2020 第四届强网杯 线上赛Misc_Writeup

原创

末 初 于 2020-11-07 17:00:12 发布 1110 收藏 4

分类专栏： CTF_MISC_Writeup

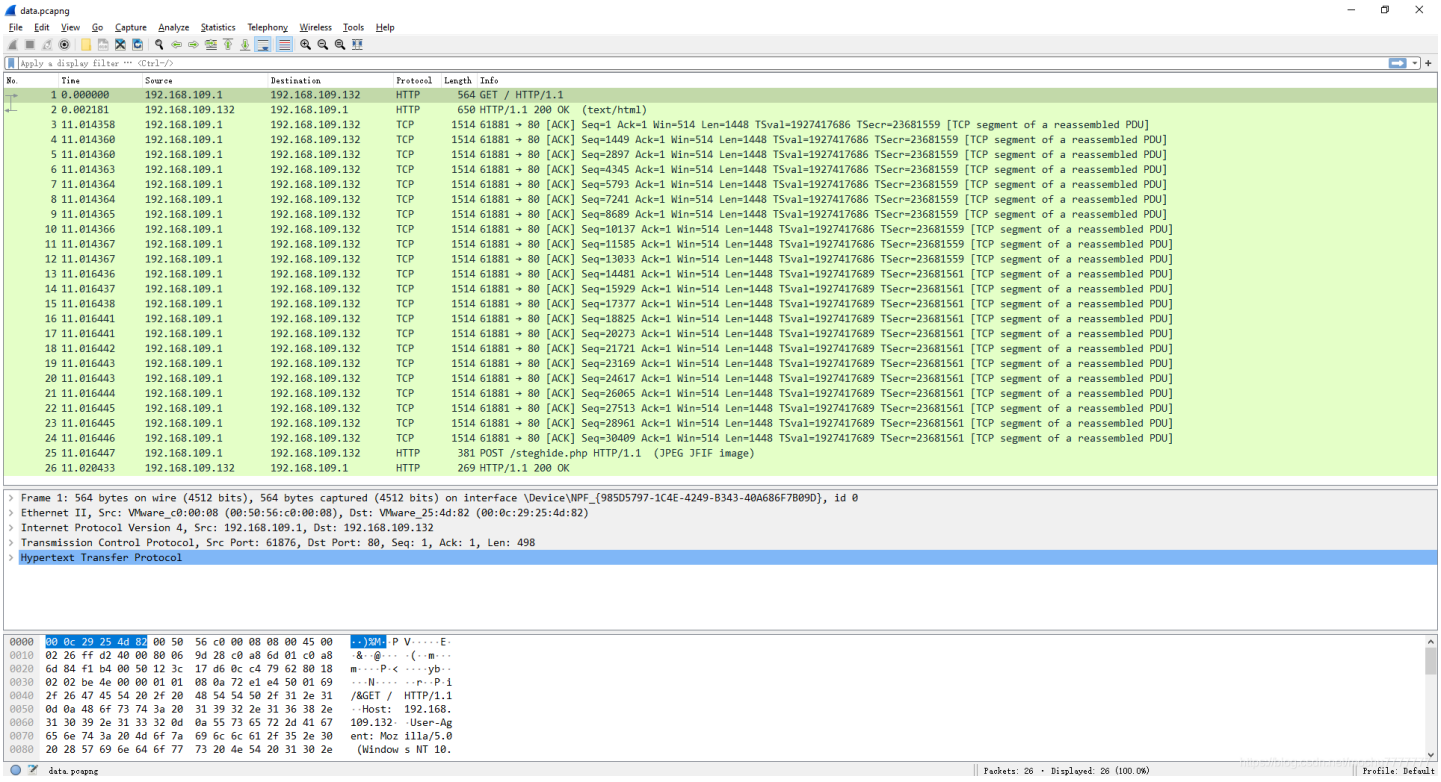
本文链接：https://blog.csdn.net/mochu7777777/article/details/109549349

目录

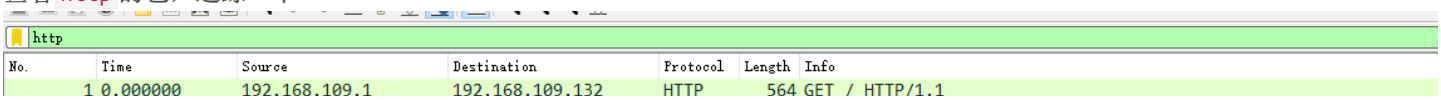
---

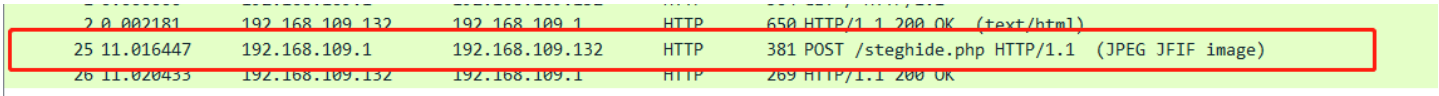
upload

下载附件，打开是流量包文件，wireshark打开

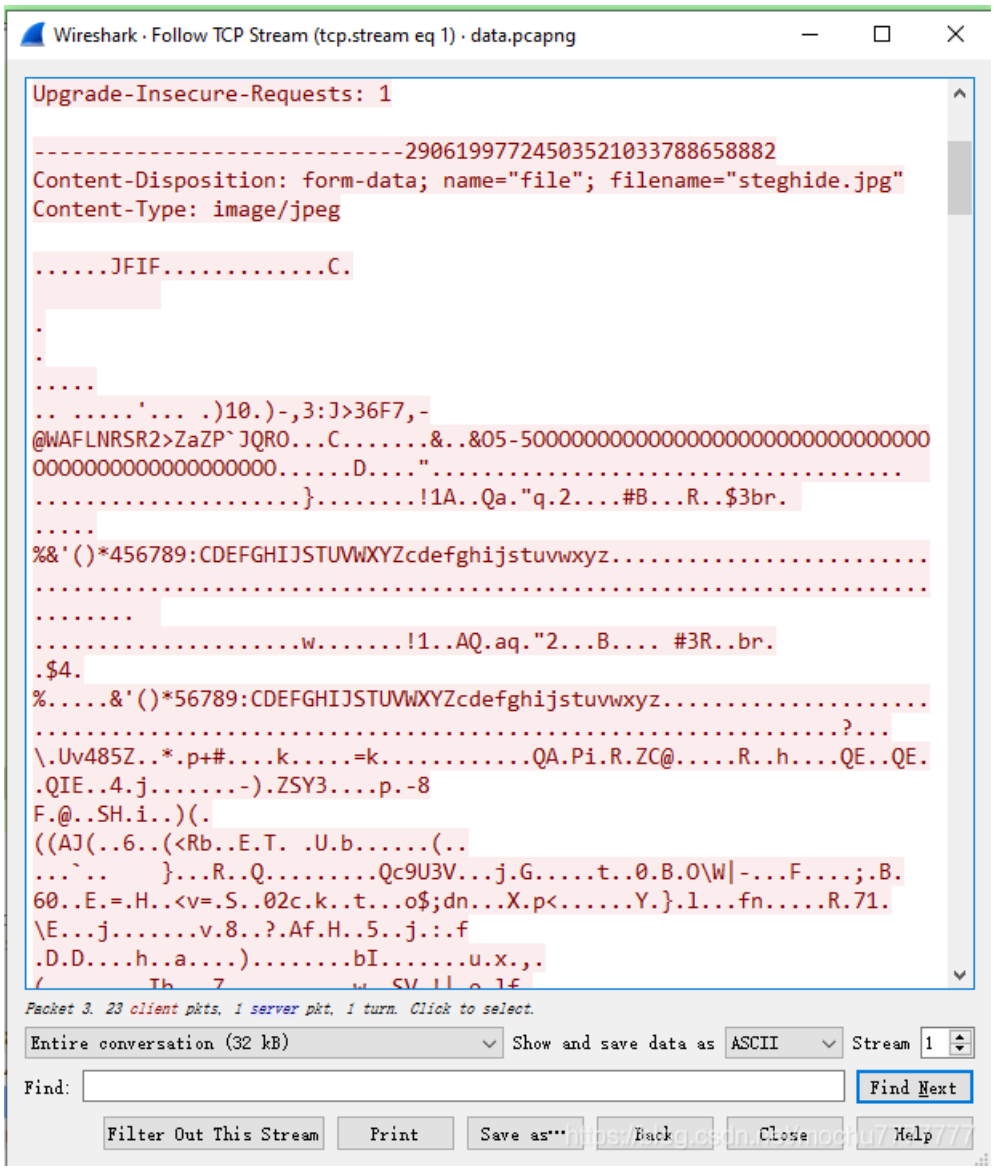


查看 `http` 的包，追踪一下

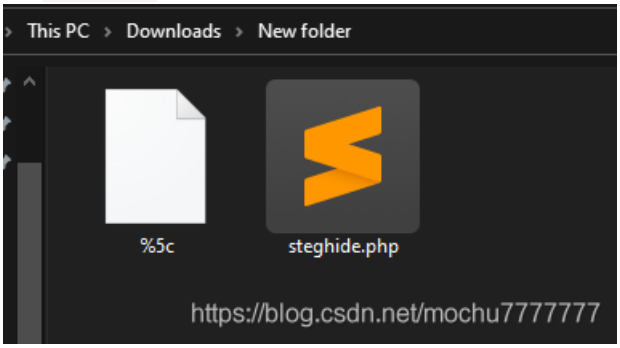




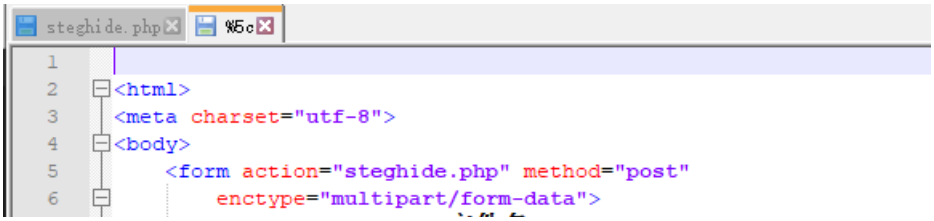
很明显是POST上传的图片



`File->Export Object->HTTP...` 将文件 `Save all` 保存出来，得到如下：

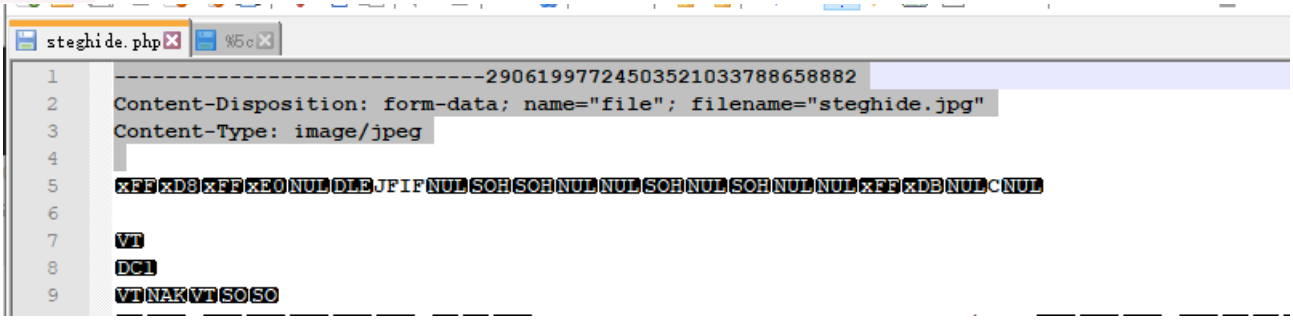


`%5c` 有提示 `steghide` 隐藏



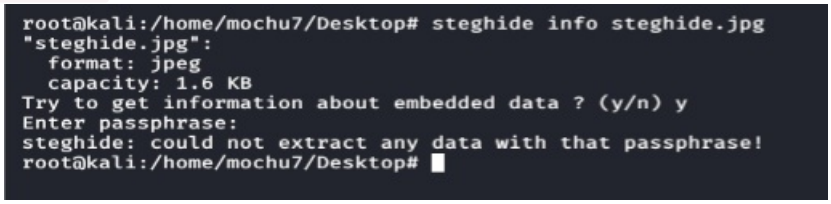
```
        <label for="file">文件名:</label>
        <input type="file" name="file" id="file" />
        <input type="submit" name="submit" value="提交" />
    <!--i use steghide with a good password-->
    </form>
</body>
</html>
```


`steghide.php` 用notepad++打开



去掉前面这四行，保存修改后缀为 `jpg` 或者 `png` 都行，得到如下图：

然后把照片丢进 `kali` 使用 `steghide` 工具提取隐藏信息



有密码，在网上找个爆破steghide密码的脚本，如下：

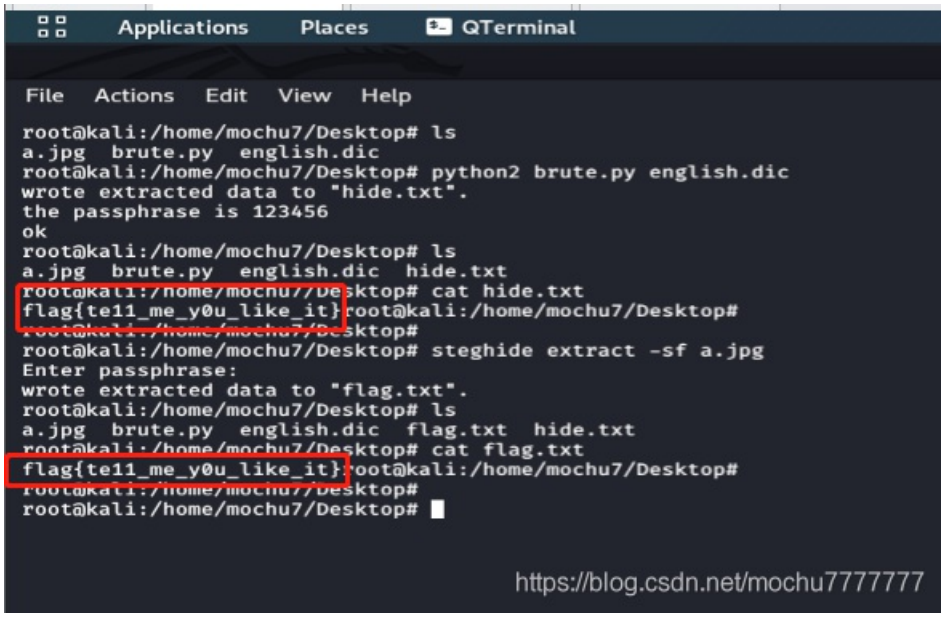
```python
# -*- coding: utf8 -*-
#python2
from subprocess import *

def foo():
    stegoFile='a.jpg'#这里填图片名称
    extractFile='hide.txt'#输出从图片中得到的隐藏内容
    passFile='english.dic'#字典，用的是Advanced Archive Password Recovery的字典

    errors=['could not extract','steghide --help','Syntax error']
    cmdFormat='steghide extract -sf "%s" -xf "%s" -p "%s"'
    f=open(passFile,'r')

    for line in f.readlines():
        cmd=cmdFormat %(stegoFile,extractFile,line.strip())
        p=Popen(cmd,shell=True,stdout=PIPE,stderr=STDOUT)
        content=unicode(p.stdout.read(),'gbk')
        for err in errors:
            if err in content:
                break
        else:
            print content,
            print 'the passphrase is %s' %(line.strip())
            f.close()
            return

if __name__ == '__main__':
    foo()
    print 'ok'
    pass
```



密码是：`123456`

`hide.txt`已经提取了隐藏的flag的内容，或者也可以`steghide extract -sf a.jpg`然后输入密码，得到`flag.txt`

```
flag{te11_me_y0u_like_it}
```

签到

老天爷，哪次比赛让我签到拿个一血也行啊~

```
flag{welcome_to_qwb_S4}
```

问卷调查

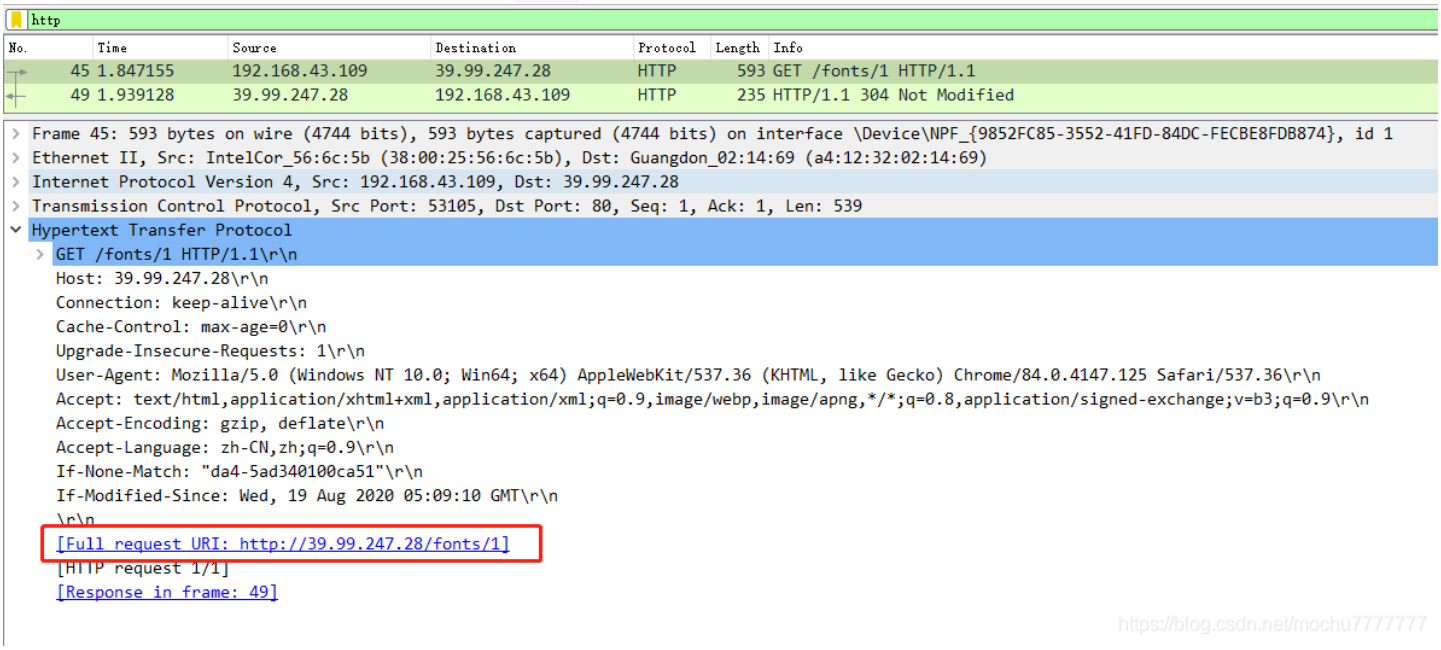
枯了，比赛的时问卷调查最后出来的，竟然没看到，发现的时候已经结束了...orz

```
flag{Welc0me_tO_qwbS4_Hope_you_play_h4ppily}
```

miscstudy

```
Hint: 本题目flag由7个部分构成，第一个部分为flag{level1...，最后一个部分为 !!!} 每一关都会存有flag的一部分，将所有flag的字符串拼接即为最后flag
```

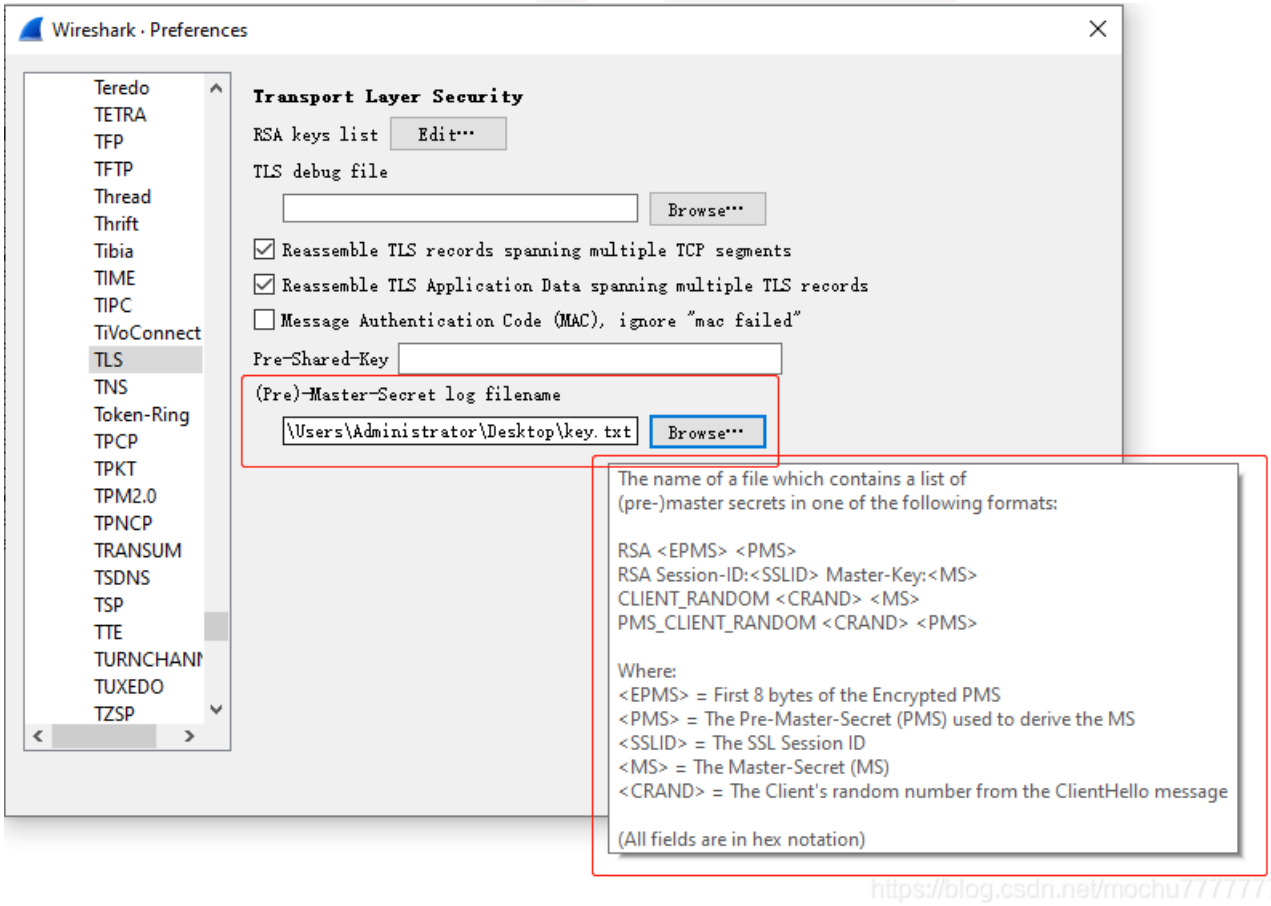
首先下载附件解压是一个流量包，打开后筛选`http`的包，访问这个url



得到`level1`和`level2`的flag


```
flag{level1_begin_and_level2_is_come
```

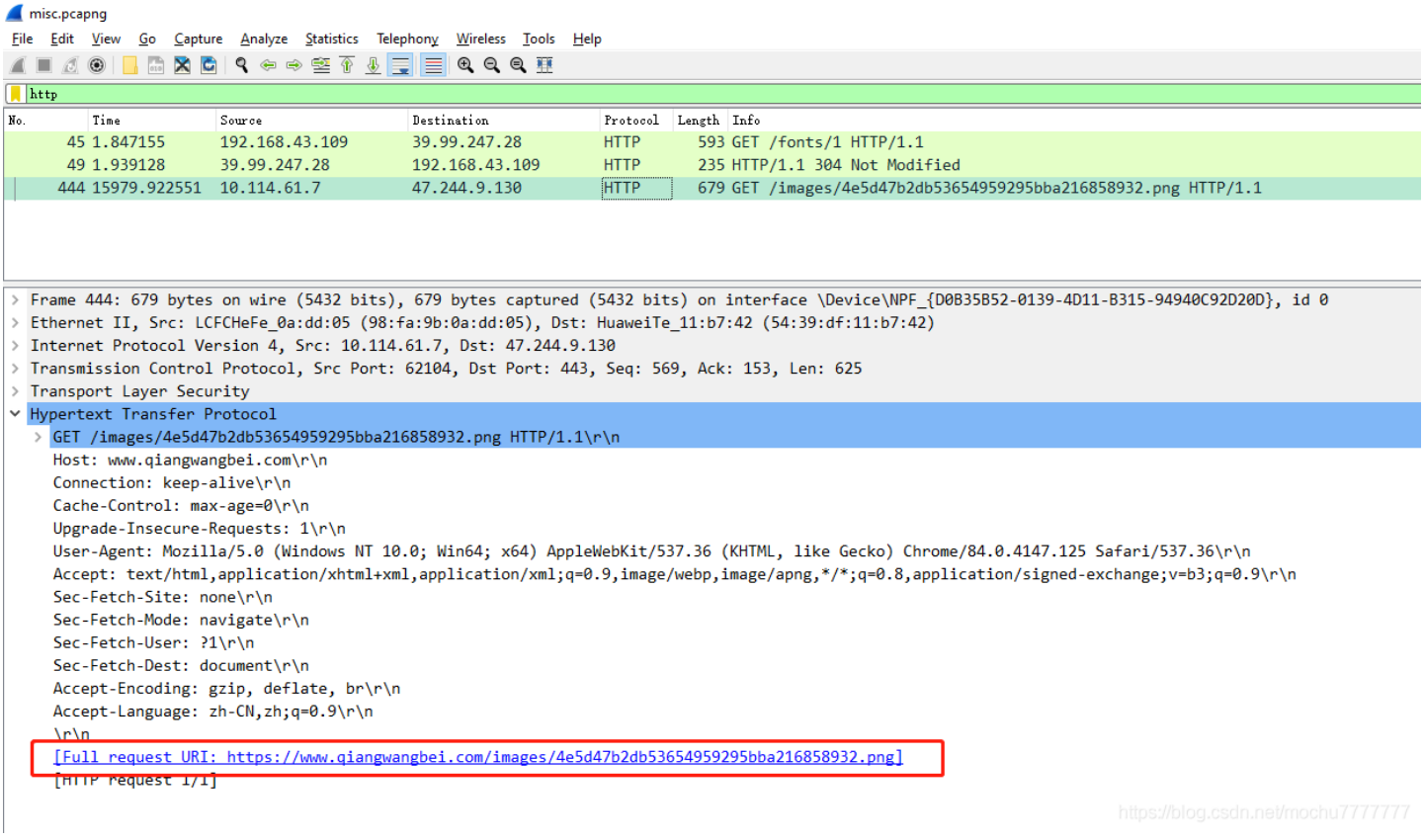
除了flag之外，页面中这些其他的参数应该很明显就是`TLS`协议的`Master-Secret log file`

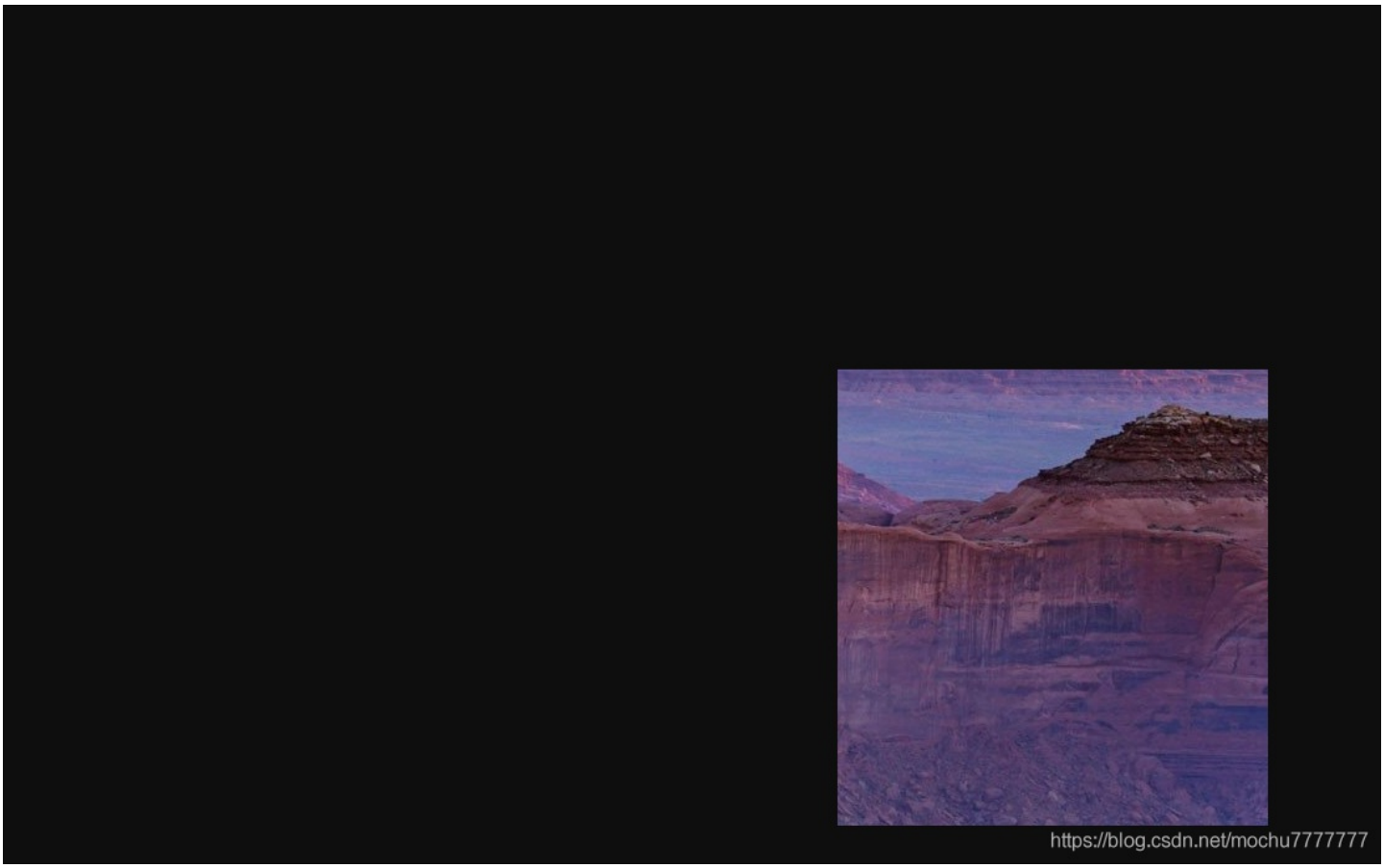


讲这些参数保存为`ket.txt`

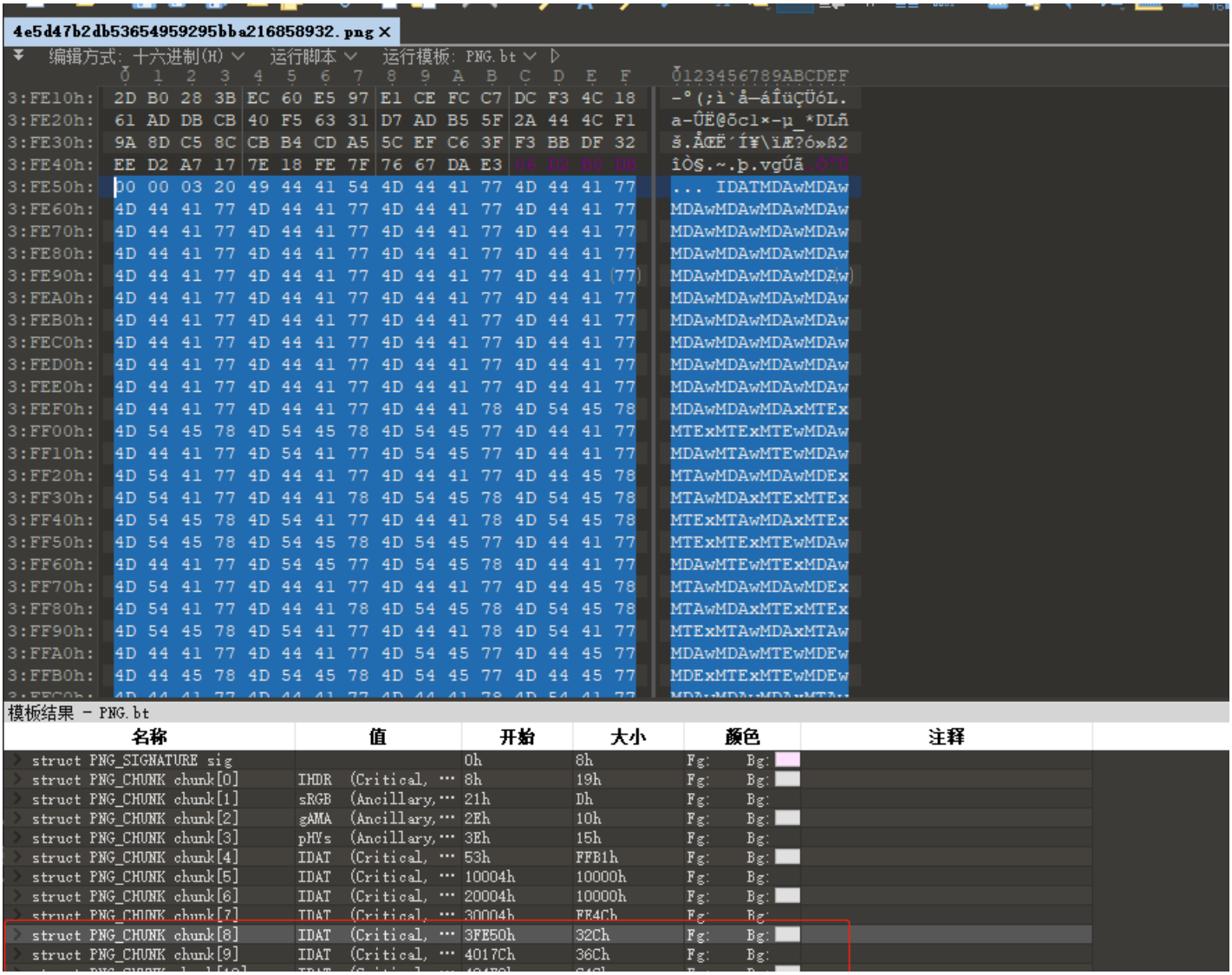
在wireshark中，`Edit->Preferences->Protocols->TLS->(Pre)-Master-Secret log filename`中选择`ket.txt`然后点击`OK`

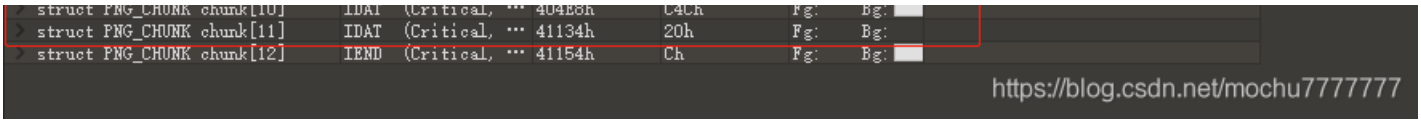
添加成功后，再次查看http协议的包，发现多了个包



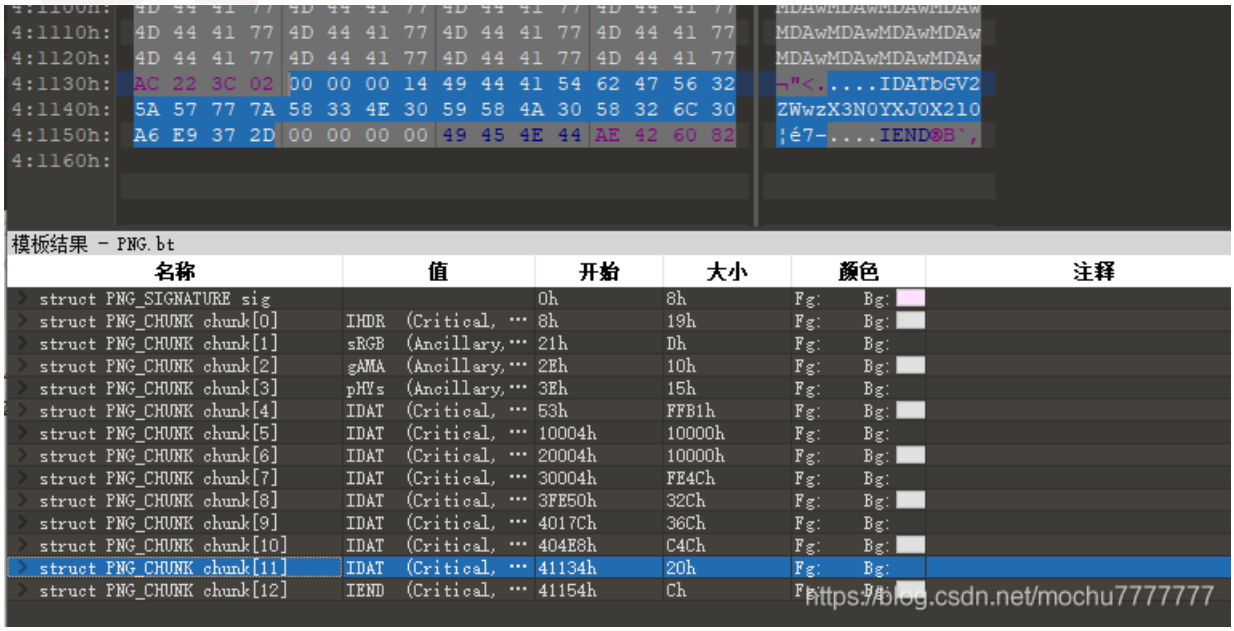


保存图片，使用010 Ediotr或者winhex之类的16进制编辑工具查看





`chunk8-chunk11` 的 `IDAT标志` 后都跟着一串类似base64的编码，而且除了 `chunk11` 其他 `chunk` 的base编码看着应该就是大量的 `01`，其中 `chunk11` 中的 `IDAT` 后的内容比较不一样，复制出来解密得到：




```
level3_start_it
```

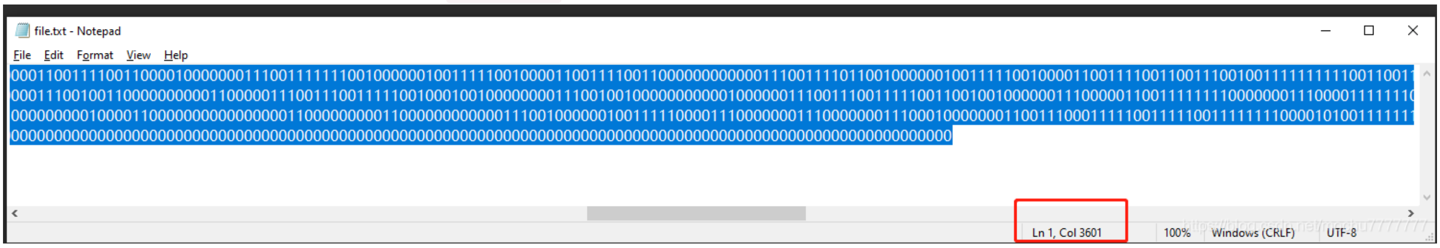
`chunk8-chunk10` 三段中得到base64编码如下：

MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAxMTExMTExMTExMTEwMDAwMDAwMTAwMTEwMDAwMTAwMDAwMDAwMDExMTAwMDAx
MTExMTExMTExMTAwMDAxMTExMTExMTExMTEwMDAwMDAwMTEwMTExMDAwMTAwMDAwMDAwMDExMTAwMDAxMTExMTExMTExMTAwMDAxMTAwMDAwMDAw
MTEwMDEwMDExMTExMTEwMDEwMDAwMDAwMDAxMTAwMTExMDAxMDAwMDAwMDAwMTAwMDAxMTAwMDAwMDAwMTEwMDEwMDAwMTEwMTEwMDEwMDAwMDAw
MDAwMDAxMTExMDAxMDAwMDAwMDAwMTAwMDAxMTAwMTExMTAwMTEwMDEwMDAwMTAwMTEwMDEwMDExMTAwMDAwMDExMTAxMDAxMDAxMTExMTAwMTAw
MDAxMTAwMTExMTAwMTEwMDAwMTAwMDAwMDAxMTEwMDExMTExMTEwMDEwMDAwMDAxMDAxMTExMTAwMTAwMDAxMTAwMTExMTAwMTEwMDAwMDAwMDAw
MDAxMTEwMDExMTEwMTEwMDEwMDAwMDAxMDAxMTExMTAwMTAwMDAxMTAwMTExMTAwMTEwMDExMTAwMTAwMTExMTExMTExMTAwMTEwMDExMTAwMDAw
MDAxMTExMTAwMTAwMDAxMTAwMDAwMDAwMTEwMDExMDAwMDAwMDAwMTExMTAwMTEwMTEwMTEwMDAwMDAxMDAwMDAwMDAwMTAwMDAxMTAwMDAwMDAw
MTEwMDEwMDExMDAwMDAwMDExMTAwMDExMTExMTEwMDExMDAxMDAwMDAwMDAwMTAwMDAxMTExMTExMTExMTEwMDEwMDExMDAxMDAxMDAxMTAwMTEw
MTEwMDEwMDExMDAxMTExMTExMTExMTAwMDAxMTExMTExMTExMTEwMDEwMDExMDAxMDAxMTAwMTAwMTEwMDEwMDExMDAxMTExMTExMTExMTAw
MDAwMDAwMDAwMDAwMDAwMDEwMDExMTExMTEwMDEwMDExMDAwMTExMDAwMTAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDEwMDAxMTEx
MTAwMDEwMDExMDAwMDExMTAxMTAwMDAwMDAwMDAwMDAwMDAwMDAxMTEwMDExMDAwMTExMTAwMDAwMDAxMDAwMDAwMDExMDAwMDAxMTExMTExMDAx
MTExMDAwMTExMDAwMDAwMTEwMDAwMDAwMDAwMDAwMDAxMDAwMDAwMDExMDAwMDAwMDAwMDExMDAxMDAwMDAwMDAwMDAwMDAwMDEwMDAwMDAx
MDAwMDExMTAwMDAxMDAwMDAwMTExMTAwMTEwMDAwMDExMDAwMDAwMDEwMDAwMDAwMDAxMTEwMDExMDAxMTExMTEwMDAwMDAwMDAwMDEwMDAx
MDAxMTEwMDExMTExMDAwMDEwMDExMDAwMDAxMTEwMDExMDAxMTExMTEwMDAwMDAwMDAwMDExMDAxMDAxMTEwMDExMTExMDAwMDEwMDExMDAw
MDAxMTAwMTExMDAwMDAwMDExMTExMDAwMDAwMDEwMDAwMTAwMDAwMTAwMTAwMTExMDAwMDAwMTAwMTAwMDAwMDAxMTAwMDAwMDAwMTExMDExMDAw
MDAwMTExMDAwMTAwMDAxMTAwMDAwMTExMDAwMDAwMDAwMDAwMDExMTAwMDAwMTExMTEwMDExMDAwMDAxMTExMTExMTAwMDAxMTAwMDExMTEx
MTEwMDEwMDAwMDAwMDAwMDAwMTExMDAxMDAxMTAwMDAwMDAwMDAxMTAwMDAwMTExMDAxMTEwMDExMTExMDAxMDAwMTAwMTAwMDAwMDAwMTExMDAx
MDAxMDAwMDAwMDAwMDAxMDAwMDAwMTExMDAxMTEwMDExMTExMDAxMTAwMTAwMTAwMDAwMDExMTAwMDAwMTEwMDExMTExMTExMDAwMDAwMDExMTAw
MDAxMTExMTExMDAwMDAxMTEwMDExMTAwMDAwMDExMDAwMDAwMDEwMDExMTExMTEwMDAwMDAwMDExMTAwMDAxMTExMTExMDAwMDAwMTEwMDAxMTAw
MDAwMDAwMDAwMTAwMDEwMDAwMDAwMDAwMDExMDAwMDAxMTEwMDExMDAxMDAwMDAwMTAwMTAwMTAwMDAwMDAwMDAwMDAwMTEwMDAw
MDAwMDAwMDAwMDAwMDAxMTEwMDExMDExMTAwMDAwMDAwMDAwMDAwMDEwMDAxMDAxMTExMTEwMDAwMTAwMDAwMDAwMTAwMDAwMDExMTExMTExMTAx
MTExMTAwMDAwMDAwMDAwMDExMDAwMDAwMDAxMTEwMDAwMDAwMTExMTExMTAwMTEwMDAwMDAxMTAwMDAwMDAwMDEwMDAwMDAwMDAwMDExMTAwMDAw
MDAxMTEwMDAwMDAwMTExMTExMTAwMTExMDAwMDAwMTAwMDAwMDEwMDAwMDAwMDAxMTEwMDAwMTAwMTExMTExMTExMTExMDAwMDAwMDExMTAw
MTExMDAwMDExMDAwMDAxMTAwMDAwMDAwMDAxMTExMDAwMTEwMDAxMDAxMTAwMTExMDAwMDAwMDExMDAwMDExMTAwMDExMDAwMDAwMDAwMDAwMDAw
MDAxMTExMTAwMTExMDAxMTAwMTAwMTAxMTExMTExMTExMDAxMTExMTAwMDAxMDAxMTEwMDExMTExMTAwMDAxMTAwMDAwMTExMTEwMDAwMTAwMDAw
MDAwMDEwMDAwMDAwMTExMTAwMDExMDAwMDAwMDExMTAwMTAwMDAxMDAwMDAwMTExMTEwMDAwMTAwMDAwMDAwMDEwMDAwMDAwMTExMDAwMDExMDAw
MDAwMDExMTAwMTAwMDAwMDExMDExMTAwMDAwMDExMTAwMDAwMDAwMDAwMDAxMTEwMDAwMDExMTExMDAwMDAxMTAwMTAwMDAwMTExMTExMTAw
MDAwMDExMDAwMDAwMDAwMDAwMDAwMDExMTAwMDAwMDAxMTAxMDAwMDAwMTAwMTAwMDAxMTExMTAwMDAwMTEwMDEwMDAwMTAwMDAwMDAwMDExMTEx
MDAwMDEwMDAwMDAwMDAxMTAwMDEwMDAwMDAwMTExMTAwMDAwMDAwMDEwMDAwMTEwMDAwMDAwMDAwMDAwMDAwMTEwMDAwMDAwMDAxMTAwMDAwMDAw
MDAwMDExMTAwMTAwMDAwMDEwMDExMTExMDAwMDExMTAwMDAwMTEwMDAwMTExMDAwMTAwMDAwMDAxMTAwMTExMDAwMTExMTEwMDExMTEx
MDAxMTExMTExMTAwMDAxMDEwMDExMTExMTExMTAwMDExMDAwMDAxMTAwMDAwMTExMTEwMDExMTExMDAxMTExMTExMTAwMDAxMTEwMDExMDEx
MTExMTAwMDExMDAwMDAwMDAwMDAwMDAwMDAwMDEwMDExMDAxMDAxMTEwMDAwMDAxMTEwMDAxMTExMDAwMDAxMTAwMTAwMDAwMDAwMDAwMDAw
MDAwMDEwMDExMDAwMDAxMTAwMDAwMDAwMTEwMDExMDAwMDAxMTAwMTAwMTAwMDAxMTExMTExMTExMTEwMDEwMDExMDAwMTExMDAwMTEwMDAw
MTExMTAwMDExMDAxMDAxMTAwMTExMTAwMDAxMTAwMDAwMTEwMDAwMTExMTAwMDAwMDExMTExMTAwMDAxMTEwMDAxMDAwMDAxMTAwMTExMDAw
MDAxMTAwMDAwMDAwMTEwMDAwMTExMTAwMDAwMDExMTExMTAxMTEwMDExMDAwMDAxMTAwMTExMDAwMDAxMTAwMTExMTAwMTEwMDAwMTAwMDAx
MDAwMDAwMTAwMDAxMDAxMTEwMDExMTExMTExMTAwMDAwMDAxMTAwMTExMTAwMTEwMDAwMTAwMDAxMDAwMDAxMTAwMDAxMDAxMTAwMDAxMTEx
MTExMDAwMDAwMDAxMTAwMTExMTAwMTEwMDExMDAwMTExMTExMDAwMTExMTEwMDAwMTEwMDAwMTExMDAwMDExMDExMTAwMDAxMTAwMTExMTAw
MTEwMDAwMDExMDAwMTEwMDAwMDAwMDAwMTEwMDAwMDAwMDExMDExMTAwMDAxMTAwMTExMTAwMTEwMDAwMDExMTExMTEwMDAwMDAwMTAw
MDAxMTEwMDExMDAwMDAwMDAwMDExMTAwMDAxMTAwMDAwMDAwMTEwMDExMTAwMTExMDAwMDExMDAxMTExMTAwMDExMTExMTExMTExMTAwMDAw
MDAxMTAwMDAwMDAwMTEwMDExMTAwMTAwMTExMDAwMDExMDAxMTExMTAwMDExMTExMTExMTExMTAwMDAwMDAxMDExMTExMTExMDEwMDEwMDExMTEx
MTEwMDExMTExMDAxMDAxMTEwMDExMTExMTAwMDExMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw

base64解密得到：

```
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000001111111111110000000100110000100000000011100001111111111110000111111111111000000011011100010000000001110
0001111111111110000110000000011001001111111001000000000110011100100000000010000110000000011001000011011001000000
0000011110010000000001000011001111001100100001001100100111000000111010010011111001000011001111001100001000000011
1001111111001000000100111110010000110011110011000000000000111001111011001000000100111110010000110011110011001110
0100111111111100110011100000000111110010000110000000011001100000000011110011011011000000100000000010000110000000
1100100110000000111000111111100110010000000001000011111111111001001100100100110011011001001100111111111111000011
1111111111001001100100110000010011001001100111111111110000000000000000000100111111100100110001110001000000000000
0000000000000000000100011111000100110000111011000000000000000000001110011000111100000000100000000110000011111110 01
1110001110000000110000000000000000001000000001100000000000110010000000000000000010000000100001110000100000001111001100
0001100000000100000000011100110011111100000000000000010001001110011111000010011000000111001100111111000000000000001
1001001110011111000010011000000110011100000000111110000000010000100000100100111000000100100000000110000000011101100
0001110001000011000000111000000000000000000111000000111110011000000111111110000110001111111001000000000000000111001 0011
0000000000110000011100111001111100100010010000000001110010010000000000000100000011100111001111100110010010000001110
0000110011111111000000001110000111111110000001110011100000001100000000100111111100000000111000011111111000000110001100
0000000001000000100000000000000011000000111001100100000010010000000000000000000110000000000000000000000111001101110
0000000000001000100111111100001000000000100000011111111101111110000000000000110000000001110000000111111100110000000110
0000000010000000000001110000000011100000000111111100111000000010000000000100000000001110000100111111111111100000000111
1110000110000001100000000000111100011000100110011100000000110000111000110000000000000000000111111001110011001001011111
1111110011111000010011100111111000011000000111110000100000000010000000111100011000000001110010000100000001111100001
0000000010000000111000011000000001110010000000110111000000011100000000000000000001110000001111100000110010000001111111100
0000110000000000000000011100000000110100000001001000011111000000110010000100000000000111110000100000000000110001000000001
1110000000000100001100000000000000000011000000000011000000000000011100100000001001111100001110000000011100000000111000 10
0000001100111000111110011111001111111100001010011111111110001100000100011100011111001111100111111111100001110011011
1111000110000000000000000000010011001001110000000111000111100000110000010000000000000000000010011000000110000000001100
0001100000011000000100001111111111110010011000111000110000111100011001001100111100001100000000110000111100000001111
1111001110001000000110011100000011000000000110000111100000000111111110011100110000001100111000000110011110011000010000 1
0000001000010011100111111111000000000000110011110011000010000100000110000100110000111111100000000000000110011110011 00
1001100011111111100011111000011100001101110000110011110011000000110001100000000000000110000000000001101110000110011
1100110000001111111000000000100001110011000000000000111000011000000000110011100100111000011001111100011111111111110000
0011000000000110011100100111000011001111100011111111111110000000010111111110100100111111110011111001001110011111111000
0001100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000
```

将base64解密得到的这些二进制保存为`file.txt`



总长度：`3600`

使用01二进制转二维码脚本：

```
import PIL
from PIL import Image

MAX = 60
img = Image.new("RGB",(MAX,MAX))
i = 0
str = "000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000001111111111110000001001100001000000000111000011111111111000011111111111100000001101110001000000
000111000011111111111000011000000001100100111111100100000000011001110010000000010000110000000011001000011011001
000000000000011110010000000001000011001111001100100001001100100111000000011101001001111100100001100111100110000100
000001110011111110010000001001111100100001100111001100000000000011100111101100100000100111110010000110011111001
100111001001111111111001100111000000011111001000011000000001100110000000001111001101101100000010000000010000110
000000011001001100000001110001111111001100100000000100001111111111110010011001001001100110110010011001111111111
100001111111111110010011001001100000100110010011001111111111100000000000000000010011111110010011000111000100000
000000000000000000000000000100011111000100110000111011000000000000000000001110011000111100000001000000011000001111
111001111000111000000110000000000000000010000001100000000011001000000000000000100000100001110000100000011100000
100110000011000000010000000001110011001111110000000000000010001001110011111000010011000001110011001111110000000
000000110010011100111110000100110000011001110000000111110000000100001000001001001110000001001000000011000000011
101100000011100010000110000011100000000000000001110000011111001100000111111110000110001111111001000000000000111
001001100000000001100000111001110011111001000100100000000111001001000000000001000000111001110011111001100100100
000111000001100111111110000000111000011111110000011100111000000110000000100111111100000000111000011111110000011
000110000000000010000001000000000000000011000001110011001000000100100000000000000000110000000000000000000001110011
011100000000000000010001001111110000100000000100000011111111101111100000000000011000000011100000001111111001100
000011000000000100000000000111000000011100000001111111001110000001000000000100000000011100001001111111111100000
001110011100001100000110000000000011110001100010011001110000000110000111000110000000000000000011111001110011001001
101111111111001111100001001110011111110000110000011111000010000000100000001111000110000000111001000010000001111
100001000000000100000001110000110000000111001000000110111000000111000000000000000011100000111110000011001000001111
111110000001100000000000000011100000001101000000100100001111100000110010000100000000111110000100000000110001000
000000111100000000010000110000000000000000110000000011000000000001110010000001001111100001110000000111000000001
110001000000011001110001111100111110011111111000010100111111111100011000001000111000111110011111001111111100001111
001101111110001100000000000000000001001100100111000000111000111100000110000010000000000000000010011000001100000
000110000011000001100000100001111111111111001001100011100011000011110011001001100111100001100000000110000111100
000111111111001110001000001100111000001100000000110000111100000011111111001110011000001100111000001100111100011000
010000100000010000100111001111111110000000000110011110011000010000100000110000100110000111111100000000001100111
100110010011000111111111000111110000111000011011100001100111100110000011000110000000000000011000000000001101110000
011001111001100000111111100000001000011100110000000001110000110000000011001110010011100001100111110001111111111
110000000110000000011001110010011100001100111110001111111111100000001011111111010010011111110011111001001110011
111100000011000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000"

for x in range(MAX):
    for y in range(MAX):
        if(str[i] == '1'):
            img.putpixel([x,y],(0,0,0))
        else:
            img.putpixel([x,y],(255,255,255))
        i = i+1
img.show()
img.save("result.png")
```

得到二维码，使用`QR Research`扫描

网盘下载得到压缩包`level4.zip`，解压得到`level4.jpg`

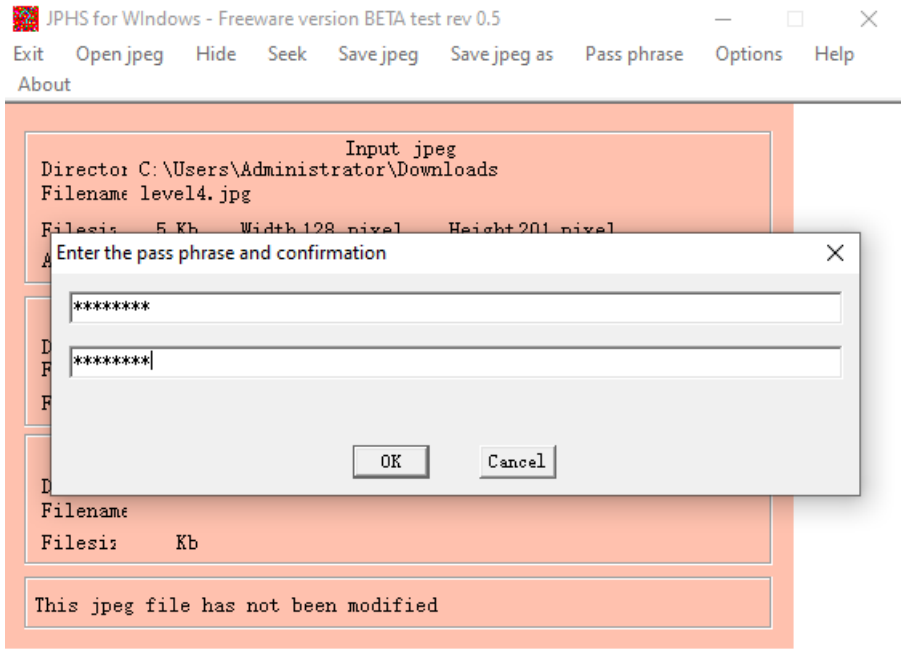
使用`stegdetect`检测发现有`jphide`

```
→ level4 stegdetect level4.jpg
level4.jpg : jphide(*)
→ level4
```

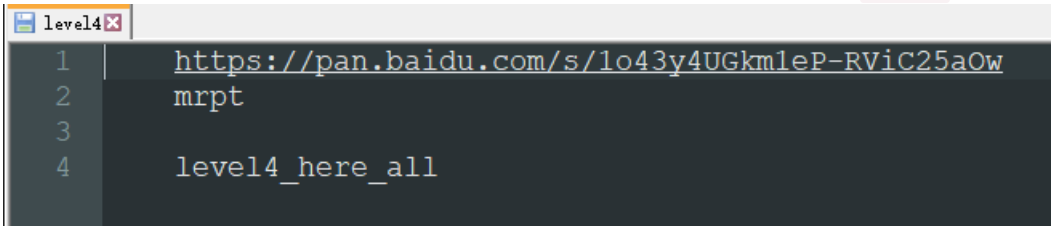
使用`stegbreak`爆破密码，得到密码为：`power123`

```
→ level4
→ level4 stegbreak -r /usr/share/stegbreak/rules.ini -f pawd_list.txt level4.jpg
Loaded 1 files...
level4.jpg : jphide[v5](power123)
Processed 1 files, found 1 embeddings.
Time: 0 seconds: Cracks: 734,       inf c/s
→ level4
```

使用`JPHS`工具打开图片，点击`Seek`输入密码



然后会导出个文件（不需要输入文件后缀什么的，直接随便输入文件名导出即可）这里导出为`level4`

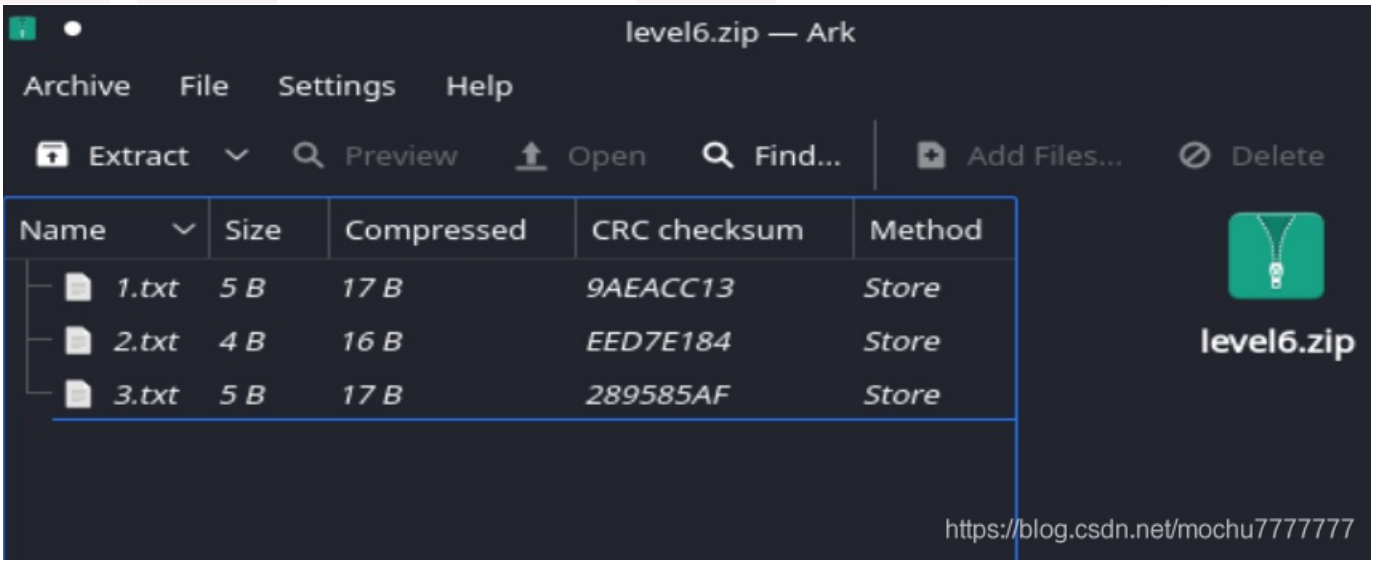



```
level4_here_all
```

网盘下载附件，得到压缩包`level5.zip`，解压，如下：


```
level5_is_aaa
```

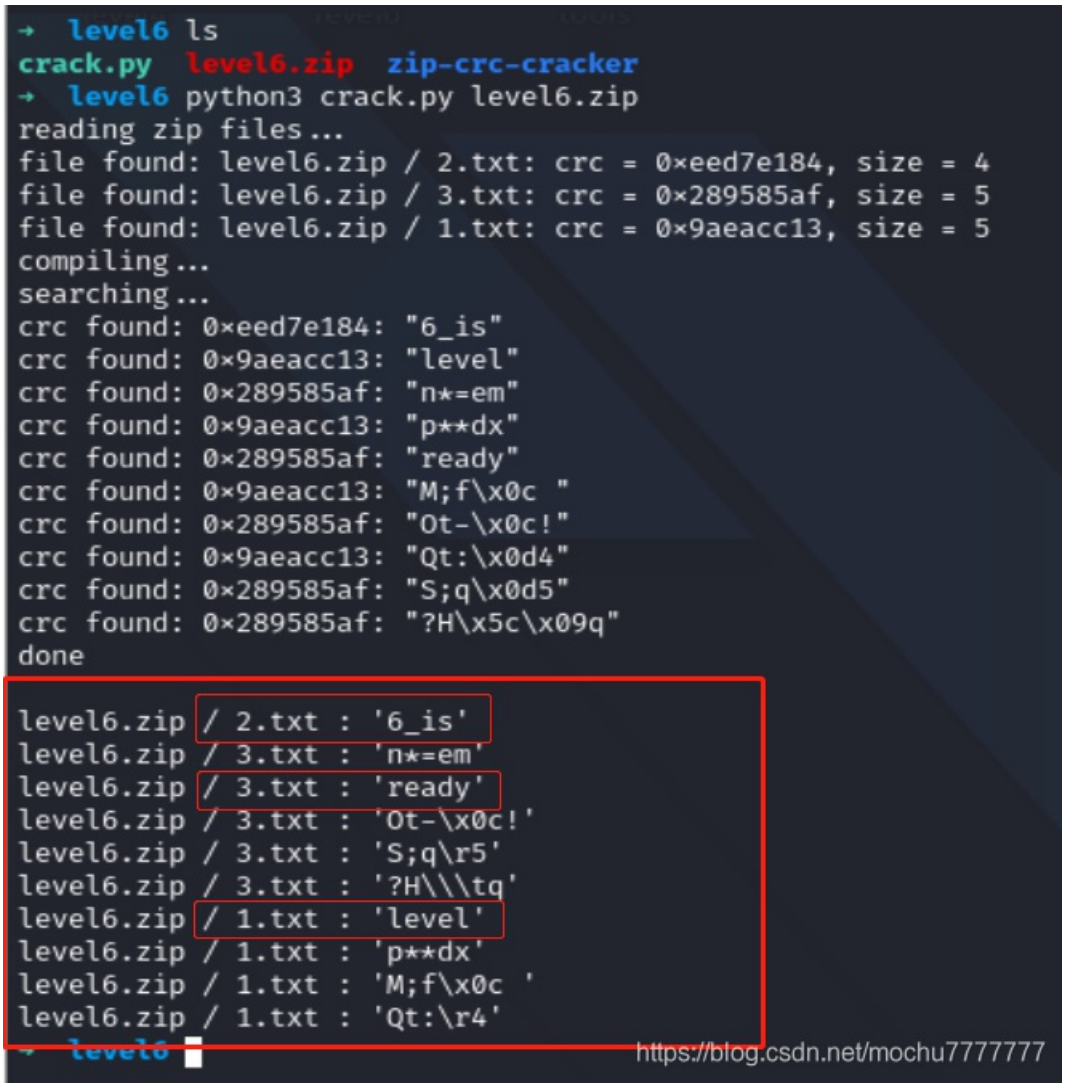
`level6.zip`和`level7.zip`都有压缩密码，解不开，先看到`level6.zip`



原始文件大小都是<=5 Byte，猜测CRC32碰撞，网上很多其他的CRC32碰撞脚本都试了不行，最后找到这个Zip-CRC32碰撞脚本：

Zip-CRC32碰撞脚本: https://github.com/kmyk/zip-crc-cracker

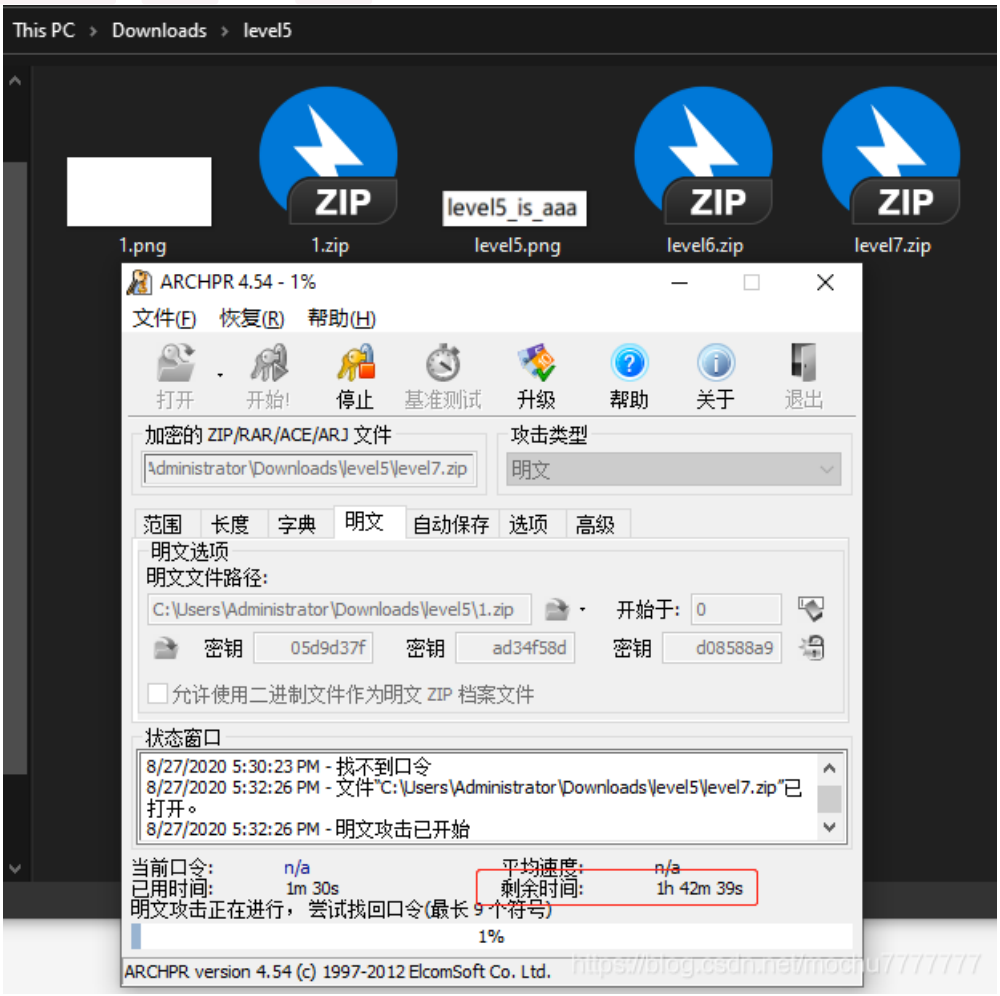
注: 该脚本笔者运行环境为Linux，Windows试过几次不行



按照`1,2,3`的顺序排好

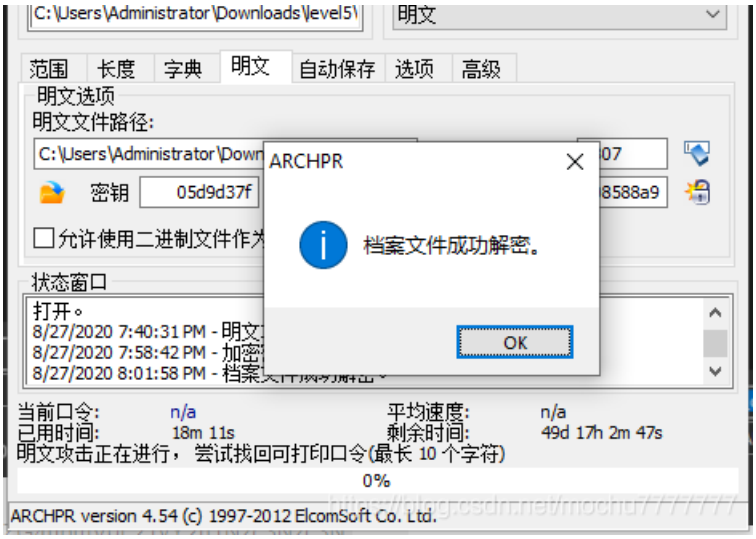
```
level6_isready
```

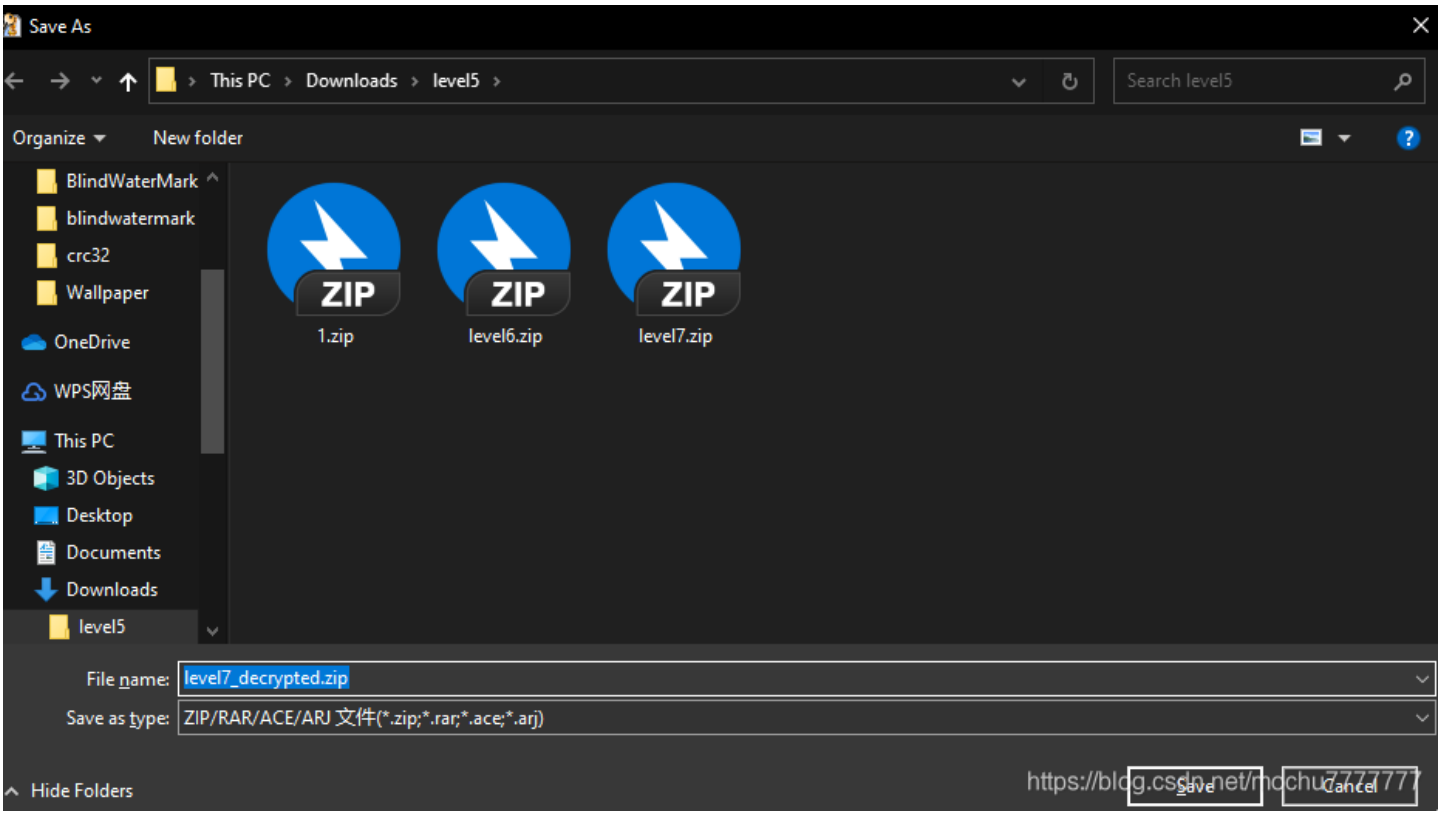
使用 `ARCHPR` 打开 `level7.zip`， `1.png` 压缩成 `1.zip` 添加到明文密钥进行明文爆破



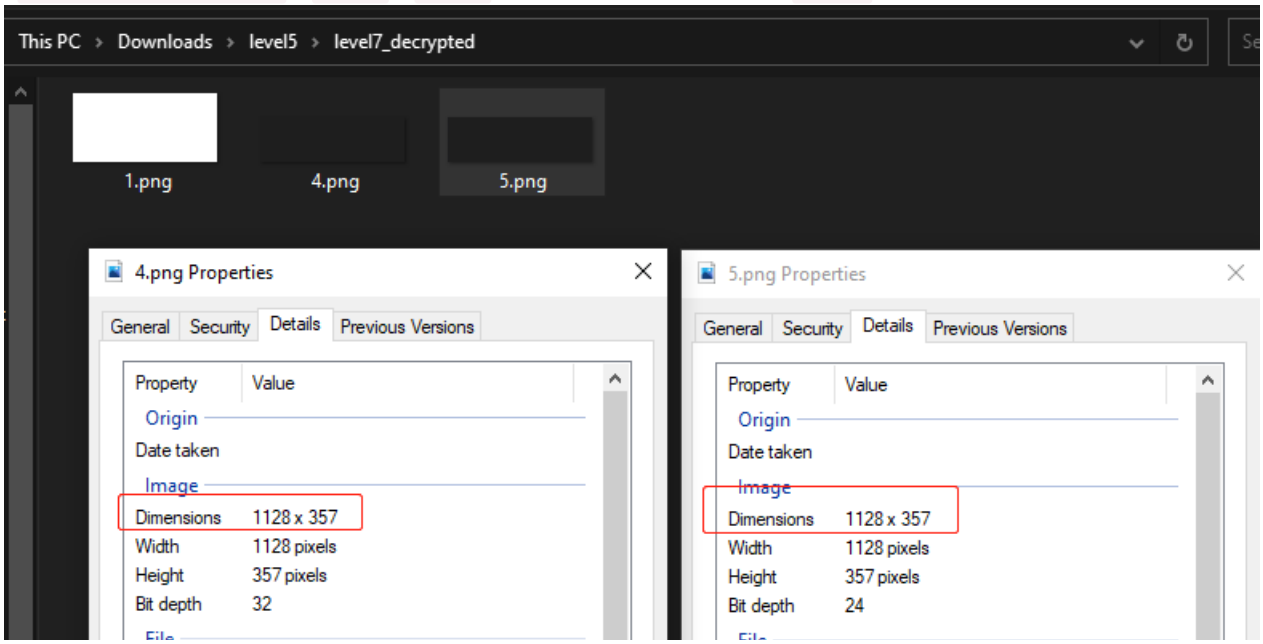
根据网上的师傅的说法，等到 `剩余时间` 小于 `1h` 即可停止

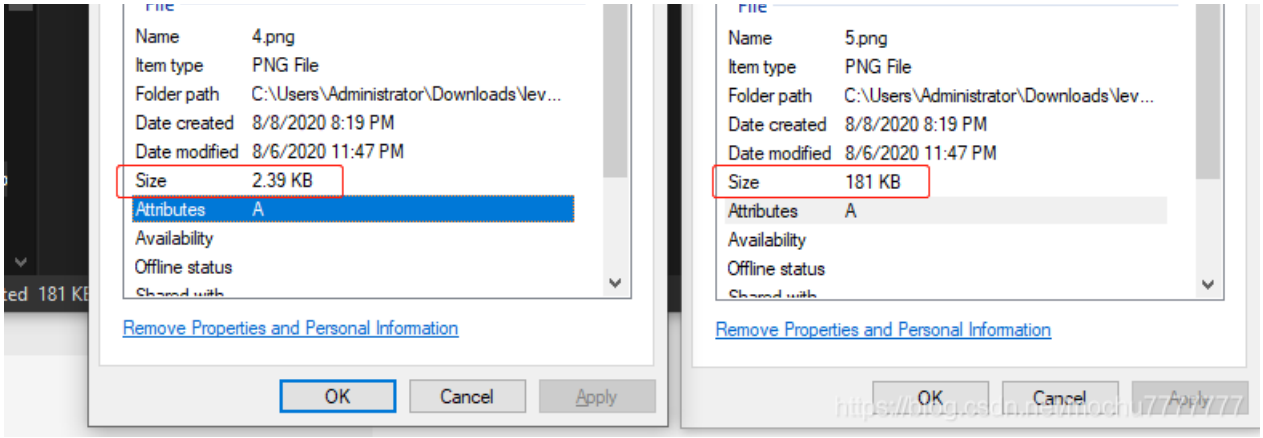
然后会自动保存名为 `level7_decrypted.zip` 的已经解开加密的压缩包





直接解压 `level7_decrypted.zip`， `4.png` 和 `5.png` 分辨率一样，size不一样， `盲水印`



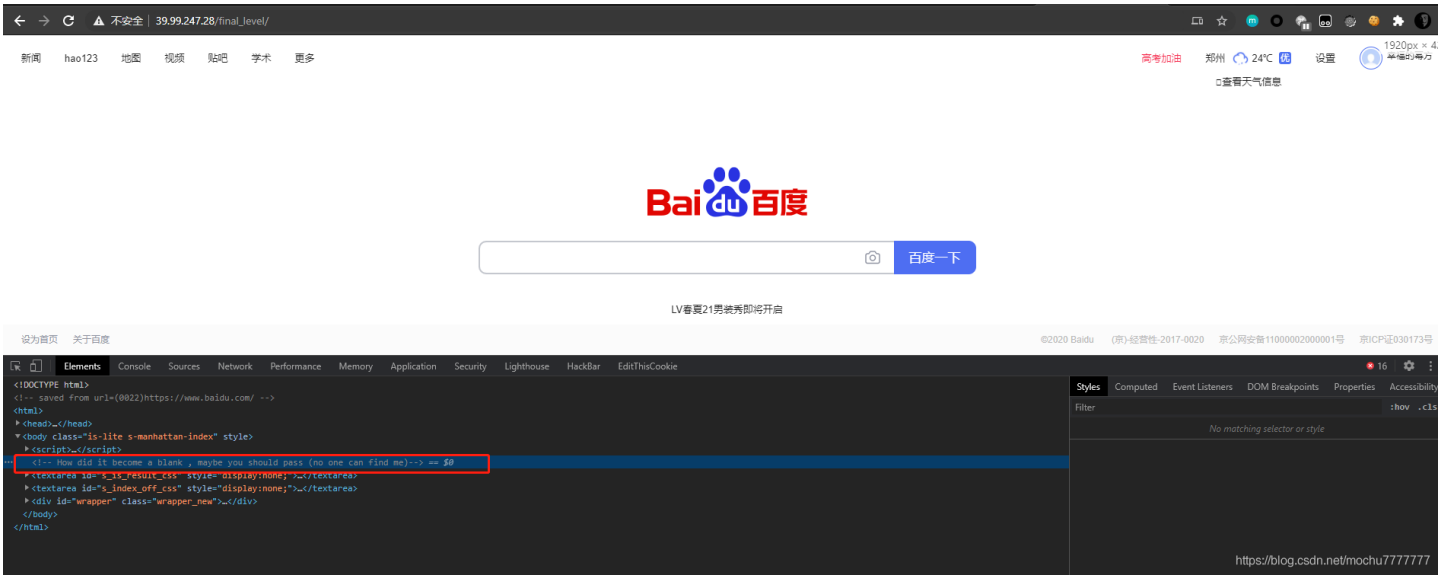


BlindWaterMask: https://github.com/chishaxie/BlindWaterMark

```
PS D:\Tools\Misc\BlindWaterMark> python3 .\bwmforpy3.py decode .\4.png .\5.png result.png
image<.\4.png> + image(encoded)<.\5.png> -> watermark<result.png>
```


```
level7ishere
```

并且得到最后一关的相关地址：http://39.99.247.28/final_level/



html snow隐写

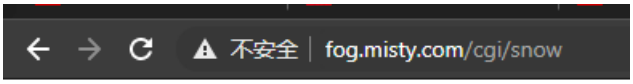
html snow隐写解密网站：http://fog.misty.com/perry/ccs/snow/snow/snow.html

把网址格式填对，Password为题目注释中的括号里的内容

Decryption

URL containing concealed message:
http://39.99.247.28/final_level/
Password: no one can find me
Decrypt

点击`Decrypt`



the_misc_examaaaaaaa_!!!}

```
the_misc_examaaaaaaa_!!!}
```

综上所述，flag为其部分拼接：

```
flag{level1_begin_and_level2_is_comelevel3_start_itlevel4_here_alllevel5_is_aaalevel6_isreadylevel7isherethe_misc_examaaaaaaa_!!!}
```