CTFshow Misc部分 做题记录

原创

FW_Suica 于 2021-06-04 16:22:58 发布 1123 收藏 7
文章标签： 信息安全

目录

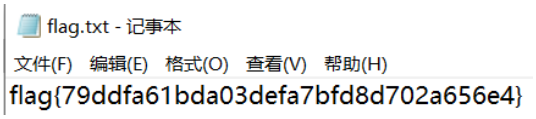
1、杂项签到

下载 附件得到zip文件，内含一个加密的flag.txt 且文件夹名称提示：忘记密码了

考虑伪加密，拖进ZipCenOp梭哈一下

```
E:\ctf\工具\软件\zipcenop>java -jar ZipCenOp.jar r download.zip
success 2 flag(s) found
```

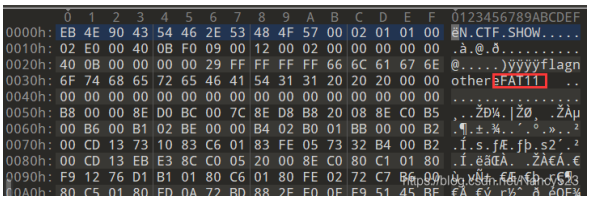
直接解压 得到flag



2、misc2

下载附件 解压 得到无后缀的file文件

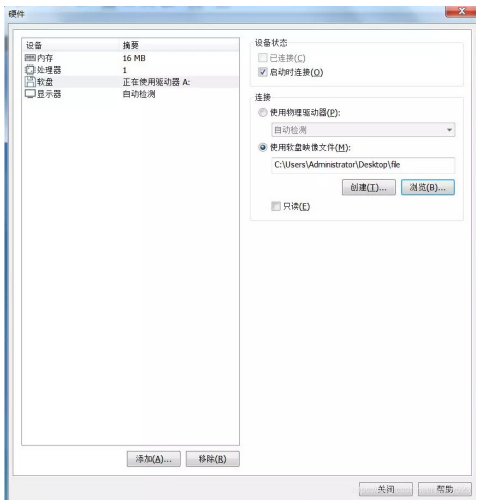
十六进制编译器打开 查看



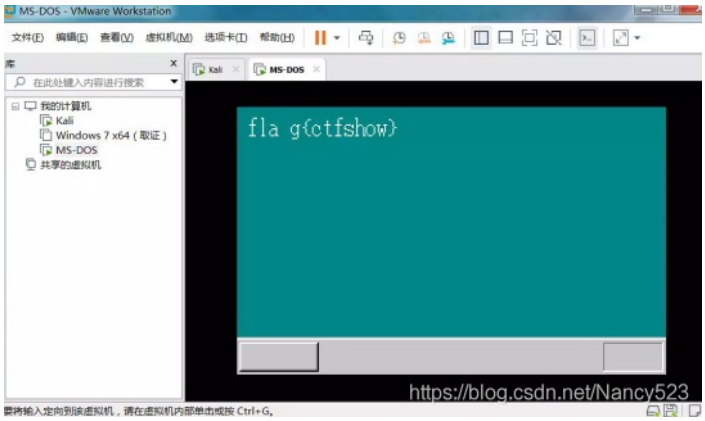
FAT文件，，不懂是啥 参考一下大佬的wp：

> 1.44MB软盘使用的FAT文件系统通常为FAT12，根据FAT12的MBR引导记录结构，第54字节开始的8位应固定为FAT12。而该文件中被修改成了FAT11，需要进行修复。

新建虚拟机并载入软盘镜像：



启动虚拟机 得到flag

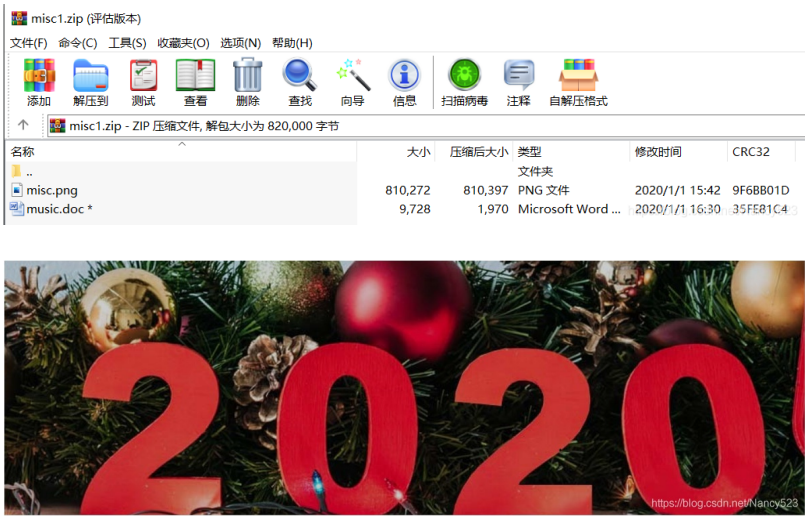


3、miscx

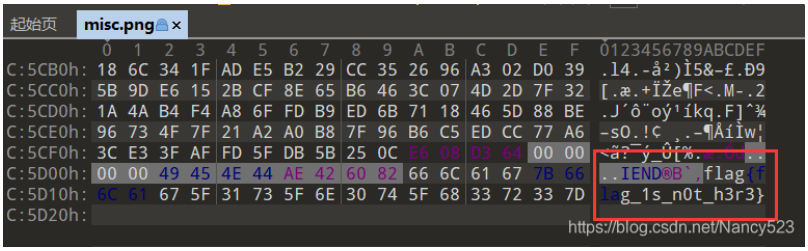
下载得到rar类型附件，内含两个加密的txt文件跟一个未加密的压缩包

附加文本：2020快乐！
rat？or？

先查看其中的misc1压缩包 内含加密的music.doc文件跟一个png后缀的图片

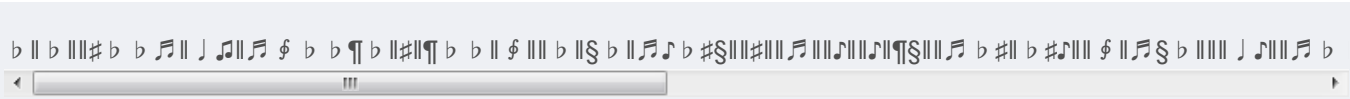


把图片用十六进制编辑器打开



尝试输入flag 错了 说明这是误导信息 结合前面提示 猜想doc文件解压密码为2020

打开发现为一串音符



遇事不决千千秀字（bushi 得到内容

```
U2FsdGVkX1/eK2855m8HM4cTq8Fquqtm6QDbcUu4F1yQpA==
```

观察特征以为为base64加密 解码后乱码 尬住 看达不溜屁去（

得到 为rabbit加密，秘钥为2020 解密后内容为

```
welcome_to_payhelp
```

回去看那俩加密的txt文件 以此为密码解压出了hint.txt

```
VmpKMFUxTXhXWGxVV0dob1RUSjRVVll3V2t0aFJscDBZMGhLYTAxWGVlaFZiRkpUWWtaYVZWSnJXbFpOVjJoeVZY
```

base64解码 且每次解码完后记得把结果末尾的%3Durl转码为=，解码6次后得到

```
welcome_to_2020%0Aflag%20is%20coming...%0Athe%20key%20is%20hello%202020%21
```

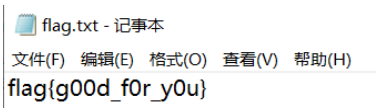
进行url转码

```
welcome_to_2020
flag is coming...
the key is hello 2020!
```

得到密码为

```
hello 2020!
```

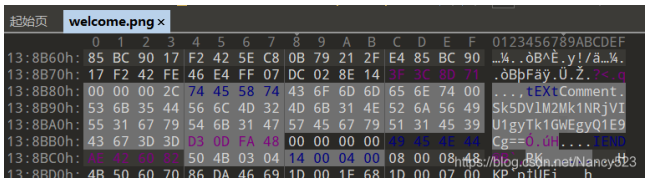
解压flag.txt 得到flag



4、misc50

下载附件解压后得到png图片

惯例 丢进十六进制编辑器 文本末尾观察到额外数据




```
Sk5DVlM2Mk1NRjVlU1gyTk1GWEgyQ1E9Cg==
```

base64解码后得到

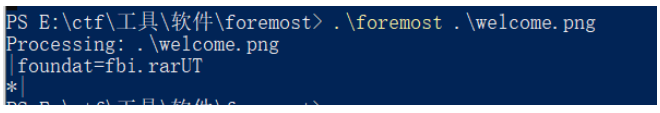
```
JNCVS62MMF5HSX2NMFXH2CQ=
```

base32解码：

```
KEY{Lazy_Man}
```

很明显了 是哪个压缩包的解压密码

把png文件丢去foremost梭哈一下



得到

有提示 base32解码得到

> 123456

再次解压 得到thienc.txt 密码就是123456

解压得到十几m大的txt文件 密密麻麻一堆数字（啥玩意 看wp去 知识盲区

文本为16进制的ASCII编码，但并不全在ASCII可打印字符范围内，且位数既有1位也有2位。注意到37 7A BC AF是7z格式压缩文件的文件头，将十六进制数字对应转为字节后保存成7z格式文件：脚本为

```
f = open('E:/ctf/ctfshow/misc/welcome/thienc.txt').read()

f1 = ''
for i in range(0, len(f), 2):
    n = int(f[i:i+2], 16)
    f1 += chr(n)

f1 = f1.split('0x')[1:]
f2 = b''
for i in f1:
    f2 += bytes.fromhex(i.zfill(2))

f3 = open('C:/Users/Administrator/Desktop/1.7z', 'wb')
f3.write(f2)
```

得到加密的7z格式压缩包 结合之前得到的key 解压得到secenc.txt（内容省略

很明显为base64编码 开开心心跑去解码 诶 好 报错（nmd 看看wp

发现这段字符串经过了多重的base64和base32混合编码。循环用正则表达式匹配判断编码类型然后解码： 脚本为

```
import base64
import re

f = open('E:/ctf/ctfshow/misc/welcome2/1/secenc.txt').read().encode('utf-8')

while True:
    if re.match('^[2-7A-Z=]+$', f.decode('utf-8')):
        f = base64.b32decode(f)
    elif re.match('^[0-9a-zA-Z+/=]+$', f.decode('utf-8')):
        f = base64.b64decode(f)
    else:
        print(f.decode('utf-8'))
        break
```

输出结果：

```
..... ..... ..!?! !.?.. ..... ..... ?.?!. ?.... ..... ..... !.!.! .!.
?.... ...!? !!.?! !!!!! ?.?!. ?!!!! !.?.. ..... !?!!. ?.... ..?.? !.?..
..!.! .!.! !.?.. ..... !?!!. ?!!!! !!?.? !.?!! !!!.? ..... ..... .....
（中间省略）
.?!!! !!.?. ..... .!?!! .?... ...?. ?!.?. ...!. !.... ...!. ?.... ...!?
!!.?. ..... ?.?!. ?.... ..... .!.?.
```

明显得到是简化形式的Ook！，运行工具得到：

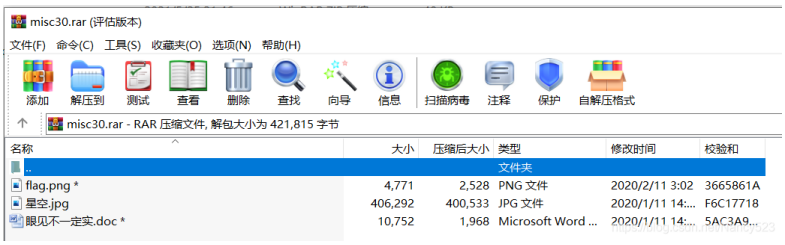
```
+++++ +++++ [->++ +++++ +++<] >++.+ +++++ .<+++ [->-- -<]>- -.+++ +++.<
++++[ ->+++ +<]>+ +++.< +++++ +[->- ----- <]>.< +++[- >+++< ]>+++ ++.++
+++++ .---- ----- .<+++ ++++[ ->--- ----< ]>--. <++++ +++[- >++++ +++<]
>++++ +++++ +++.- ----- --.-- ----. <++++ [->++ ++<]> +++++ .<+++ +++[-
>---- --<]> -.<++ ++[-> ++++< ]>.++ ++.<+ ++[-> ---<] >---- --.<+ +++[-
>++++ <]>++ .---- ---.< +++++ +[->- ----- <]>-- ----- -.<++ +++++ [->++
+++++ <]>++ ++.++ +++++ .++++ ++++. <++++ +++++ [->-- ----- --<]> -----
.<+++ +++++ +[->+ +++++ +++<] >++++ +++++ ++.<
```

明显为brainfuck 运行工具得到：

```
flag{Welc0me_tO_cTf_3how!}
```

5、misc30

下载附件 得到加密的压缩包 内含一个加密的png图片 一个梵高的星空图片 一个doc文件



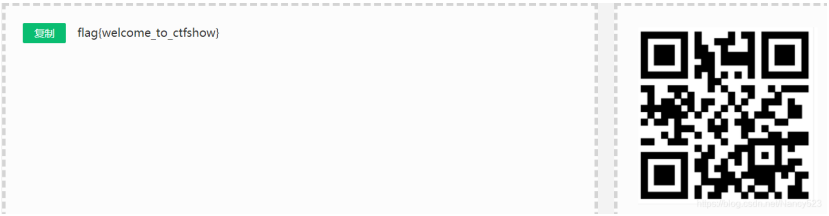
将星空解压 发现没什么特殊的地方 查看图片属性：

发现little stars为doc文件的解压密码 解压 观察得到文本末尾有白色字体 改色 得到

这里什么都没有
里什么都没有
什么都没有
么都没有
都没有
没有
有

你知道梵高的星空吗？
Hello friend!

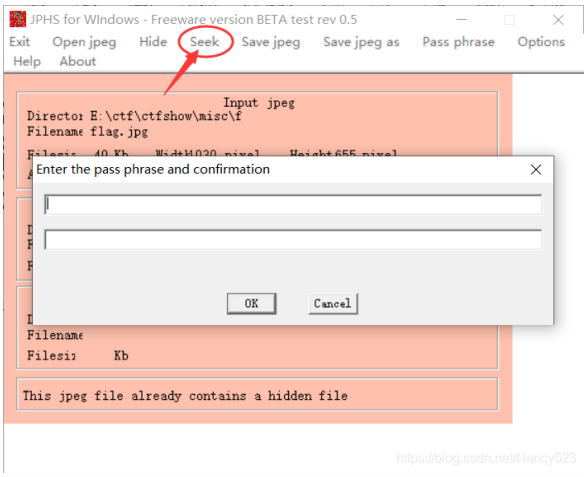
发现Hello friend！为flag.png的解压密码 解压 得到二维码 扫码得到flag

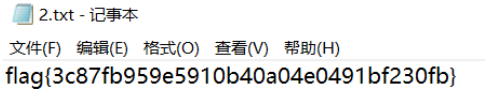


6、stega1

下载附件得到jpg文件 常用图片隐写手段无果、十六进制编辑器没有额外字段 梭不出来 不懂 看wp

得到为JPHS隐写，密码为空





flag{3c87fb959e5910b40a04e0491bf230fb}

7、misc3

密文：

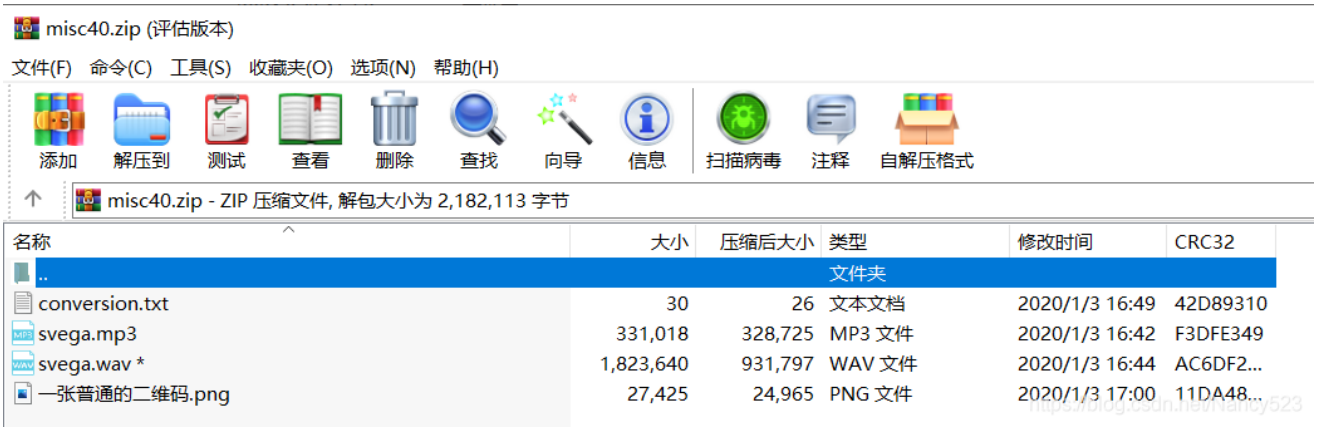
> zse4rfvsdf 6yjmko0

结合提示 猜想是键盘布局 观察得到flag

> flag{av}

8、misc40

下载附件 得到一个txt文件 一个mp3文件 一个png二维码 还有一个加密的wav文件



打开conversion.txt

猜想应该是进制转换 得到

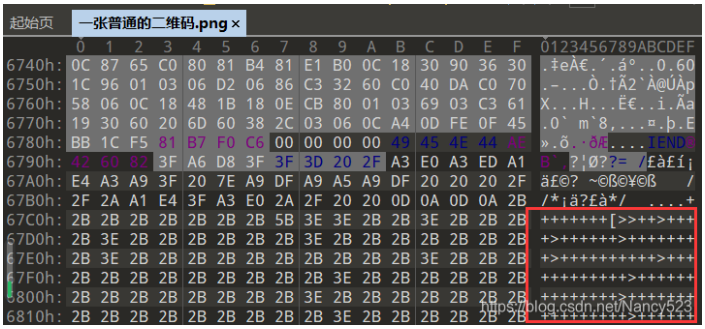
> 202013

不知道干嘛 丢着 看二维码去

扫码得到

> flag不在这里哦~~

丢进十六进制编辑器 梭哈一下 文本末尾有附加数据 明显为brainfuck



在线翻译得到：

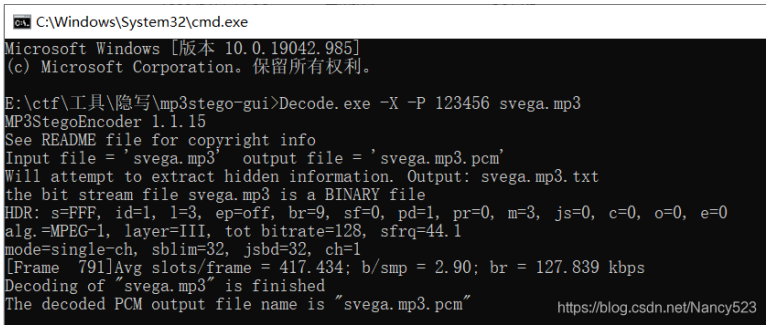
> 和谐民主和谐文明和谐和谐和谐自由和谐平等和谐公正

社会主义核心价值观加密 千千秀字解密得到：

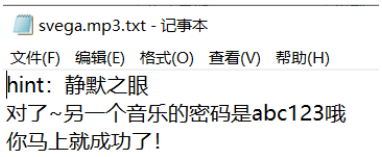
> 123456

还是不知道干嘛，，，丢着先

然后从mp3文件入手，丢进MP3Stego工具 尝试密码后发现密码为123456：

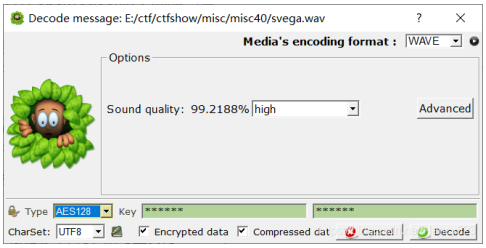


得到隐写结果：

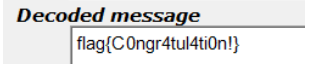


解压svega.wav 根据提示得到 隐写工具为SilentEye

打开后 联想到之前conversion.txt中的数字 勾选Encrypted data选项：

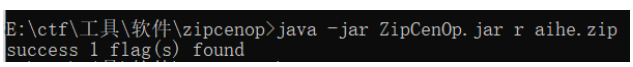


尝试后得到Sound quality应该为high Type为AES128 得到flag



9、misc30

下载附件得到zip文件 丢进十六进制编辑器发现zip伪加密 丢进ZipCenOp梭哈一下



解压得到一个mp3文件 尝试丢进foremost 得到jpg格式图片文件

```
Windows PowerShell
PS E:\ctf\工具\软件\foremost> .\foremost .\aihe.mp3
Processing: .\aihe.mp3
|*|
```

感觉像是下半段图片被截断了 查看图片详细信息

图像 ID	
分辨率	895 x 371
宽度	895 像素
高度	371 像素
水平分辨率	96 dpi
垂直分辨率	96 dpi
位深度	24

> 371的16进制是01 73
> 895的16进制是03 7F

搜索01 73 改成03 7F

发现为猪圈密码 解密后得到：

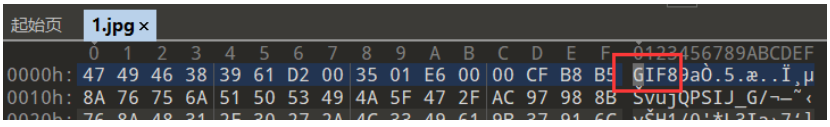
> well done

得到flag{well done}

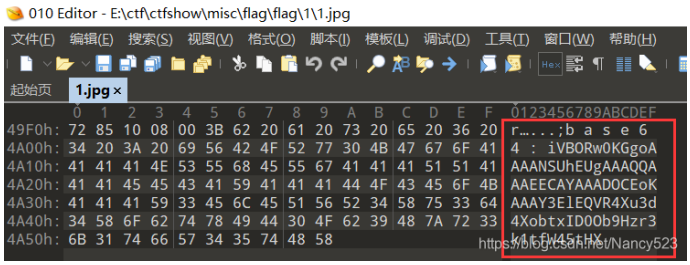
10、红包题第一弹

下载附件解压后得flag文件夹，内含编号从1-86的压缩包 打开后发现应该是由一张gif图拆分而成

随便拆一张丢进十六进制编辑器查看 发现果然是gif格式



且每张图片的末尾均有一段额外字符 根据第一张图的额外字符提示得到应该是base64编码



当时不会写脚本，，，80多段文字手撸下来的（嗯 没错 我是笨比　这里附上8神的脚本：

```
import zipfile
path1 = 'C:/Users/Administrator/Desktop/flag/'
path2 = 'C:/Users/Administrator/Desktop/flag/out/'

res = ''
for i in range(1, 87):
    zip = zipfile.ZipFile(path1 + str(i) + '.zip')
    jpg = str(i) + '.jpg'
    zip.extract(jpg, path2)
    zip.close()

    f = open(path2 + jpg, 'rb').read()
    res += f[len(f)-100:len(f)+1].decode('utf-8')
print(res)
```

在输出结果的开头加上

>

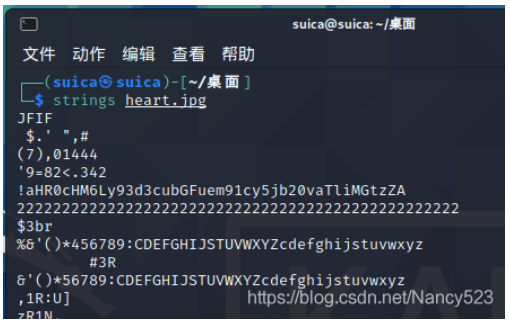
然后得到png格式图片：

扫码得到flag

复制 flag{gif_is_so_easy}

11、stega10

下载附件得到一张jpg格式图片　啥都没得 丢进kali看看



一串奇奇怪怪的东西 试试那段a开头的是不是base64 解码得到

好家伙 访问网址 下载得到flag.zip（我改了个名字

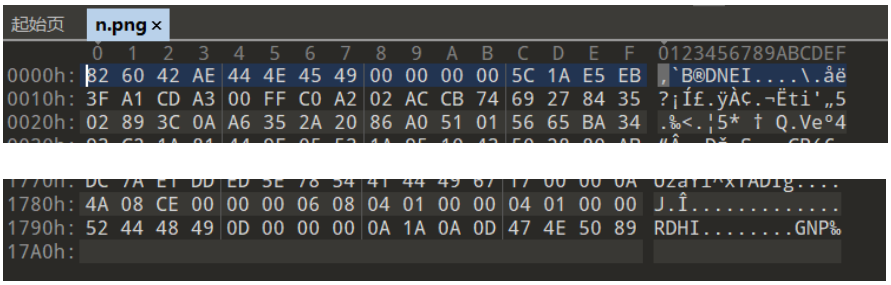
密码文件夹中含有9个加密的txt文件 上脚本 得到文本内容

```
import zipfile
import string
import binascii

crc = []
zip = zipfile.ZipFile('E:/ctf/ctfshow/misc/stega10.zip', 'r')
for i in zip.namelist():
    crc += [zip.getinfo(i).CRC]
crc = crc[1:10]

for i in range(9):
    for j in string.printable:
        c = binascii.crc32(j.encode('utf-8'))
        if c == crc[i]:
            print(j, end = '')
```

输出结果：

> 447^*5#)7

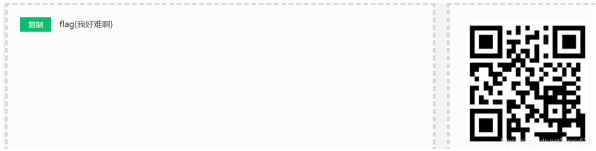
以此为密码 解压n.zip 得到n.png 诶好 打不开 十六进制编辑器打开 观察头尾



发现整个文件的字节被反过来了 倒转后写入文件：

```
f = open('E:/ctf/ctfshow/misc/stega10/n/n.png', 'rb').read()
res = open('E:/ctf/ctfshow/misc/stega10/n/n2.png', 'wb')
res.write(f[::-1])
```

得到一个二维码 扫码得到flag



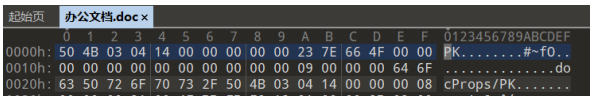
12、stega11

下载得到jpg格式图片 十六进制编辑器打开 发现文件末尾有额外数据而且好像又套了一张jpg文件？ 先解码base32解码得到：

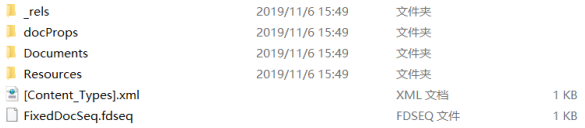
啊这，，直接给了 那就给了呗

13、misc4

下载附件解压后得到doc文件 打不开 报错 十六进制编辑器查看



PK开头，，zip格式压缩包 修改后缀后解压得到



不知道该咋整了，，一个个瞅呗 在 办公文档\Documents\1\Pages\1.txt得到线索：

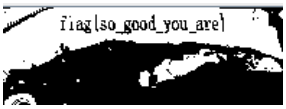
观察 按规律提取字符 得到flag：

> flag{}{xps?Oh,Go0d!}

修改一下，提交

14、misc5

下载附件得到png格式图片，十六进制编辑器查看没有特殊的 考虑图片隐写 stegsolve打开 在Red Plane4处找到flag

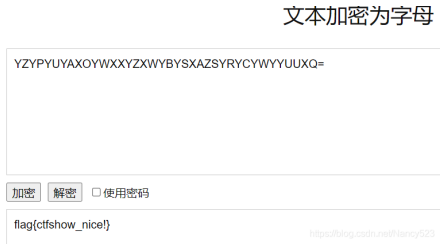


15、misc6

下载附件得到txt格式问价，内容为：

> YZYPYUYAXOYWXXYZXWYBYSXAZSYRYCYWYYUUXQ=

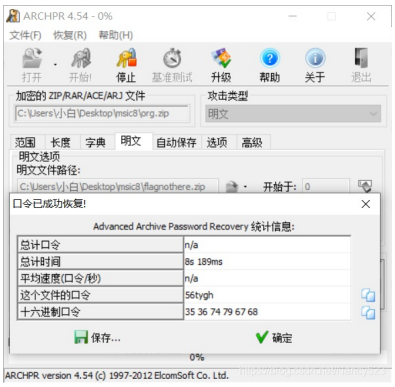
考虑base64或者base32解码 发现解码失败 嗯 遇事不决千千秀字 得到flag



16、misc8

下载附件得到flagnothere.jpg跟org.zip 且压缩包加密

压缩包中也含有flagnothere.jpg 根据之前DASCTF五月赛的忘了哪题的经验 考虑明文破解 嗯 成功了



解压得到flag.png 扫码得到flag

17、stega2

下载附件得到png格式图片 正常打开没啥问题 十六进制编辑器打开也没啥不一样 上linux看看 吼哟 CRC错误 考虑图片高度修改 上脚本爆破

```
import struct
import binascii
from Crypto.Util.number import bytes_to_long

img = open('E:/ctf/ctfshow/misc/stega2/flag.png', 'rb').read()

for i in range(0xFFFF):
    stream = img[12:20] + struct.pack('>i', i) + img[24:29]
    crc = binascii.crc32(stream)
    if crc == bytes_to_long(img[29:33]):
        print(hex(i))
```

> 输出结果:
> 0x1ce

在十六进制编辑器中修改图片高度得到flag

18、stega3

下载附件得到rar格式文件，内含一张图片 解压后 十六进制编辑器查看 没啥特别的 考虑各种常见的图片隐写也莫得 我q****

回头从压缩包入手 用7z打开压缩包 联想到NTFS流隐写

打开a.flag:flag1.txt 得到flag