100 Forensics / Wi Will H4CK YOU!!

Description

> Wifi Security Standards have been increased a lot in recent times.
>
> But are they secure enough??? Get the password for our Wifi Network "encryptCTF"
>
> Submit flag as encryptCTF{</password/>} captured.cap
>
> ```
> Author:@mostwanted002
> ```

Files provided

- encryptCTFWPA.cap

Solution

This challenge is similar to It's a WrEP challenge. The only difference is that it is using the WPA protocol instead of the WEP protocol.

Using `aircrack-ng` again and waiting for a long time, give us the password.

```
$ aircrack-ng -a 2 -w rockyou.txt encryptCTFWPA.cap
```

We pass in a wordlist (rockyou.txt in this case) and picking the network to crack and waiting for quite some time, it outputs the password `ThanckYou`.

```
encryptCTF{ThanckYou}
```

恶臭的数据包

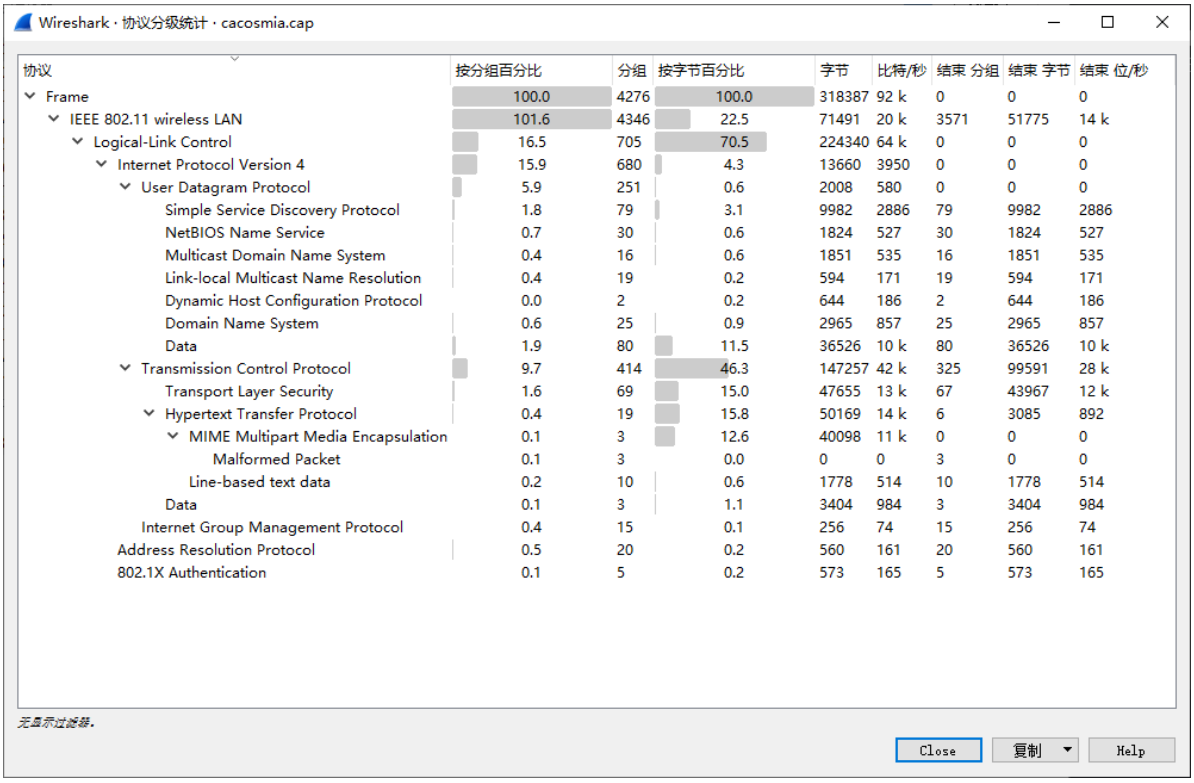
题目

野兽前辈想玩游戏，但是hacker妨碍了他连上无线网，前辈发出了无奈的吼声。

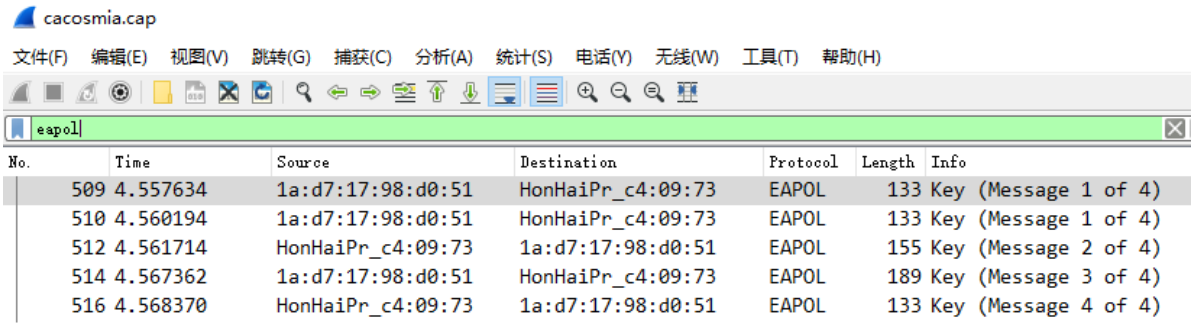
题目存档：恶臭的数据包.7z

解决方案

解压得到一个cap流量包，Wireshark打开看看统计信息：



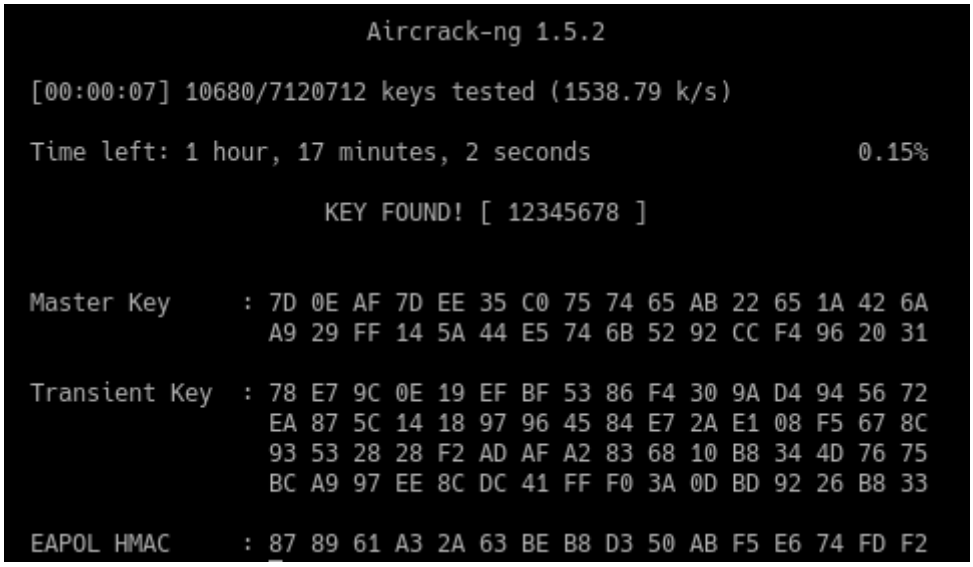
一个无线流量包，信息都被加密了。来看看有没有握手包：



这个时候可以来尝试爆破了。Kali Linux，启动：

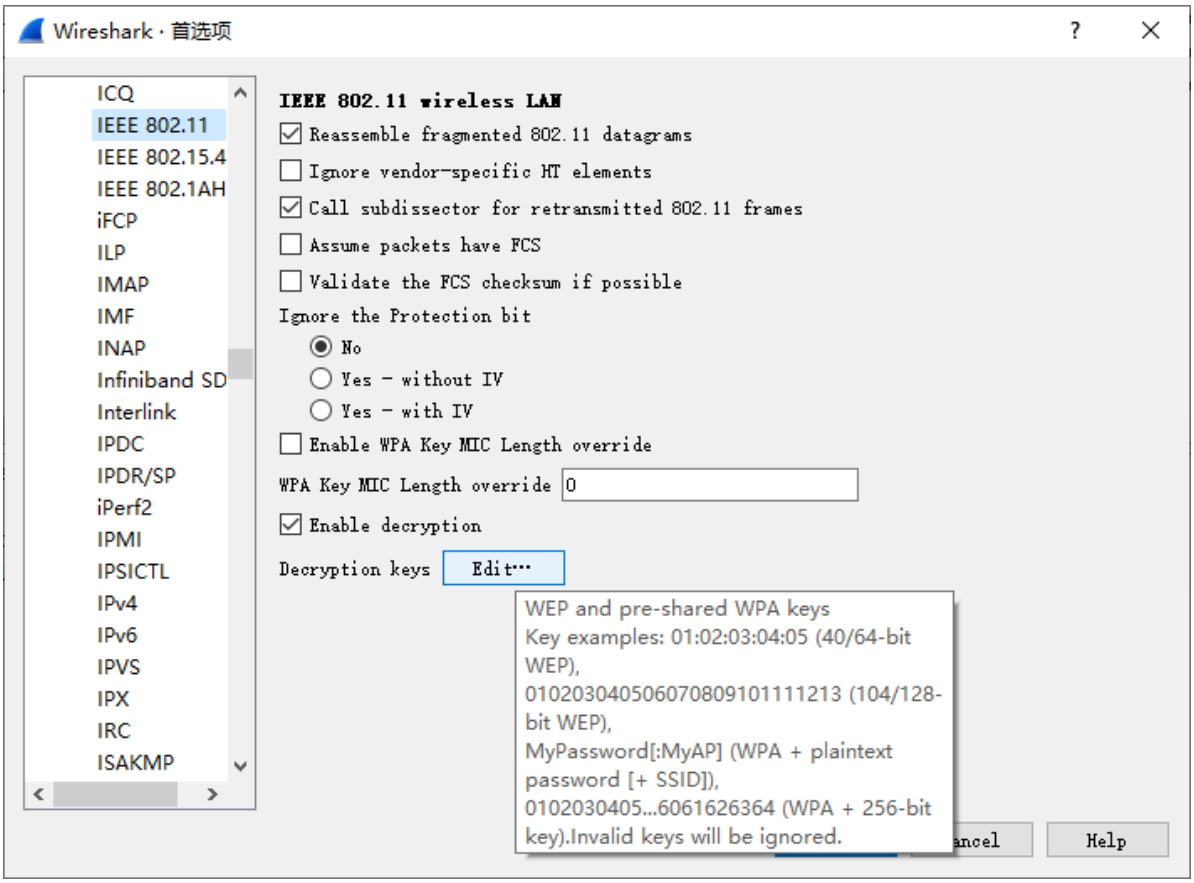
```
aircrack-ng -w ./rockyou.txt ./cacosmia.cap
```

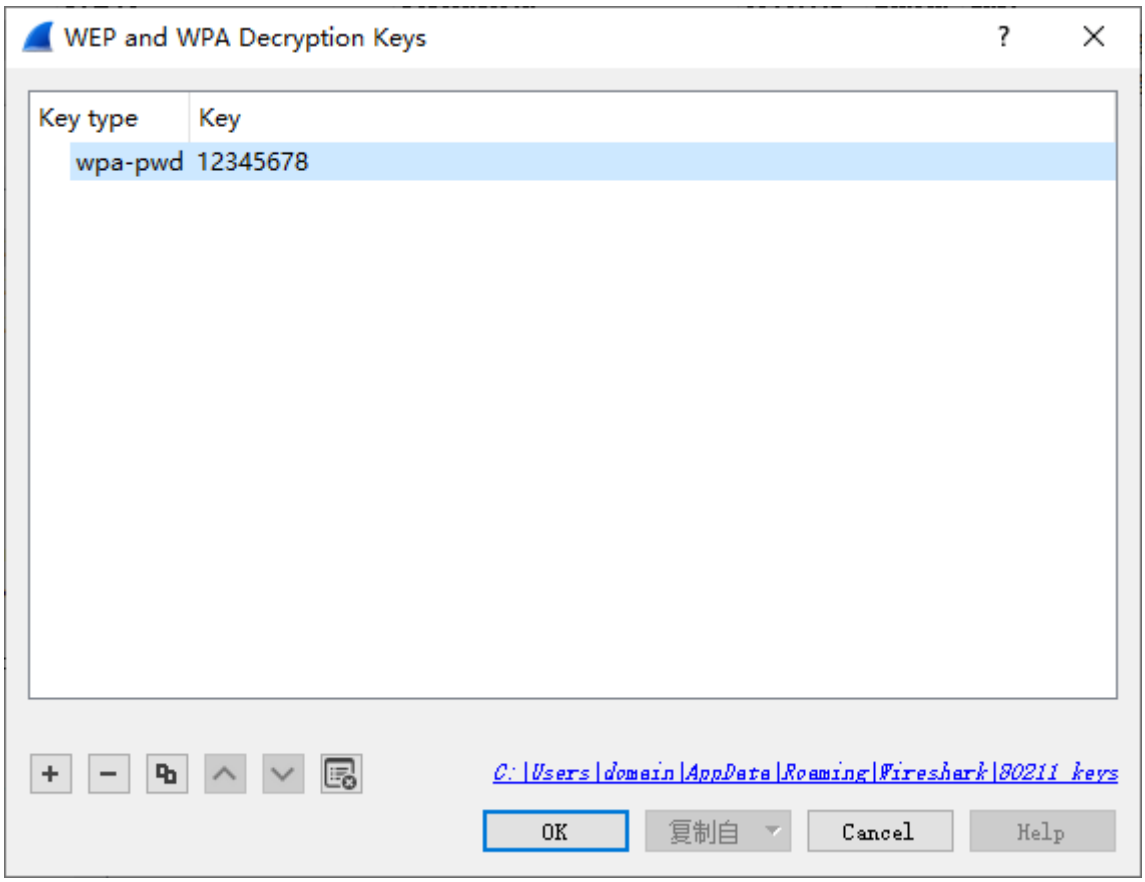
还是那个经典的字典。



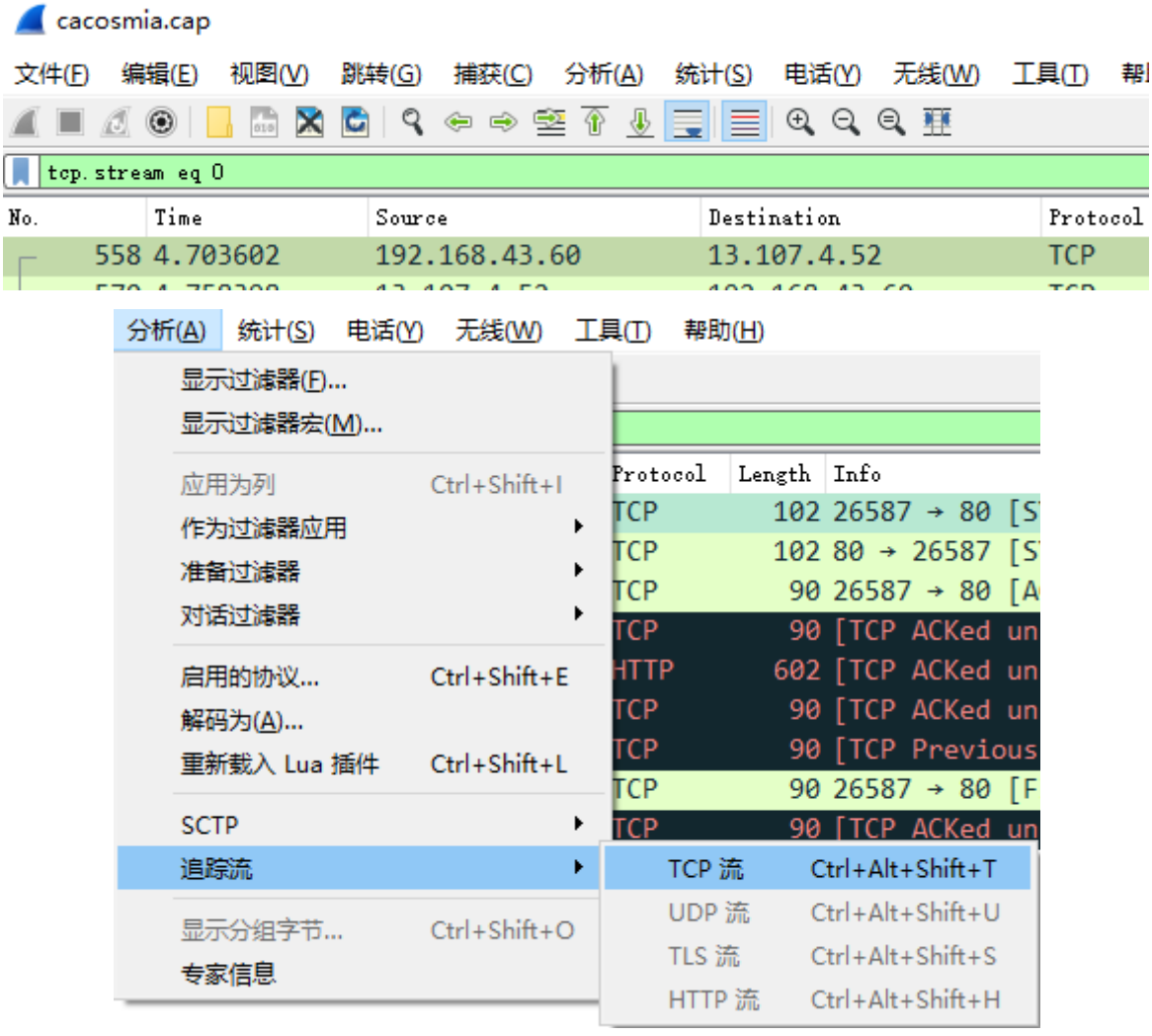
很快就爆出来了，无线密码是`12345678`。这个时候可以来解密了：

Protocols -> IEEE 802.11 -> Decryption keys直达：





保存退出，此时应该能看到已经解密了。开始追踪流，先过滤一下再开始：

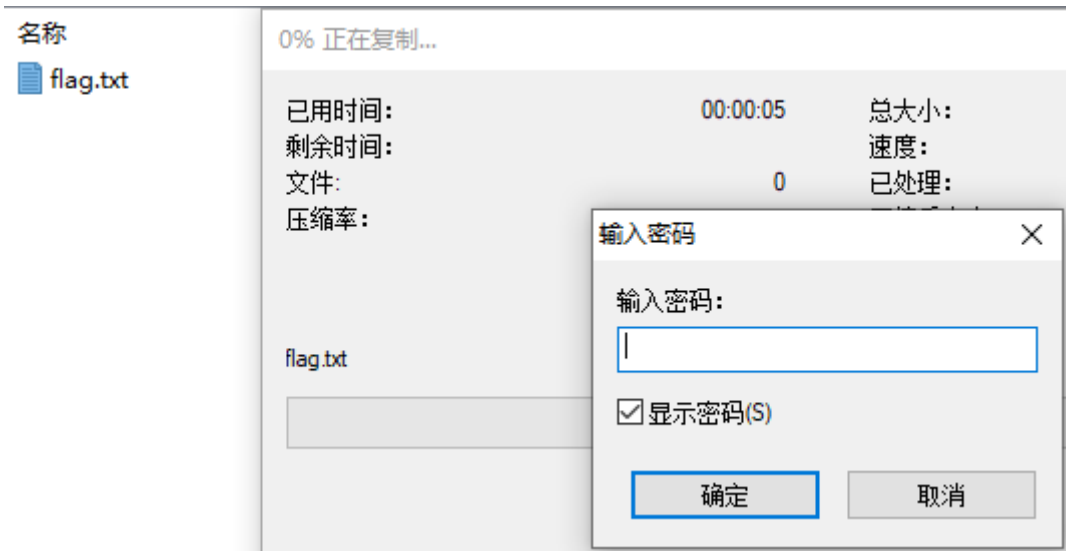


翻到后面就发现有些上传图片的流量。熟悉的PNG文件头：

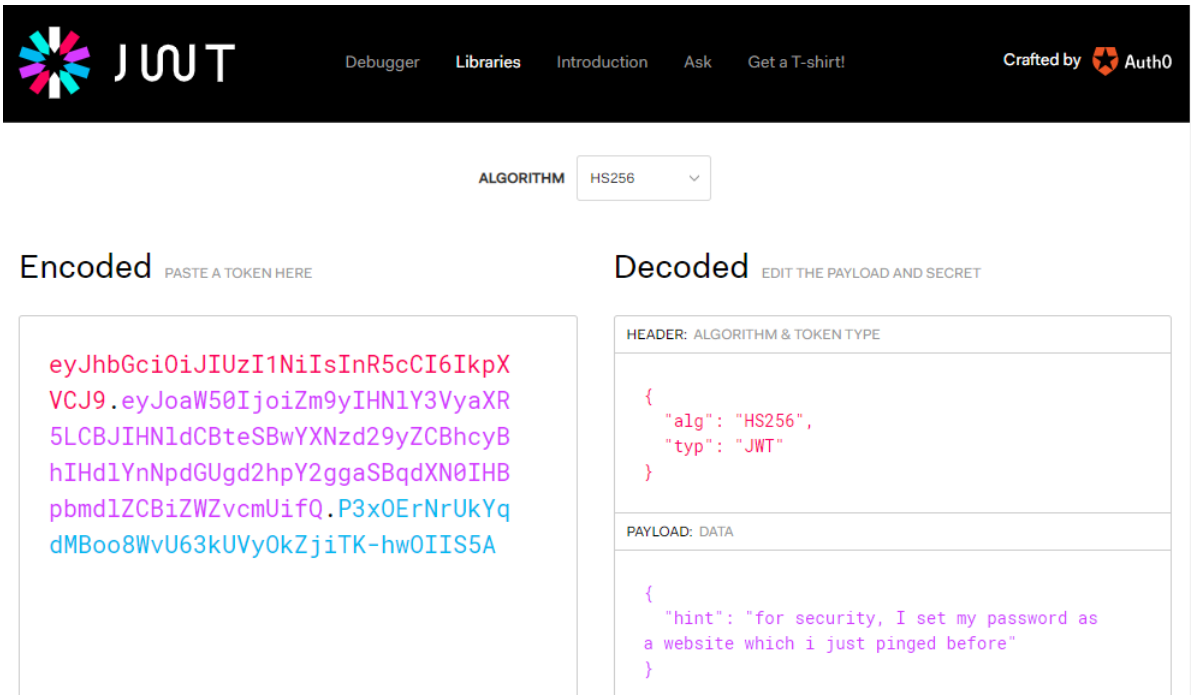
还有让人激动的PK文件头和flag.txt:

先提取出来再说:

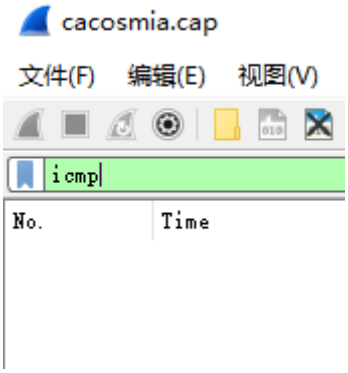
这样导出有HTTP头部信息，用010 Editor去掉即可得到一个图种，可以继续编辑把图和 种 压缩包拆开。也可以用binwalk或者foremost之类的工具分离出来。图用Stegsolve看了看似乎没有什么信息，压缩包被加密了:



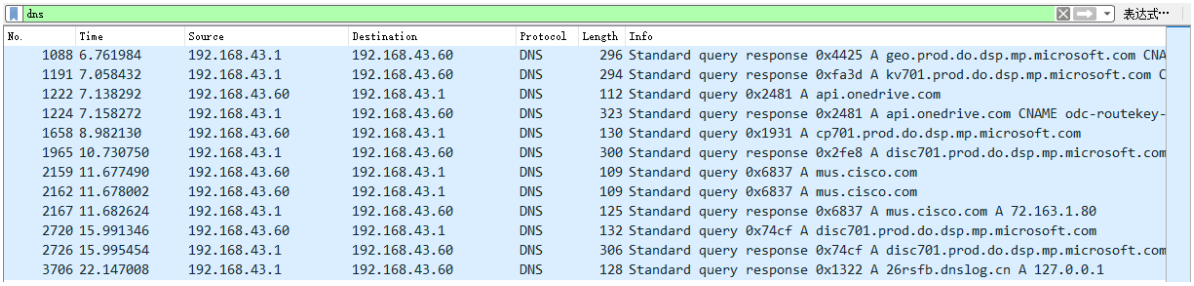
不是个伪加密，回到流量包里找线索。发现Cookie有个JWT，尝试解开：



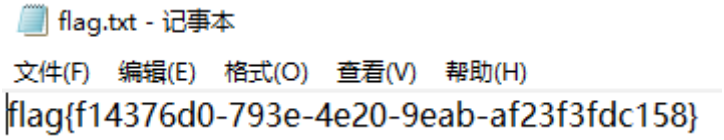
提示密码是被ping过的一个站点。回到流量包中尝试过滤：



没有发现。不过ping一个站点通常会先进行DNS解析：



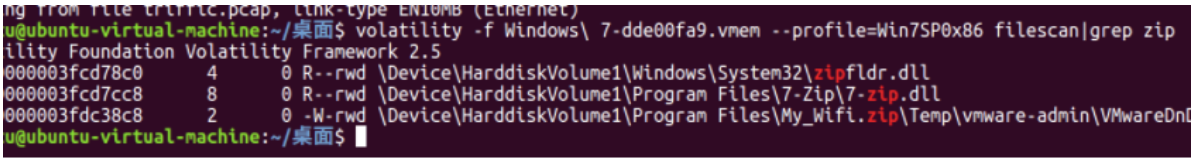
发现最后一次解析有点可疑，返回了环回地址，如果ping环回的话，确实是会抓不到包的。尝试用`26rsfb.dnslog.cn`解压：



`flag{f14376d0-793e-4e20-9eab-af23f3fdc158}`

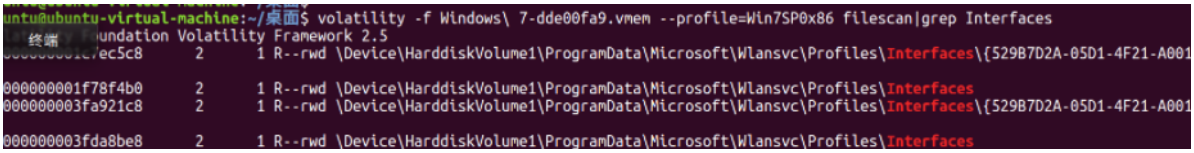
wifi

vol内存取证，直接找zip，找到个奇怪的

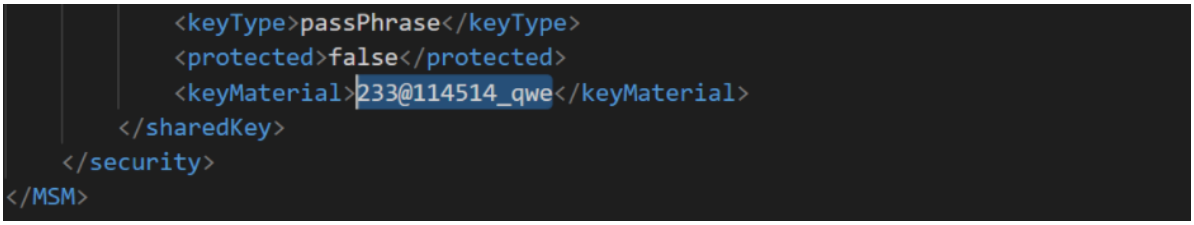


导出，发现需要密码，

老考点了



花括号以内加花括号是密码，解压后获得xml，xml获得密码



对加密的客户端流量进行解密，即可

随后进入流量分析环节。

首先分析服务器流量，直接看http

这是哥斯拉shell的初始化

我们解密得到加密函数和key

```
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';
```

加密函数也是解密函数

分析客户端流量的回显

```
68 74 6d 6c 3b 20 63 68  61 72 73 65 74 3d 55 54   html; ch arset=UT
46 2d 38 0d 0a 0d 0a 37  32 61 39 63 36 39 31 63   F-8····7 2a9c691c
63 64 61 61 62 39 38 66  4c 31 74 4d 47 49 34 59   cdaab98f L1tMGI4Y
54 6c 6a 4d 6e 37 35 65  33 6a 4f 42 53 35 2f 56   TljMn75e 3jOBS5/V
33 31 51 64 31 4e 78 4b  51 4d 43 65 33 68 34 4b   31Qd1NxK QMCe3h4K
77 46 51 66 56 41 45 56  77 6f 72 43 69 30 46 66   wFQfVAEV worCi0Ff
67 42 2b 42 6c 57 5a 68  6a 52 6c 51 75 54 49 49   gB+BlWZh jRlQuTII
42 35 6a 4d 54 55 3d 62  34 63 34 65 31 66 36 64   B5jMTU=b 4c4e1f6d
64 64 32 61 34 38 38                               dd2a488
```

去掉前面的16位和后面的16位 得到

fL1tMGI4YTljMn75e3jOBS5/V31Qd1NxKQMCe3h4KwFQfVAEVworCi0FfgB+BlWZhjRlQuTIIB5jMTU=

哥斯拉输出结果是会将结果压缩然后加密

```
 $result=gzencode($result,6);

 echo base64_encode(encode(@run($data),$key));
```

然后进行解密

```
function encode($D,$K){

    for($i=0;$i<strlen($D);$i++) {

        $c = $K[$i+1&15];

        $D[$i] = $D[$i]^$c;

    }

    return $D;

}

$a=
'fL1tMGI4YTljMn75e3jOBS5/V31Qd1NxKQMCe3h4KwFQfVAEVworCiOFfgB+BlWZhjRlQuTIIB5jMTU
=';

echo gzdecode(encode(base64_decode($a),'3c6e0b8a9c15224a'));
```

得到flag

flag{5db5b7b0bb74babb66e1522f3a6b1b12}