图片隐写相关

原创

枯叶难辉 于 2018-12-14 13:30:29 发布 639 收藏 4

分类专栏： 测试性

本文链接：https://blog.csdn.net/qq_43610974/article/details/84999448

前言

图片隐写总的来讲，就是对图片进行一些加密，将自己所要隐藏的信息进行隐藏，而隐藏的方式多重多样，破解的方式也多种多样。

话不多说，开始正题。

一般来讲，我们在拿到图片后都会看到一个后缀名，大部分人在看到后缀名后就会下意识的认为这个就是图片本身的后缀，但有些时候并不是。所以我们在这时需要对图片头部进行分析，精确的确认图片到底是什么图片类型。以下是各个头文件类型：

- **1.JPEG**
  - 文件头标识 (2 bytes): 0xff, 0xd8 (SOI) (JPEG 文件标识)
  - 文件结束标识 (2 bytes): 0xff, 0xd9 (EOI)

2.TGA
- 未压缩的前5字节　00 00 02 00 00
- RLE压缩的前5字节　00 00 10 00 00

3.PNG
- 文件头标识 (8 bytes)　89 50 4E 47 0D 0A 1A 0A

4.GIF
- 文件头标识 (6 bytes)　47 49 46 38 39(37) 61
  G I F 8 9 (7) a

- **5.BMP**
  - 文件头标识 (2 bytes)　42 4D
    B M

6.PCX
- 文件头标识 (1 bytes)　0A

7.TIFF
- 文件头标识 (2 bytes)　4D 4D 或 49 49

8.ICO

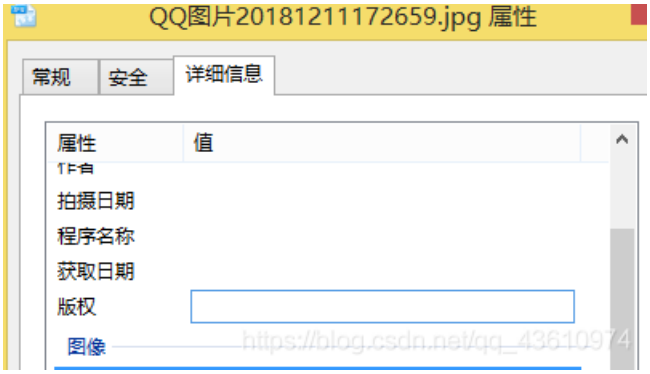
- 文件头标识 (8 bytes) 00 00 01 00 01 00 20 20

- **9.CUR**
  - 文件头标识 (8 bytes) 00 00 02 00 01 00 20 20

10.IFF
- 文件头标识 (4 bytes) 46 4F 52 4D
  F O R M

11.ANI
- 文件头标识 (4 bytes) 52 49 46 46
  R I F F

而你自己想要查看图片的头文件的话，可以尝试用binwalk查看，而且还可以找出是否有其他图片在该文件下。
首先最简单的一种隐写方式是在图片的



这些区域编辑要隐藏的文本，但很容易识别的。不过都是在jpeg图片中加入，具体原因下方会讲。
而隐写有个最常用的方式是将两个不同文件合并。首先我们需要制作一个 .zip随后准备一张图片，然后用指令copy /b .jpg+ .zip output.jpg来进行合并，随后就能得到一张图片。而这个图片可以用binwalk进行查看

```
root@leaves:~/桌面# binwalk whale1.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
103315        0x19393         Zip archive data, at least v2.0 to extract, compre
ssed size: 25, uncompressed size: 23, name: flag.txt
103468        0x1942C         End of Zip archive
```

而这种方式，你可以用foremost将图片解开，并且在foremost后面推荐加一对参数，-o 123
-o是将解出的文件扔如文件夹，而123是文件夹名字
其次就是对文本进行编辑，在头部或者尾部，甚至中间插入编码。所以这时可以用ue打开图片，进行查找。有时候是base64等编码格式，在看的时候你可以先找找奇怪的编码，进行解密。
jpeg图片隐写
jpeg分为压缩数据和标记码两部分。如果在标记码中间加入数据，不会影响文件的打开，标记码如下。

SOI : FF D8
DQT : FF DB
DRI : FF DD
SOF0 : FF C0 这是图像真正开始的地方
DHT : FF C4 huffman表
SOS : FF DA
.....
EOI : FF D9 文件结束标志

jpeg以0xff 0xd9为结束 而由于jpeg具备exif文件描述信息的特殊性，所以附带的属性多得很，我们也可以进行编辑。
我们可以用exiftool可以查看jpeg图片的exif。

png图片

Png图片会将图片源码无损压缩入idat块中进行储存，这时我们可以使用pngcheck查看png的idat模块。每个idat块长度为65524如果我们在查找时，一个块没有填满就开始填下一个块，我们这时候可以用winhe或者ue查找出错的idat块。
然后可以用python编写一个脚本，由于png是用的zlib的压缩编码方式，所以用zlib解出，然后解出一堆0101。我们会想到二进制，不过这个也可能是二维码，可以看看0101总计有多少字。如果是个正方形块，可以用python进行编辑，解出一个二维码。
ihdr：而有时我们拿到一张png图片，却没法用上述方法解出，那么我们可以用winhex等看图片的编码。通过明文查看ihdr标记头，我们在后面可以改参数，改图片的长和宽，最后发现图片拉长后解出了隐写的内容。

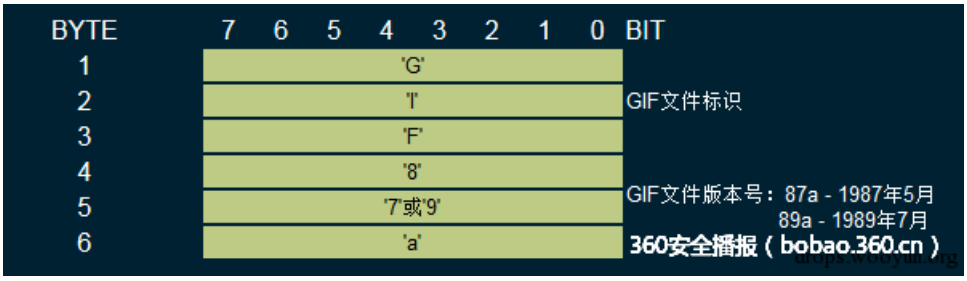
lsb隐写

有种常见的隐写为lsb隐写，lsb也就是最低有效位 (Least Significant Bit)，一般用于bmp图片。
原理就是图片中的像数一般是由三种颜色组成，即三原色，由这三种原色可以组成其他各种颜色，例如在PNG图片的储存中，每个颜色会有 8bit，LSB隐写就是修改了像数中的最低的1bit，在人眼看来是看不出来区别的，也把信息隐藏起来了。譬如我们想把'A'隐藏进来的话，如下图，就 可以把A转成16进制的0x61再转成二进制的01100001，再修改为红色通道的最低位为这些二进制串。
而要解这种方式可以用stegsolve，图片放入后，翻找会儿可以看到一个二维码。（懒得翻文件了，所以，没图）
有时候我们也会遇到文字替换
这时打开文件编码后，会发现bm开头，这时候我们用winhex打开图片，

BMP文件头结构体定义如下:

```
typedef struct tagBITMAPFILEHEADER
{
UINT16 bfType;      //2Bytes，必须为"BM"，即0x424D 才是Windows位图文件
DWORD bfSize;       //4Bytes，整个BMP文件的大小
UINT16 bfReserved1; //2Bytes，保留，为0
UINT16 bfReserved2; //2Bytes，保留，为0
DWORD bfOffBits;    //4Bytes，文件起始位置到图像像素数据的字节偏移量
} BITMAPFILEHEADER;
```

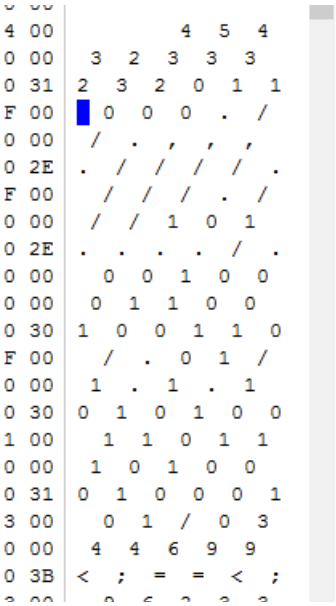
这时我们在文件第10-13位可以找到数据位置。可以用python编译个脚本，找到位置后，奇数1，偶数0
其他类型
首先是gif



gif有时无法打开，而用winhex会发现gif8的头被删除了部分，所以我们需要修复，用winhex等恢复即可。
而有时候gif跳转较快，导致我们没法捕捉，于是可以用2345看图王等软件进行破解。

还有类型是图片锐化，导致我们难以识别图片上的信息，我们可以用ps，美图秀秀等文件进行处理，

还有一种储存了两张图片的，这时候用froemost可以分解出两张一模一样的图，我们可以用compare进行比较，这时候可以看出俩图片的差异所在位置。然后用hex等看可以看出一些不对的地方，注，在这之前我把图片合成了下。用stegsolve。



```
3 00    9  6  2  3  3
0 00   1  0  1  1  0
0 31  1  1  0  0  1  1
0 00    1  1  1  0  0
0 00   1  1  0  0  0
0 31  1  0  1  0  1  1
0 00    0  1  1  1  0
0 00   0  1  0  1  1
0 30  0  1  1  1  0  0
0 00    0  0  0  1  0
0 00   0  1  0  0  0
0 31  1  0  0  0  0  1
0 00    1  0  1  0  0
0 00   0  0  0  1  0
0 00
```

然后可以用脚本解决。

```
#coding:utf-8
import PIL.Image
import random
img1 = PIL.Image.open("00000000.png")
im1 = img1.load()
img2 = PIL.Image.open("00003754.png")
im2 = img2.load()
a=0
i=0
s=''
for x in range(img1.size[0]):
    for y in range(img1.size[1]):
        if(im1[x,y]!=im2[x,y]):
            print im1[x,y],im2[x,y]
            #print im2[x,y][0]
            if i == 8:
                s=s+chr(a)
                a=0
                i=0
            a=im2[x,y][0]+a*2
            i=i+1
s=s+'}'
print s
```


最后推荐下隐写工具。

- stegosolve 神器，能解决好多图片隐藏信息。
- Binwalk 帮助我们识别文件结构。
- Foremost 将多个合并文件进行分离。
- Stegdetect 利用jpeg压缩原理，检查DCT(离散余弦变换)频率系数，通过检测异常判定是否隐写了内容，非常好用。
- Exiftool 读取JPEG图片中的exif信息。
- Pngcheck 分析png图片中的压缩数据信息。
- Python Image库 读取图片像素，分析图片信息的好帮手
- Ulead GIF Animator 动图编辑器，分块识别动图
- 2345看图王 超强的图片查看工具，支持各类图片
- Notepad++ 可查看图片源代码
- 美图秀秀 除了美化图片外，还有很多方便功能
- Photoshop 真正图片编辑工具
- Matlab 对图片锐化、分离水印、正弦余弦变化等功能