CTF 杂项---图片隐写

原创

一依儿 于 2021-03-26 22:16:04 发布 1517 收藏 19

一、图片隐写的常见隐写方法

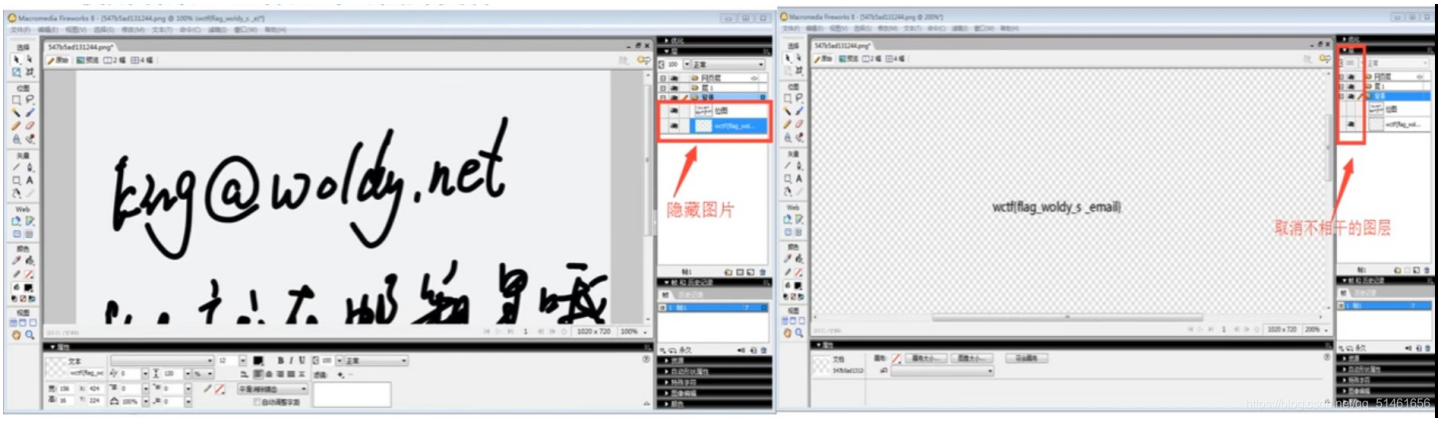
1. 细微的颜色差别
2. GIF图多帧隐藏
1.颜色通道隐藏
2. 不同帧图信息隐藏
3. 不同帧对比隐写
3. Exif信息隐藏
4. 图片修复
1.图片头修复
2. 图片尾修复
3.CRC校验修复
4. 长、宽、高度修复
5.最低有效位LSB隐写
6.图片加密
1 Stegdetect
2. outguess
3. Jphide
4 .F5

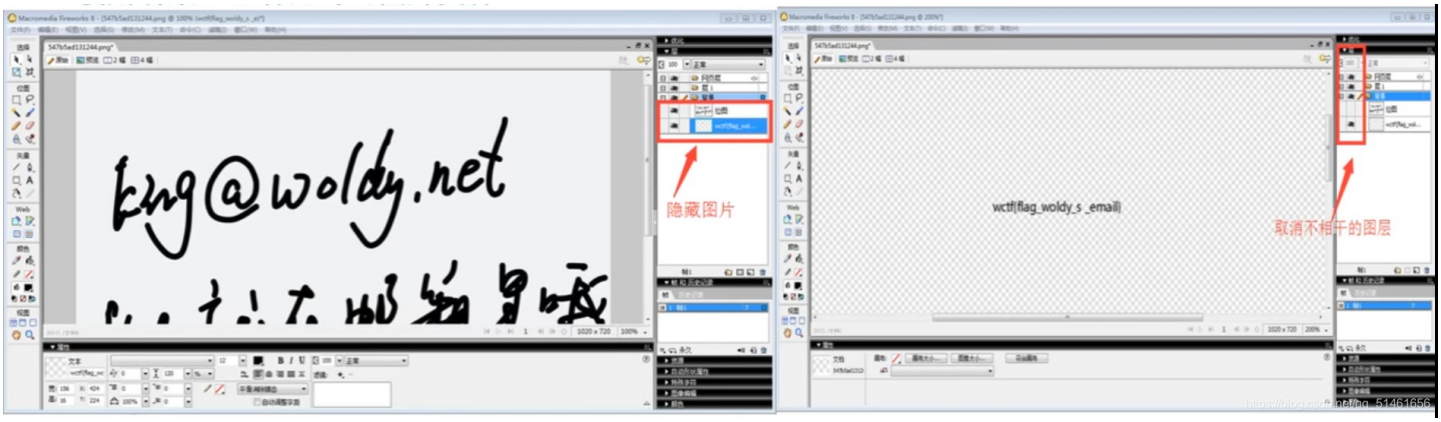
二、图片隐写

1.Firework

使用winhex打开文件时会看到文件头部中包含firework的标识，通过firework可以找到隐藏图片。

使用场景：查看隐写的图片文件

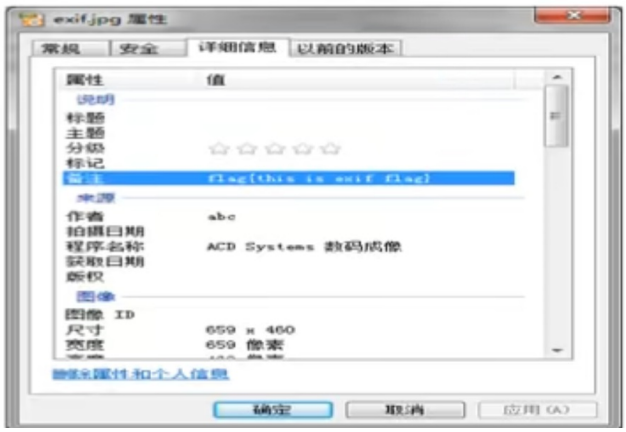


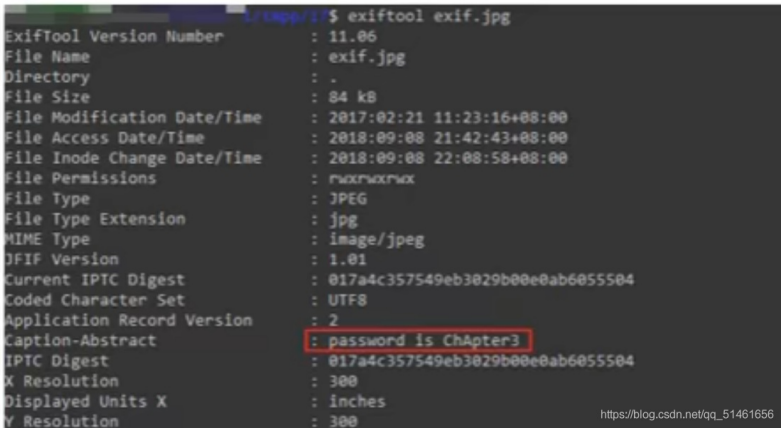


2.Exif

Exif按照JPEG的规格在JPEG中插入一些图像/数字相机的信息数据以及缩略图像. 可以通过与JPEG兼容的互联网浏览器/图片浏览器/图像处理等一些软件来查看Exif格式的图像文件.就跟浏览通常的JPEG图像文件一样。

.图片右键属性，查看exif或查看详细信息，在相关选项卡中查找flag信息。





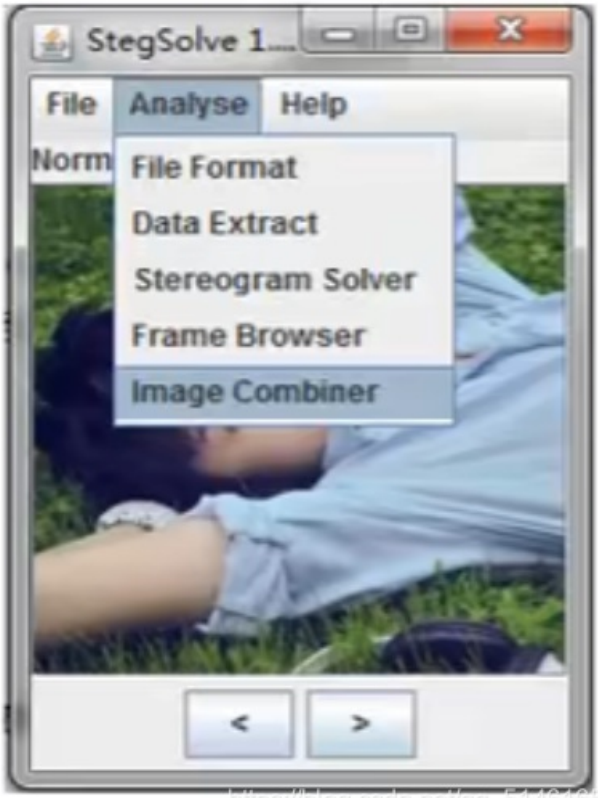
3.Stegsolve

当两张jpg图片外观、大小、像素都基本相同时，可以考虑进行结合分析，即将两个文件的像素RGB值进行XOR、ADD、SUB等操作，看能否得到有用的信息，StegSolve可以方便的进行这些操作。

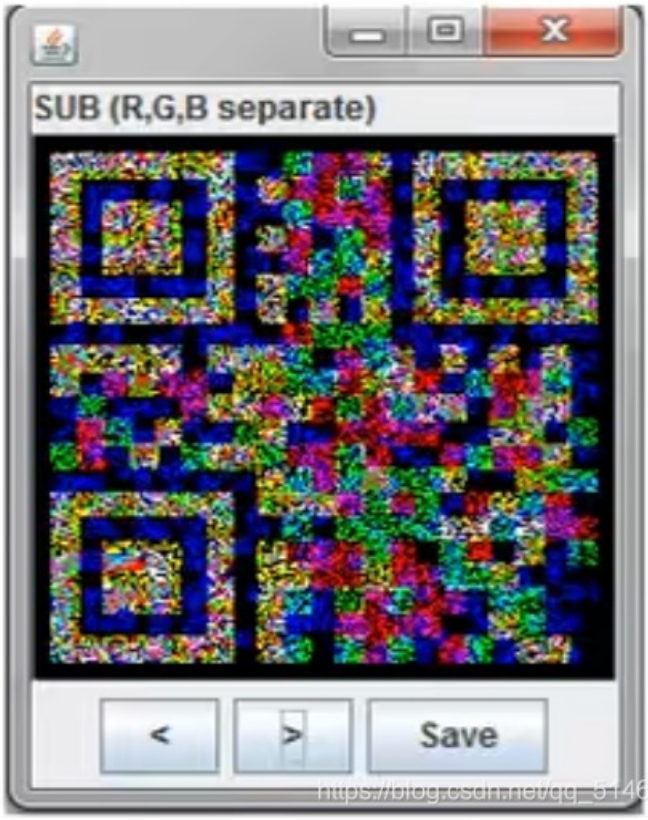
使用场景：两张图片信息基本相同

1.打开第一张图片，点击

analyse->Image combiner



2.在弹出的窗口中点击左右按钮
选择处理方式，点击save保存有价
口 值的结果。



4.LSB(最低有效位Least Significant Bit)

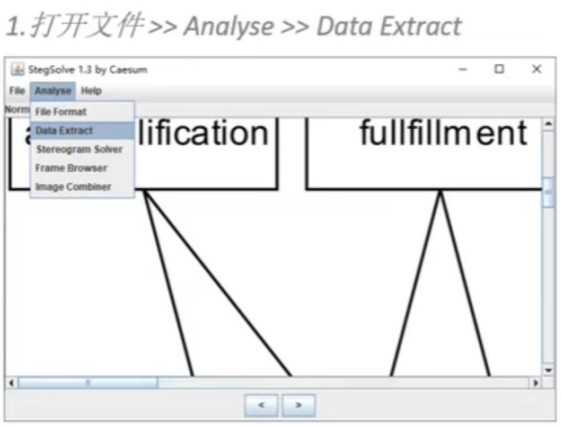
LSB替换隐写基本思想是用嵌入的秘密信息取代载体图像的最低比特位，原来的的7个高位平面与替代秘密信息的最低位平面组合成含隐藏信息的新图形。

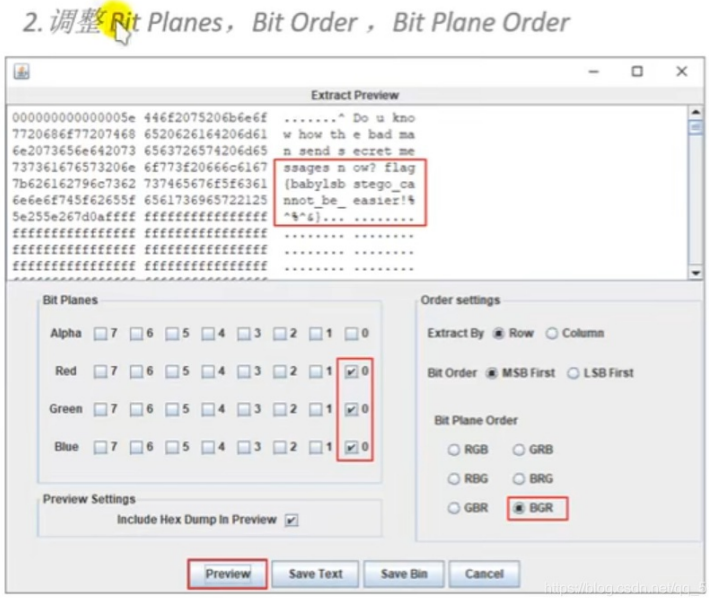
1.像素三原色(RGB)

2. 通过修改像素中最低位的1bit来达到隐藏的效果

3. 工具：stegsolve、zsteg、wbstego4、python脚本

（1）、stegsolve.jar 工具





（2）、zsteg 工具

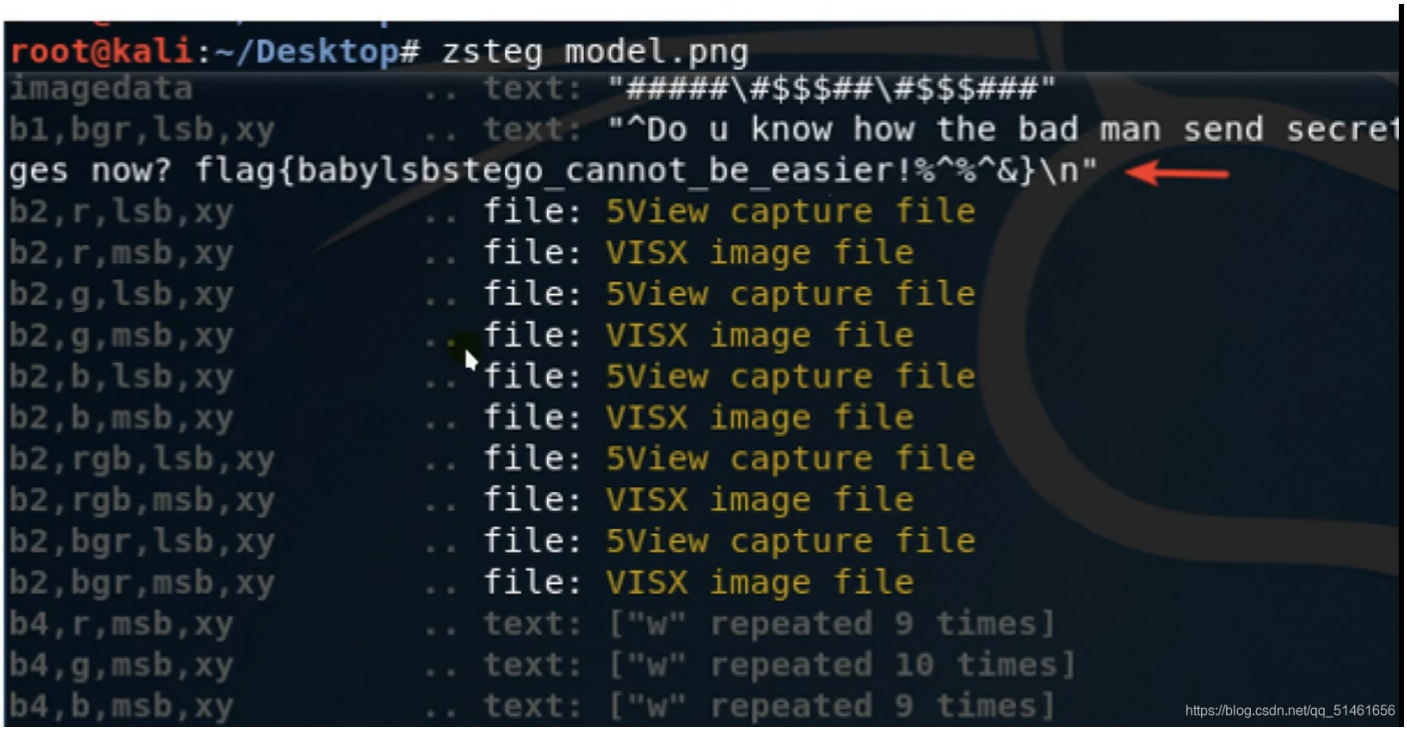
detect stegano-hidden data in PNG & BMP

Installation

```
gem install zsteg
```

检测LSB隐写

zsteg xxx.png



（3）、wbstego4 工具

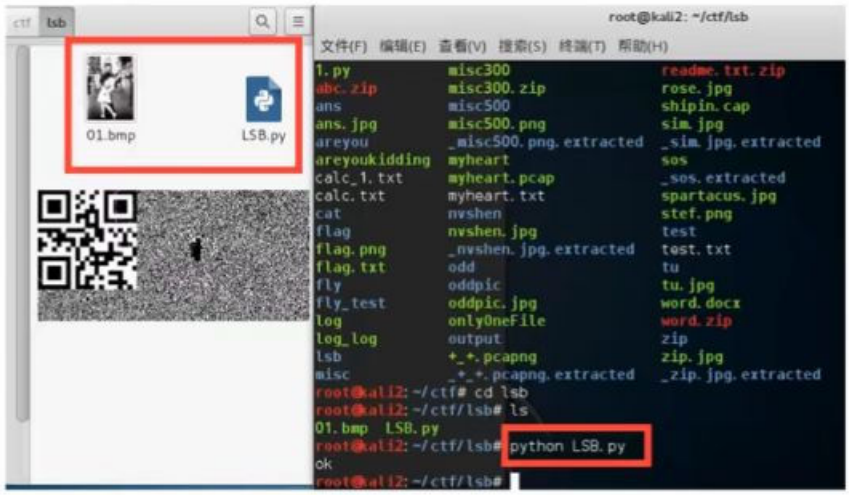
解密通过sb加密的图片（按照提示来就行）

（4）python脚本来处理

将以下脚本放在kali中运行，将目标文件放在脚本同目录下，将脚本中的文件名修改为文件名，运行python即可

```
LSB.py
#coding:utf-8
import PIL.Image
def foo():
    im = PIL.Image.open('01.bmp')
    im2 = im.copy()
    pix = im2.load()
    width,height=im2.size

    for x in xrange(0,width):
        for y in xrange(0,height):
            if pix[x,y]&0x1 == 0:
                pix[x,y]=0
            else:
                pix[x,y]=255
    im2.show()
    pass
if __name__ == '__main__':
    foo()
    print 'ok.'
    pass
```

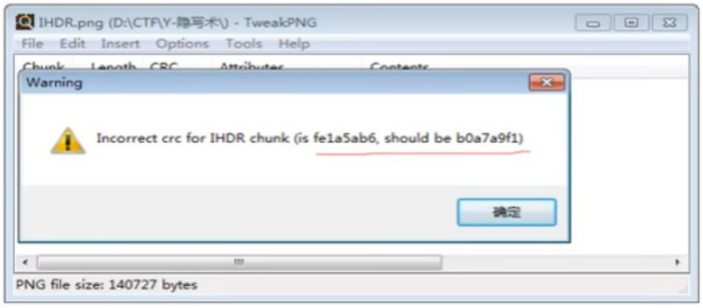


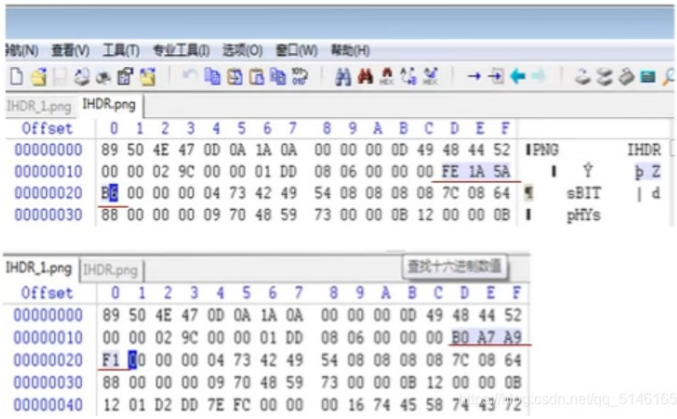
5.TweakPNG

TweakPNG是一款简单易用的PNG图像浏览工具，它允许查看和修改一些PNG图像文件的元信息存储。使用场景：文件头正常却无法打开文件，利用TweakPNG修改CRC

例：

1.当PNG文件头正常但无法打开文件，可能是CRC校验出错，可以尝试通过TweakPNG打开PNG，会弹出校验错误的提示，这里显示CRC是 fe1a5ab6，正确的是 b0a7a9f1。打开winhex搜索 fe1a5ab6 将其改为 b0a7a9f1。

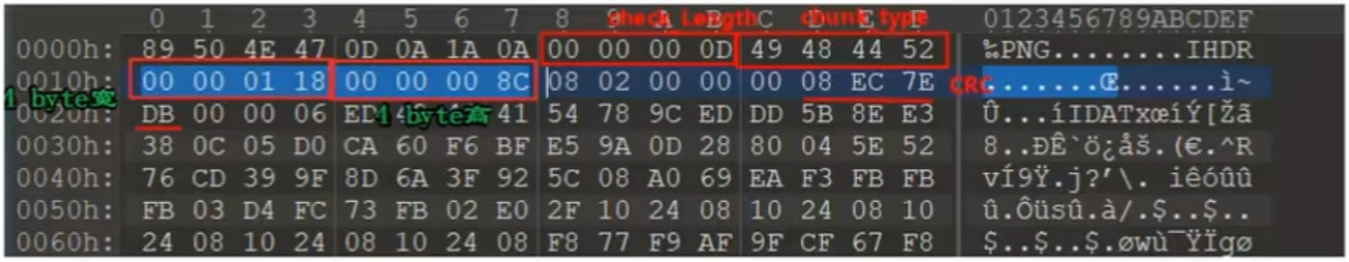




但是…

文件头正常却无法打开文件，利用TweakPNG修改CRC…

有时CRC没有错误，但是图片的高度或者宽度发生了错误，需要通过CRC计算出正确的高度或者宽度。




```
#coding:utf-8
import binascii
import struct
crcbp = open("xxx.png","rb").read()  #此处填上文件名
for i in range(1024):
   for j in range(1024):
      data = crcbp[12:16] + struct.pack('>i',i) + struct.pack('>i',j) + crcbp[24:29]
      crc32 = binascii.crc32(data) & 0xffffffff
      if crc32 == 0x08ec7edb:  #此处填上CRC值
         print i,j
         print "hex",hex(i),hex(j)
```

6.Bftools

bftools用于解密图片信息。

使用场景：在windows的cmd下，对加密过的图片文件进行解密格式：

Bftools.exe decode braincopter` 要解密的图片名称-output 输出文件名Bftools.exe run
上一步输出的文件

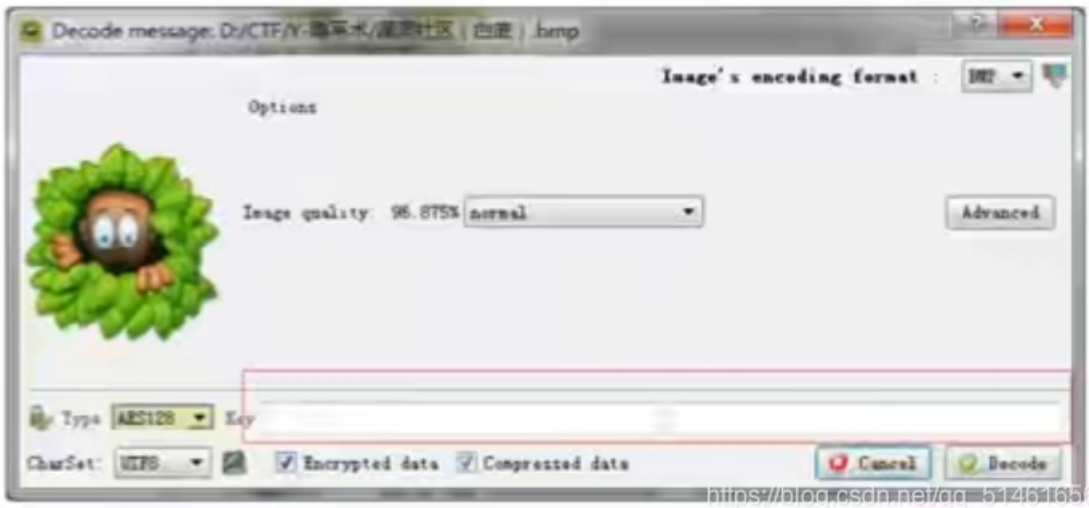
```
D:\CTF\bftools\bftools>bftools.exe decode braincopter zzzzzzyu.png --output 123.
png

D:\CTF\bftools\bftools>bftools.exe run 123.png
XDCTF{ji910-dad9jq0-iopuno}      ¤▶§♫¤↔
D:\CTF\bftools\bftools>
```

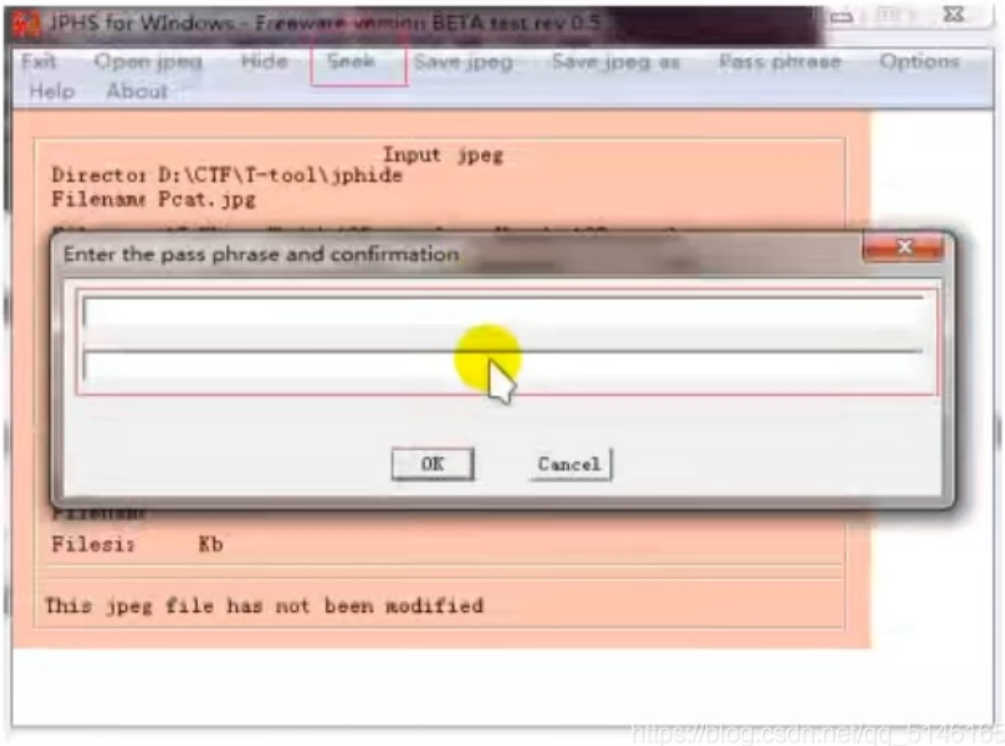

7.SilentEye

silenteye是一款可以将文字或者文件隐藏到图片的解密工具。
使用场景：windows下打开silentEye工具，对加密的图片进行解密
例：
1.使用silentEye程序打开目标图片，点击image->decode，点击decode，可以查看隐藏文件，点击保存即可

2、如果需要密码，勾选encrypted data ，输入密码和确认密码，点击decode再解密



8.JPG图像加密



1）

Stegdetect 工具探测加密方式

Stegdetect程序主要用于分析JPEG文件。因此用stegdetect可以检测到通过`Steg、JPHide、OutGuess、Invisible Secrets、F5、appendx和Camouflage`这些隐写工具隐藏的信息。

stegdetect xxx.jpg

stegdetect -s 敏感度xxx.jpgexi

```
thinking@ubuntu:~/Desktop$ stegdetect 123456.jpg
123456.jpg : f5(***)
thinking@ubuntu:~/Desktop$ stegdetect angrybird.jpg
angrybird.jpg : outguess(old)(*)
thinking@ubuntu:~/Desktop$ stegdetect Pcat.jpg
Pcat.jpg : negative
thinking@ubuntu:~/Desktop$ stegdetect -s 10.0 Pcat.jpg
Pcat.jpg : jphide(*)
thinking@ubuntu:~/Desktop$
```

2)Jphide

Jphide是基于最低有效位LSB的JPEG格式图像隐写算法.

例：

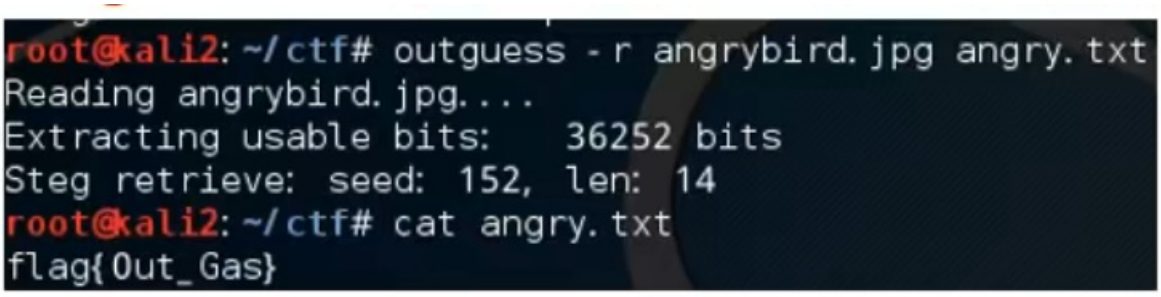
Stegdetect提示jphide加密时，可以用Jphs工具进行解密，打开jphswin.exe，使用open jpeg打开图片，点击seek，输入密码和确认密码，在弹出文件框中选择要保存的解密文件位置即可，结果保存成txt文件。

Outguess

outguess一般用于解密文件信息。

使用场景：Stegdetect识别出来或者题目提示是outguess加密的图片该工具需编译使用：**./configure & make & make install**

格式：`outguess-r要解密的文件名输出结果文件名`



F5

F5一般用于解密文件信息。

使用场景：Stegdetect识别出来是F5加密的图片或题目提示是F5加密的图片

进入F5-steganography_F5目录，将图片文件拷贝至该目录下，从CMD进入该目录

格式：`Java Exrtact要解密的文件名-p密码 运行结束后我们可以直接在目录下的output.txt中看到结果`

9.—维码处理

1.使用二维码扫描工具CQR.exe打开图片，找到内容字段

2.如果二维码某个定位角被覆盖了，该工具有时候也可以自动识别，如果识别失败，需要使用PS或画图工具将另外几个角的定位符移动到相应的位置，补全二维码。