CCTF重邮（绿盟）杯_部分解密题WriteUp

原创

焚卷觅光 于 2017-07-31 10:43:32 发布 4205 收藏 3

分类专栏： CTF 解密 文章标签： 解密 ctf cctf writeup

本文链接：https://blog.csdn.net/wenliheng0/article/details/76422252

CTF 同时被 2 个专栏收录

5 篇文章 0 订阅

订阅专栏

解密

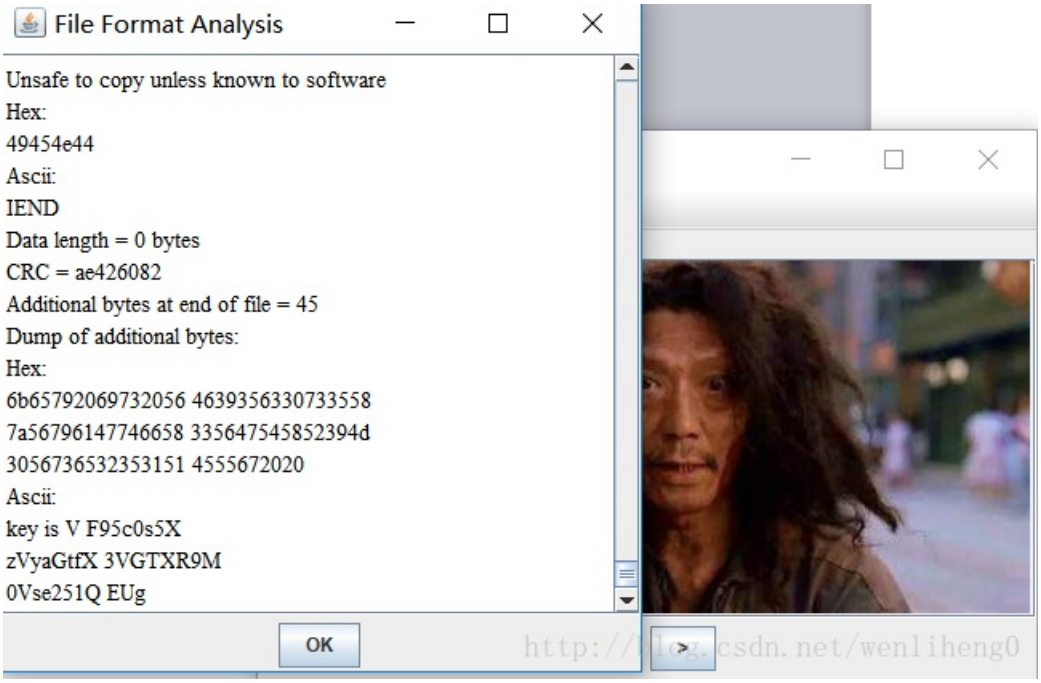
3 篇文章 0 订阅

订阅专栏

CCTF重邮（绿盟）杯_部分解密题

0X00 Kungfu

解开压缩包，里面就是一个图片（又是那个卖如来神掌的老乞丐），按照脑洞惯例跑Stegsolve。



在文件底发现了“key is .....”把它复制出来再说。

```
VF95c0s5XzVyaGtfX3VGTXR9M0Vse251QEUg
```

明显是一个base64加密，放解码器跑出来是这个：

```
T_ysK9_5rhk__uFMt}3El{nu@E
```

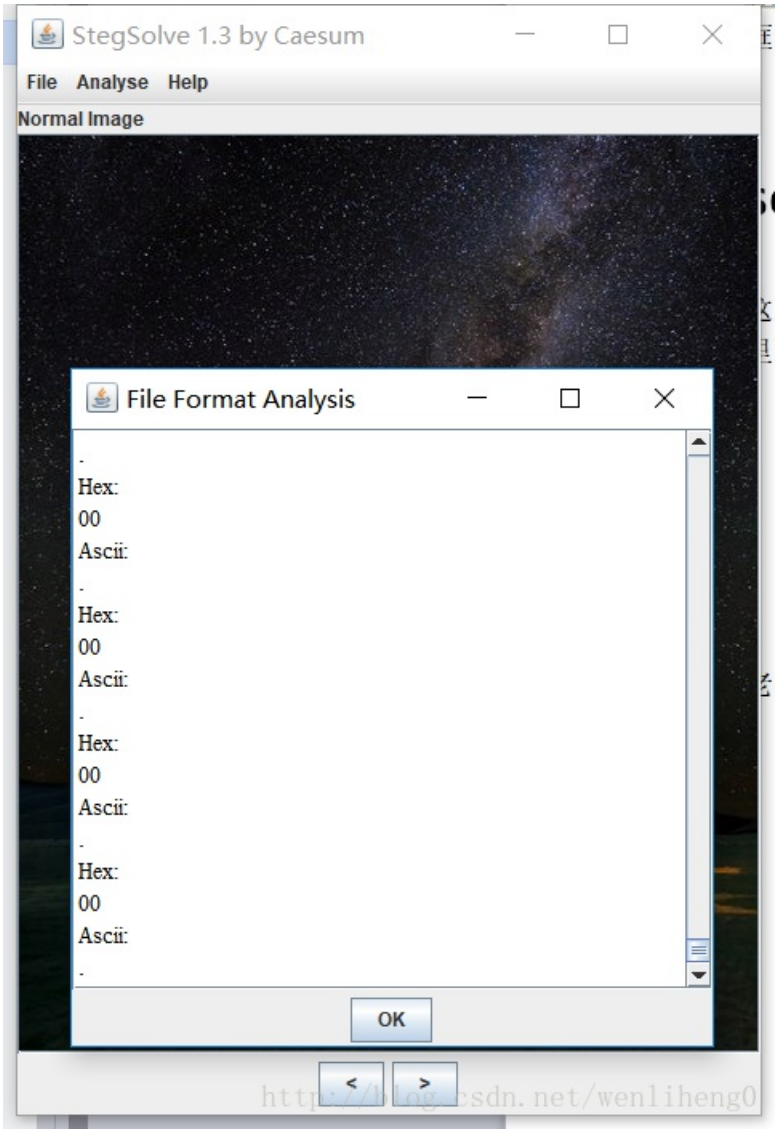
这已经很接近flag的格式了，只要再换一下字符的顺序就好了，试一下栏栅加密。

框中那个去掉@就是这个题目的flag了。

0X01 so_easy

这是解密签到题了，的确so easy(lll￣ω￣)。下载文件是一个压缩包（都没加密），解压后里面是一个lsb.bmp图片。

老套路，Stegsolve打开。



文件底部并没有藏东东，下面就看看色阶。

在Red的0阶可以看到隐藏的flag。

0X02 你知道吗？这是什么

下载下来是一个名叫`hidden.png`的文件，打开是白板一片，什么都没有，在`Stegsolve`里面也没有发现有价值的信息。猜测是文件包含类的题目，把文件放到`binwalk`里面看一下文件结构，发现的确包含了一个zip压缩包。

```
root@kali2017:~/图片# binwalk hidden

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 351 x 560, 8-bit/color RGBA, non-interl
aced
41            0x29            Zlib compressed data, default compression
1975          0x7B7           Zip archive data, at least v1.0 to extract, compre
ssed size: 291742, uncompressed size: 291742, name: hidden2.jpg
293851        0x47BDB         End of Zip archive
```

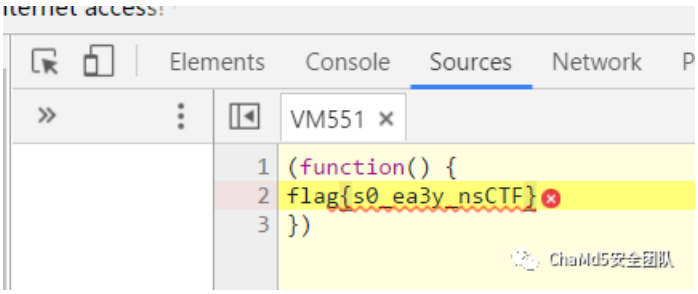
用binwalk将它提取出来：`binwalk -e hidden.png`
在提取的文件里面就有一个hidden2.jpg文件，flag就在图片上。

0X03 表情包

这道题是颜文字加密，其实就是一段js代码，以前做过这个题，直接在浏览器的控制台跑一下就可以了，但是这次在浏览器跑了却报错。
前前后后想了很久，最后没有办法只有把网站上的例子拿来实验看，发现这个颜文字加密的前面都是一样的。再回过头来看题目，发现有部分“表情”不对。

```
把//<em>´▽｀</em>/改为//*´▽｀*/
把<em>改为_
把</em>改为_
把'\'改为'\\'
```

解密站点
这里虽然报错了，但是你可以点那个VM，里边就有他解释的信息。

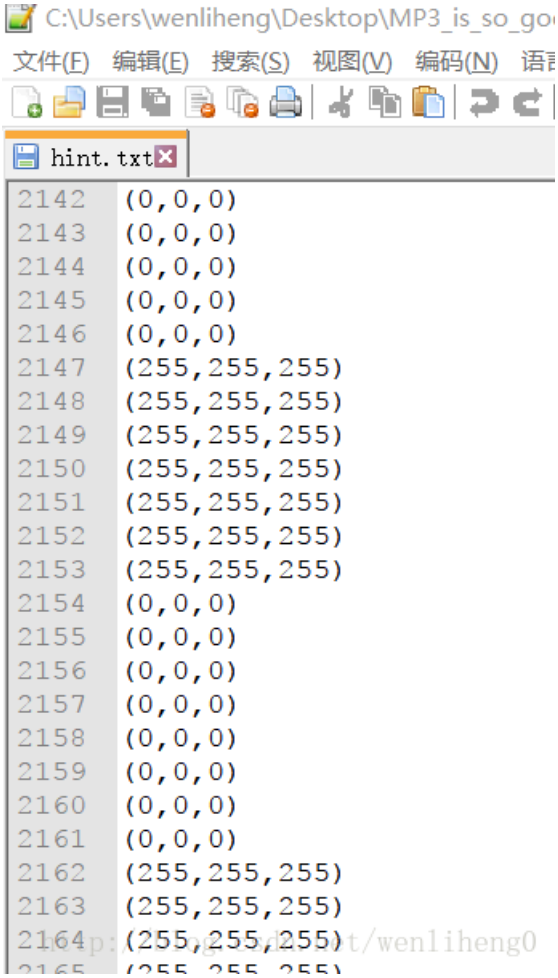


（PS：由于做的时候没有截图，之后官方把站点关了，没有保存题目，在这里就借用以下P猫的图片。）

```
以上两张图片感谢ChaMd5团队pcat大佬提供。
```

0X04 MP3_is_so_good

这是一个典型的MP3加密题，在压缩包中有mp3文件和密码文件。mp3解密是需要密码的，但是这次的密码他没有明确的给出。在hint.txt中是这样的：



这个东西很明显是色标号，由黑和白组成，估计就是二维码。写个脚本把它还原出来看看。

```
#将原文件中的（255，255，255）转成0，（0，0，0）转成1
import re

file = open('hint.txt')
f = open('new.txt','a+')

for i in range(78400):
    line = file.readline()#读取一行
    if line[1:4] == '255':
        f.write("0\n")
    else:
        f.write('1\n')
f.close()
file.close()
```

```python
#-*- coding:utf-8 -*-
from PIL import Image
import re

x = 280 #x坐标  通过对txt里的行数进行整数分解
y = 280 #y坐标  x*y = 行数
im = Image.new("RGB",(x,y))#创建图片
file = open('new.txt') #打开rbg值文件

#通过一个个rgb点生成图片

for i in range(0,x):
    for j in range(0,y):
        line = file.readline()#获取一行
        if int(line) == 0:
            im.putpixel((i,j),(255,255,255))#rgb转化为像素
        else:
            im.putpixel((i,j),(0,0,0))#rgb转化为像素
im.show()
```

PS：python不熟，dalao们别笑，(✿◡‿◡)

里面提示了MP3stego的密码在1000到1300之间，然后就是批量爆破密码了，最后得到密码是1067

```
E:\Tools\隐写\音频隐写\MP3Stego>Decode.exe -X -P 1067 mp3.mp3
MP3StegoEncoder 1.1.17
See README file for copyright info
Input file = 'mp3.mp3'  output file = 'mp3.mp3.pcm'
Will attempt to extract hidden information. Output: mp3.mp3.txt
the bit stream file mp3.mp3 is a BINARY file
HDR: s=FFF, id=1, l=3, ep=off, br=9, sf=0, pd=1, pr=0, m=0, js=0, c=0, o=0, e=0
alg.=MPEG-1, layer=III, tot bitrate=128, sfrq=44.1
mode=stereo, sblim=32, jsbd=32, ch=2
[Frame  202]Avg slots/frame = 415.909; b/smp = 2.89; br = 127.372 kbps
Decoding of "mp3.mp3" is finished
The decoded PCM output file name is "mp3.mp3.pcm"
```


最后拿到flag：`flag{Brute_f0Rce_iS_W0nderful}`