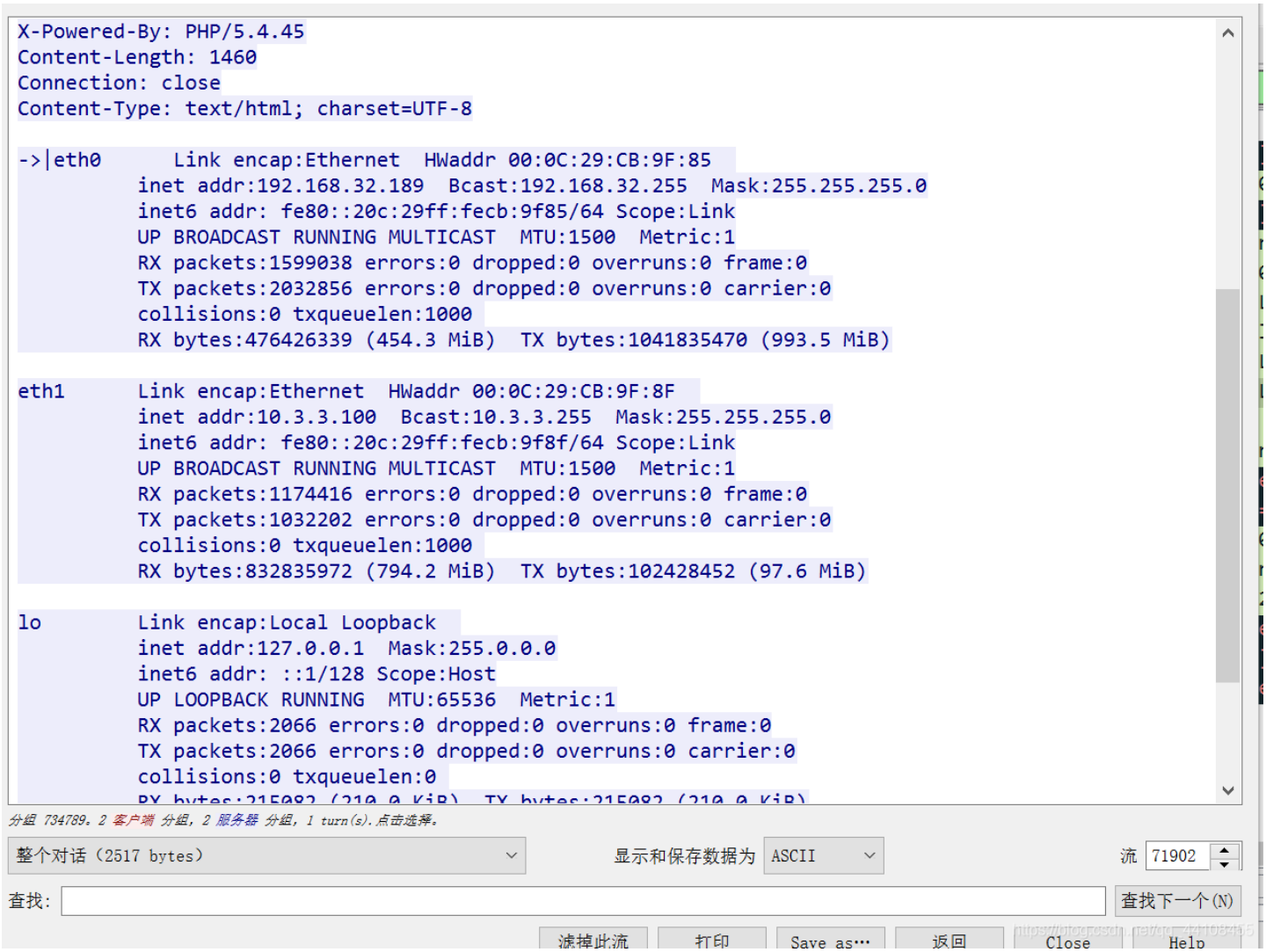


9 被黑客攻击的web服务器，网卡配置是什么，提交网卡内网ip

回到webone.pcap这个流量包中，这个问题问的是网卡的配置，一般网卡的名称都为eth0,所以可以利用这个关键词进行查询

```
tcp contains "eth0"
```

追踪一下tcp流，即可发现网卡的相关配置



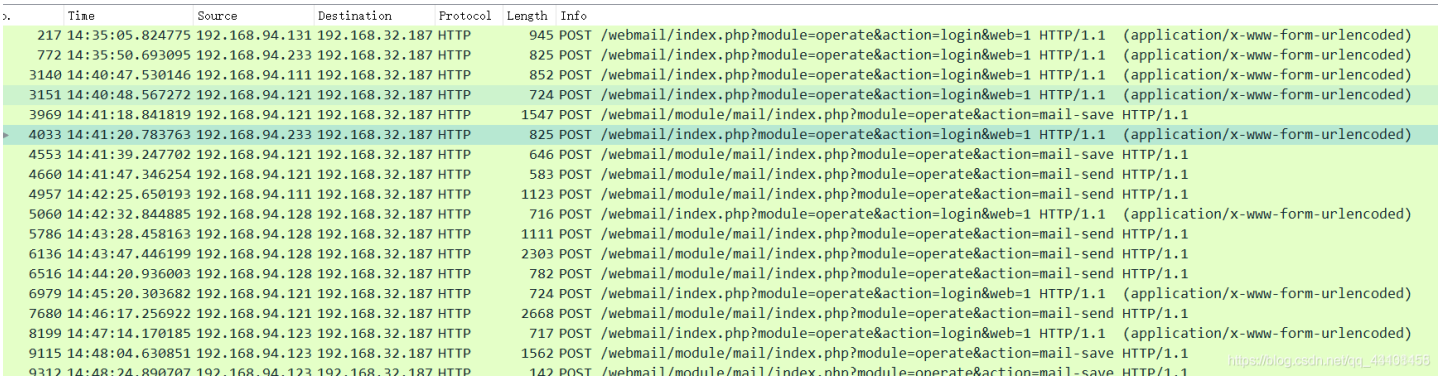
内网IP为10.3.3.100

10 黑客使用了什么账号登陆了mail系统

这题需要综合来看mailtwo.pcap和mailtwo1.pcap两个数据包。

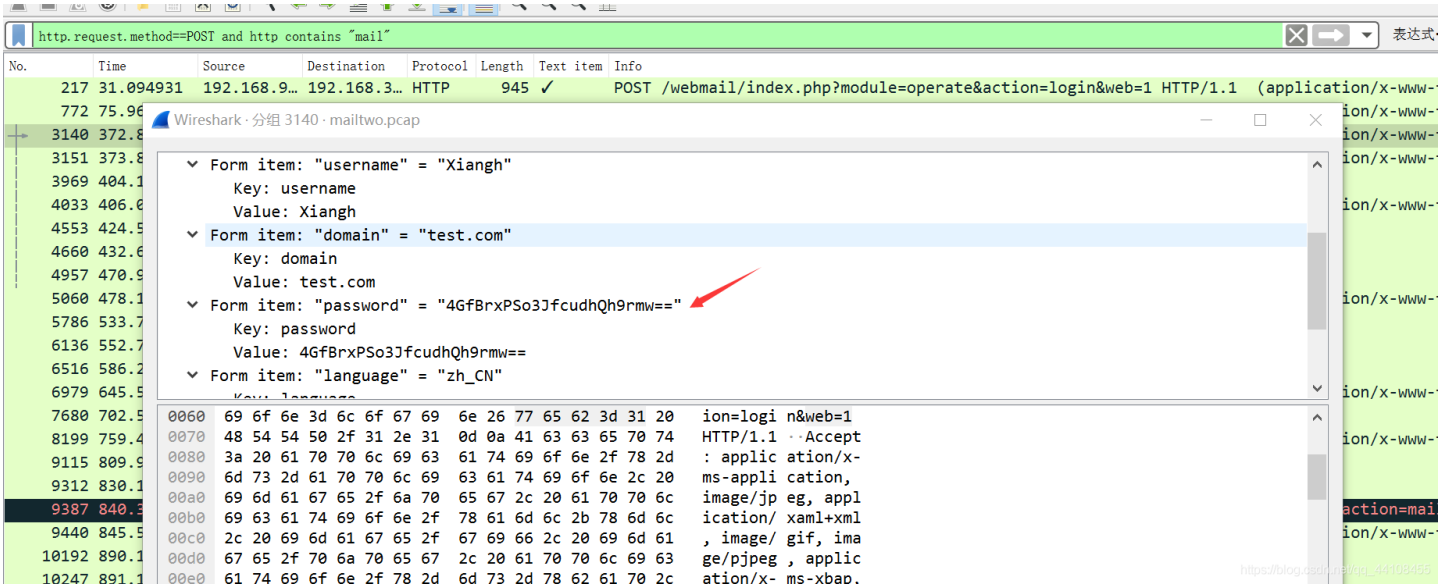
先查询下mailtwo.pcap这个数据包，一开始利用POST和mail过滤了下

```
http.request.method==POST && http contains "mail"
```



No.	Time	Source	Destination	Protocol	Length	Info
217	14:35:05.824775	192.168.94.131	192.168.32.187	HTTP	945	POST /webmail/index.php?module=operate&action=login&web=1 HTTP/1.1 (application/x-www-form-urlencoded)
772	14:35:50.693095	192.168.94.233	192.168.32.187	HTTP	825	POST /webmail/index.php?module=operate&action=login&web=1 HTTP/1.1 (application/x-www-form-urlencoded)
3140	14:40:47.530146	192.168.94.111	192.168.32.187	HTTP	852	POST /webmail/index.php?module=operate&action=login&web=1 HTTP/1.1 (application/x-www-form-urlencoded)
3151	14:40:48.567272	192.168.94.121	192.168.32.187	HTTP	724	POST /webmail/index.php?module=operate&action=login&web=1 HTTP/1.1 (application/x-www-form-urlencoded)
3969	14:41:18.841819	192.168.94.121	192.168.32.187	HTTP	1547	POST /webmail/module/mail/index.php?module=operate&action=mail-save HTTP/1.1
4033	14:41:20.783763	192.168.94.233	192.168.32.187	HTTP	825	POST /webmail/index.php?module=operate&action=login&web=1 HTTP/1.1 (application/x-www-form-urlencoded)
4553	14:41:39.247702	192.168.94.121	192.168.32.187	HTTP	646	POST /webmail/module/mail/index.php?module=operate&action=mail-save HTTP/1.1
4660	14:41:47.346254	192.168.94.121	192.168.32.187	HTTP	583	POST /webmail/module/mail/index.php?module=operate&action=mail-send HTTP/1.1
4957	14:42:25.650193	192.168.94.111	192.168.32.187	HTTP	1123	POST /webmail/module/mail/index.php?module=operate&action=mail-send HTTP/1.1
5060	14:42:32.844885	192.168.94.128	192.168.32.187	HTTP	716	POST /webmail/index.php?module=operate&action=login&web=1 HTTP/1.1 (application/x-www-form-urlencoded)
5786	14:43:28.458163	192.168.94.128	192.168.32.187	HTTP	1111	POST /webmail/module/mail/index.php?module=operate&action=mail-send HTTP/1.1
6136	14:43:47.446199	192.168.94.128	192.168.32.187	HTTP	2303	POST /webmail/module/mail/index.php?module=operate&action=mail-send HTTP/1.1
6516	14:44:20.936003	192.168.94.128	192.168.32.187	HTTP	782	POST /webmail/module/mail/index.php?module=operate&action=mail-send HTTP/1.1
6979	14:45:20.303682	192.168.94.121	192.168.32.187	HTTP	724	POST /webmail/index.php?module=operate&action=login&web=1 HTTP/1.1 (application/x-www-form-urlencoded)
7680	14:46:17.256922	192.168.94.121	192.168.32.187	HTTP	2668	POST /webmail/module/mail/index.php?module=operate&action=mail-send HTTP/1.1
8199	14:47:14.170185	192.168.94.123	192.168.32.187	HTTP	717	POST /webmail/index.php?module=operate&action=login&web=1 HTTP/1.1 (application/x-www-form-urlencoded)
9115	14:48:04.630851	192.168.94.123	192.168.32.187	HTTP	1562	POST /webmail/module/mail/index.php?module=operate&action=mail-save HTTP/1.1
9312	14:48:24.890707	192.168.94.123	192.168.32.187	HTTP	142	POST /webmail/module/mail/index.php?module=operate&action=mail-save HTTP/1.1

发现黑客进行大量的登陆尝试，随便找了一个密码，先看看是什么加密的



不是base64,应该是AES加密，但需要找到加密的密钥，所以还是得重新过滤在服务器返回的信息中去查找，就先只过滤一下http,随便找一个状态码为200的追踪下TCP流，在服务器返回的信息中发现

```
    var loginCheck = function(form) {
        if(form.username.value == "") {
            alert(".........................");
            form.username.focus();
            return false;
        }
        if(form.password.value == "") {
            alert(".........................");
            form.password.focus();
            return false;
        }
                else{
                        var key_hash = CryptoJS.MD5('1234567812345678');
                        var key = CryptoJS.enc.Utf8.parse(key_hash);
                        var iv  = CryptoJS.enc.Utf8.parse('1234567812345678');
                        form.password.value = CryptoJS.AES.encrypt(form.password.value, key, { iv:
v,mode:CryptoJS.mode.CBC,padding:CryptoJS.pad.ZeroPadding});
                }
```

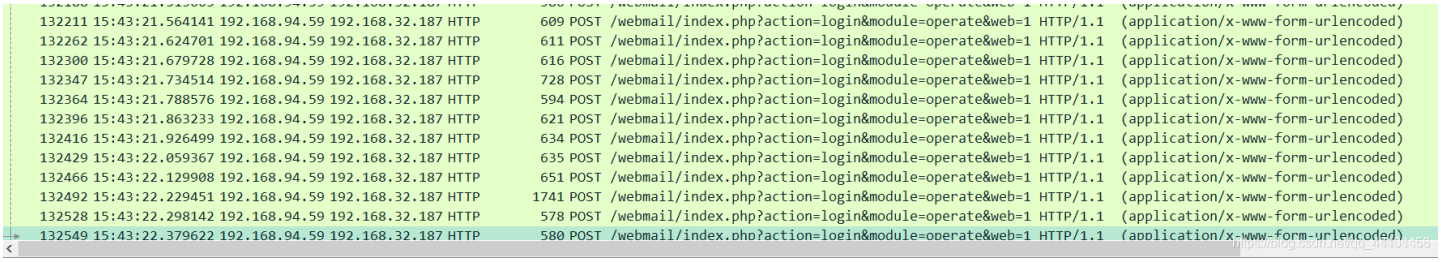

这是AES的CBC加密，填充格式为ZeroPadding，密钥为字符串1234567812345678的hash值，偏移量为1234567812345678

既然加密方式知道了，下面只需要找到正确的账号密码即可

在过滤了http后，发现第一条数据有logout,查看了一下Cookie信息，发现了登陆的用户名

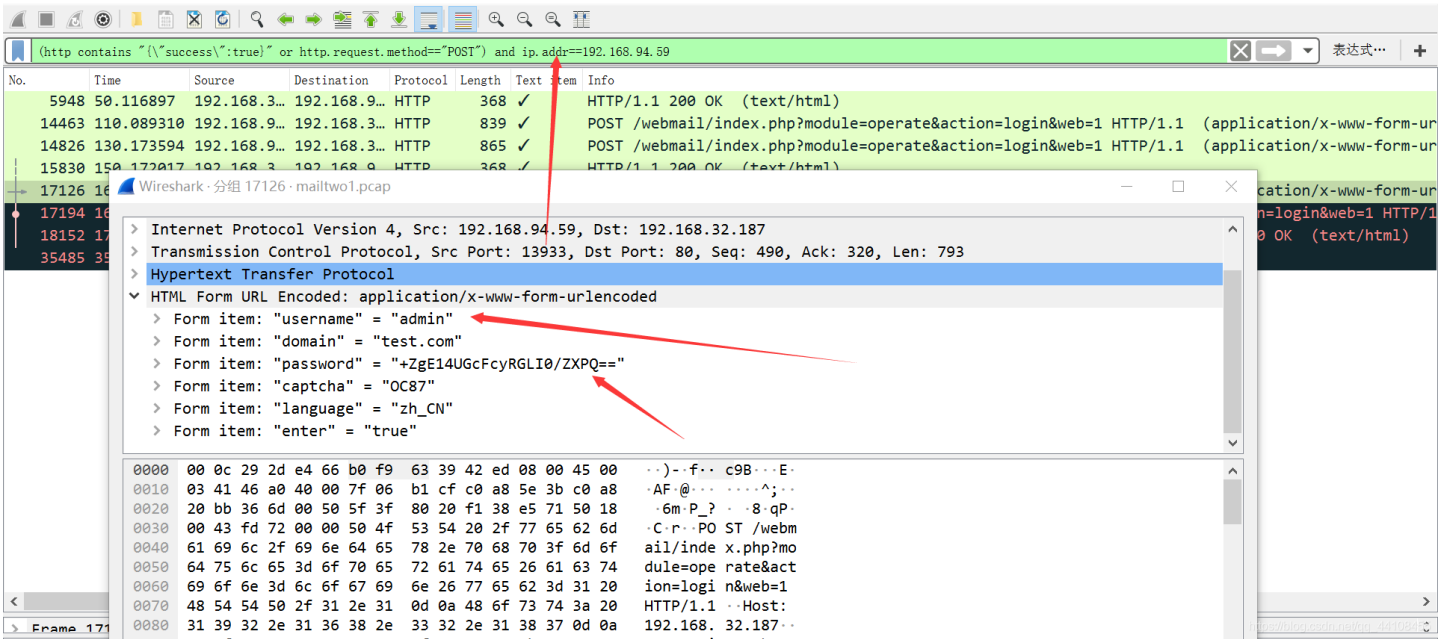
在42号数据请求中，发现登录用户为wenwenni，再查看一下返回数据44号中出现{“success”:true}，代表登陆成功。

```
(http contains "{\"success\":true}" or http.request.method=="POST") and ip.addr==192.168.94.59
```



发现都是在爆破，而且最后也没有出现成功的，利用这个过滤方法查询第二个包mailtwo1.pcap

从后往前看，18152是登陆成功的返回结果，对应的17126则应该就是正确的加密后的密码



明文:

1234567812345678

散列/哈希算法:

SHA1 SHA224 SHA256 SHA384 SHA512 MD5

HmacSHA1 HmacSHA224 HmacSHA256 HmacSHA384 HmacSHA512 HmacMD5 PBKDF2

哈希值:

d959caadac9b13dcb3e609440135cf54

aes解密工具：http://tool.chacuo.net/cryptaes

最终账号密码：

```
admin/admin!@#PASS123
```

11黑客获得的vpn的ip是多少

在做题之前需要了解下VPN的一些协议，如PPTP
https://blog.csdn.net/zhaqiwen/article/details/10083025

PPTP原理

```
1.PPTP客户机使用动态分配的TCP端口号，与PPTP服务器使用的保留TCP端口号123建立控制连接
（PPTP控制连接携带PPTP呼叫控制盒管理信息，用于维护PPTP隧道）。

2.客户端与服务器通过控制连接来创建、维护、终止一条隧道。

3.PPP帧的有效载荷经过加密、压缩或是两者的混合处理。

4.使用通用路由封装GRE对PPP帧进行封装。

5.将PPP帧封装进IP数据报文中。通过IP网络如Internet或其他企业准用INTRANET灯发送给PPTP服务器。

6.服务器接收到PPTP数据包后进行常规处理。
```

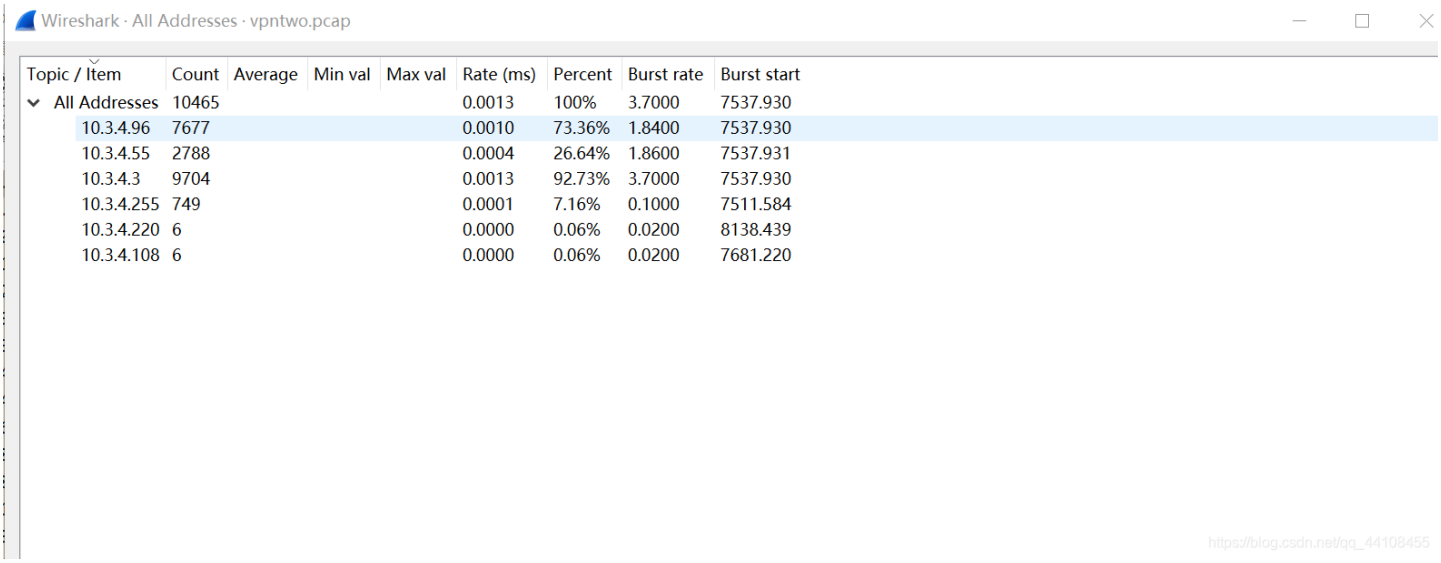
此外，还需要了解下SMB服务
SMB服务详解

先打开vpnone.pcap，发现

26390	16:49:24.171557	192.168.32.131	192.168.94.59	PPP Comp	1447	Compressed data
26391	16:49:24.171579	192.168.32.131	192.168.94.59	PPP Comp	1447	Compressed data
26392	16:49:24.171599	192.168.32.131	192.168.94.59	PPP Comp	1447	Compressed data
26393	16:49:24.171637	192.168.32.131	192.168.94.59	PPP Comp	1447	Compressed data
26394	16:49:24.171661	192.168.32.131	192.168.94.59	PPP Comp	1447	Compressed data
26395	16:49:24.171681	192.168.32.131	192.168.94.59	PPP Comp	1447	Compressed data
26396	16:49:24.171713	192.168.32.131	192.168.94.59	PPP Comp	1447	Compressed data
26397	16:49:24.188550	192.168.94.59	192.168.32.131	GRE	60	Encapsulated PPP
26398	16:49:24.204834	192.168.94.59	192.168.32.131	GRE	60	Encapsulated PPP
26399	16:49:24.237647	192.168.94.59	192.168.32.131	GRE	60	Encapsulated PPP
26400	16:49:24.237671	192.168.94.59	192.168.32.131	GRE	60	Encapsulated PPP
26401	16:49:24.237676	192.168.94.59	192.168.32.131	GRE	60	Encapsulated PPP
26402	16:49:24.238859	192.168.94.59	192.168.32.131	PPP Comp	107	Compressed data
26403	16:49:24.239994	192.168.32.131	192.168.94.59	PPP Comp	1451	Compressed data
26404	16:49:24.240024	192.168.32.131	192.168.94.59	PPP Comp	1447	Compressed data
26405	16:49:24.240070	192.168.32.131	192.168.94.59	PPP Comp	1447	Compressed data
26406	16:49:24.240092	192.168.32.131	192.168.94.59	PPP Comp	1447	Compressed data
26407	16:49:24.240113	192.168.32.131	192.168.94.59	PPP Comp	1447	Compressed data
26408	16:49:24.240154	192.168.32.131	192.168.94.59	PPP Comp	1447	Compressed data

vpnone.pcap应该只是在尝试登陆VPN，再来查看下vpntwo.pcap

在统计——>IPV4中发现



10.3.4.96 、10.3.4.55 、10.3.4.3出现的次数最多

先过滤一下SMB，发现

18711	16:48:59.328716	10.3.4.3	10.3.4.96	DCERPC	175	Request: call_id: 0, Fragment: Mid, opnum: 19, Ctx: 12 [DCE/RPC Mid fragment, reas: #18737]
18712	16:48:59.329040	10.3.4.96	10.3.4.3	SMB	117	Write AndX Response, FID: 0x4006, 42 bytes

所以10.3.4.96是SMB服务器，排除，再来查询下10.3.4.55

```
ip.addr==10.3.4.55
```

16501	16:43:57.233839	10.3.4.3	10.3.4.55	ICMP	162	Echo (ping) request id=0xca34, seq=295/9985, ttl=36 (reply in 16503)
16503	16:43:57.234163	10.3.4.55	10.3.4.3	ICMP	162	Echo (ping) reply id=0xca34, seq=295/9985, ttl=64 (request in 16501)
16506	16:43:57.235401	10.3.4.3	10.3.4.55	ICMP	192	Echo (ping) request id=0xca35, seq=296/10241, ttl=45 (reply in 16507)
16507	16:43:57.235668	10.3.4.55	10.3.4.3	ICMP	192	Echo (ping) reply id=0xca35, seq=296/10241, ttl=64 (request in 16506)

10.3.4.3先PING10.3.4.55

```
故因此可以推断是黑客获得VPN的IP是10.3.4.3
```

参考链接：https://blog.csdn.net/qq_43431158/article/details/107176918