隐写术

原创

谢公子 于 2018-11-19 15:44:59 发布 9519 收藏 48
分类专栏： CTF

本文链接：https://blog.csdn.net/qq_36119192/article/details/84232463

CTF 专栏收录该内容
5 篇文章 17 订阅
订阅专栏

目录

隐写(信息隐藏，steganography)

目的：以表面正常的数字载体如静止图象、数字音频和视频信号等作为掩护，在其中隐藏秘密信息。额外数据的嵌入既不改变载体信号的视、听觉效果，也不改变计算机文件的大小和格式（包括文件头），使隐蔽信息能以不为人知的方式进行传输

隐写分析：对多媒体信号进行统计分析，判断其中是否含有隐蔽信息

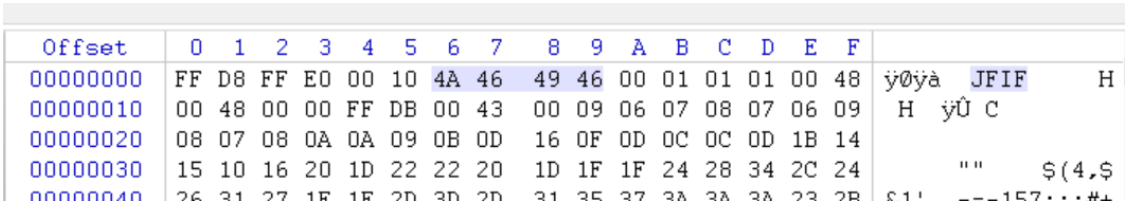
而隐写领域用的最多的就是对将要隐写的信息以二进制形式写入图片中，如：

```
copy /b 1.jpg+1.txt 2.jpg       #将1.txt以二进制形式写入到1.jpg的末尾，形成新文件2.jpg
```

一些图片格式16进制常见的头部和尾部(winhex打开查看)

jpg格式

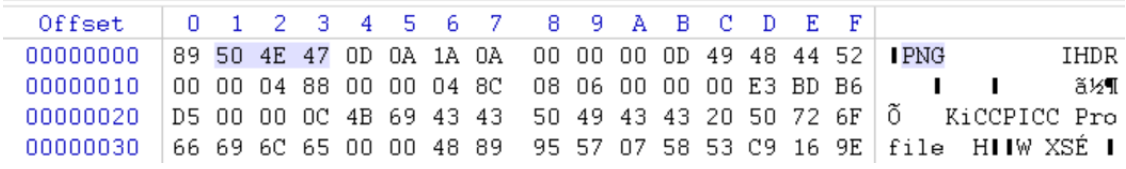
- 头部: 4A 46 49 46　JFIF



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	FF	D8	FF	E0	00	10	4A	46	49	46	00	01	01	01	00	48	ÿØÿà JFIF H
00000010	00	48	00	00	FF	DB	00	43	00	09	06	07	08	07	06	09	H ÿÛ C
00000020	08	07	08	0A	0A	09	0B	0D	16	0F	0D	0C	0C	0D	1B	14	
00000030	15	10	16	20	1D	22	22	20	1D	1F	1F	24	28	34	2C	24	"" $(4,$

png格式

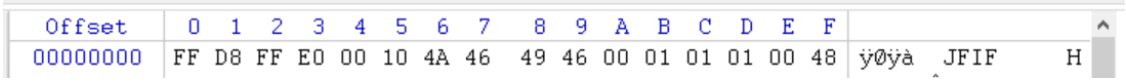
- 头部：50 4E 47　PNG



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	‰PNG IHDR
00000010	00	00	04	88	00	00	04	8C	08	06	00	00	00	E3	BD	B6	ˆ Œ ã½¶
00000020	D5	00	00	0C	4B	69	43	43	50	49	43	43	20	50	72	6F	Õ KiCCPICC Pro
00000030	66	69	6C	65	00	00	48	89	95	57	07	58	53	C9	16	9E	file H‰•W XSÉ ž

jfif格式

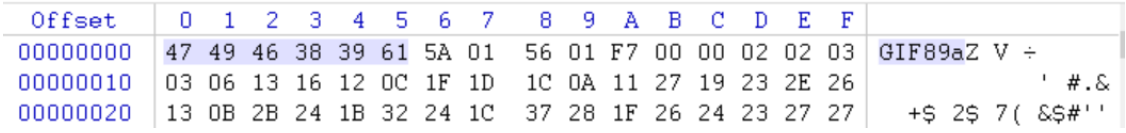
头部：JFIF



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	FF	D8	FF	E0	00	10	4A	46	49	46	00	01	01	01	00	48	ÿØÿà JFIF H

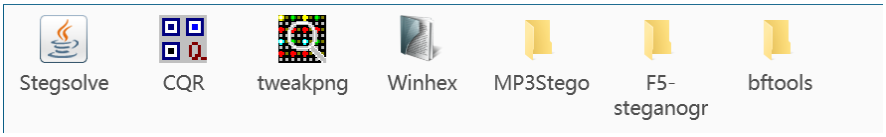
gif格式

头部：47 49 46 38 39 61　GIF89a



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	47	49	46	38	39	61	5A	01	56	01	F7	00	00	02	02	03	GIF89aZ V ÷
00000010	03	06	13	16	12	0C	1F	1D	1C	0A	11	27	19	23	2E	26	' #.&
00000020	13	0B	2B	24	1B	32	24	1C	37	28	1F	26	24	23	27	27	+$ 2$ 7(&$#''

在分析隐写文件的时候，会用到很多的工具，比如下面这些。我们接下来会讲些这些工具的用法。

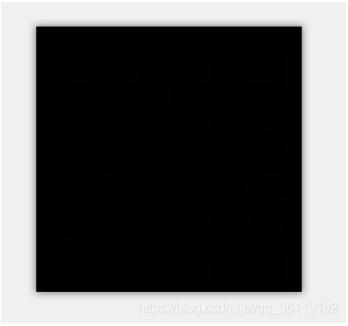


Stegsolve

stegsolve是一款用 java 写的图像隐写工具，可以查看图片的详细信息，每一层色阶等等，功能强大。

示例：http://www.shiyanbar.com/ctf/1768

把图片下载下来，打开，发现漆黑一片。



于是乎用Stegsolve打开，点击下面的向左或者向右按钮，出来了类似于二维码的图片，可是发现扫不出来。发现这跟正常的二维码图片有点不一样，好像反了。于是保存该图片

用PS打开，ctrl+i 进行反色，得到下面的二维码。扫描，得到flag。

Binwalk

Binwalk是用于搜索给定二进制镜像文件以获取嵌入其中的文件和代码的工具。 具体来说,它被设计用于识别嵌入固件镜像内的文件和代码。 Binwalk使用libmagic库,因此它与Unix文件实用程序创建的魔数签名兼容。 Binwalk还包括一个自定义魔数签名文件,其中包含常见的诸如压缩/存档文件,固件头,Linux内核,引导加载程序,文件系统等的固件映像中常见文件的改进魔数签名。

相关文章：https://www.aliyun.com/jiaocheng/122252.html

MP3Stego

MP3Stego是在将WAV文件压缩成mp3的过程中，将水印嵌入到mp3文件中。嵌入数据先被压缩、加密，然后隐藏在mp3比特流中，默认输出的mp3格式是单声道的128bit

示例：http://www.shiyanbar.com/ctf/58

下载该图片，是个美女，哇哇哇

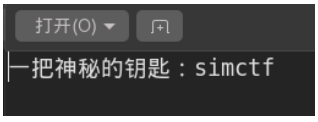
把它交给 binwalk 处理下，发现里面含有zip压缩文件，zip压缩文件里面包含mp3格式的文件和另一个文件

```
root@kali:~/Desktop# binwalk 3.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
8204          0x200C          TIFF image data, little-endian offset of first image directory: 8
15164         0x3B3C          Copyright string: "Copyright (c) 1998 Hewlett-Packard Company"
18187         0x470B          Unix path: /www.w3.org/1999/02/22-rdf-syntax-ns#"> <rdf:Description rdf:about="" xmlns:xmp="
http://ns.adobe.com/xap/1.0/" xmlns:dc="http://
80985         0x13C59         Zip archive data, at least v2.0 to extract, compressed size: 1407224, uncompressed size: 142
8163, name: music.mp3
1488494       0x16B66E        End of Zip archive
```

于是乎将其分离，使用 -e 参数。得到 music.mp3 和一个 txt 文件。

```
root@kali:~/Desktop# ls
3.jpg
root@kali:~/Desktop# binwalk -e 3.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
8204          0x200C          TIFF image data, little-endian offset of first image directory: 8
15164         0x3B3C          Copyright string: "Copyright (c) 1998 Hewlett-Packard Company"
18187         0x470B          Unix path: /www.w3.org/1999/02/22-rdf-syntax-ns#"> <rdf:Description rdf:about="" xmlns:xmp="
http://ns.adobe.com/xap/1.0/" xmlns:dc="http://
80985         0x13C59         Zip archive data, at least v2.0 to extract, compressed size: 1407224, uncompressed size: 142
8163, name: music.mp3
1488494       0x16B66E        End of Zip archive

root@kali:~/Desktop# ls
3.jpg  _3.jpg.extracted
root@kali:~/Desktop# cd _3.jpg.extracted/
root@kali:~/Desktop/_3.jpg.extracted# ls
13C59.zip  ÃèÊö.txt  music.mp3
```

打开txt文件，里面出现了钥匙。



把music.mp3文件放到MP3Stego里面，运行命令

```
decode.exe -X -P simctf music.mp3   #-X是获取隐藏的东西 -P指定密码
```

```
D:\Cracer渗透工具包v3.0\Cracer渗透工具包v3.0\Cracer渗透工具\隐写术\MP3Stego\MP3Stego>Decode.exe -X -P simctf music.mp3
MP3StegoEncoder 1.1.16
See README file for copyright info
Input file = 'music.mp3'  output file = 'music.mp3.pcm'
Will attempt to extract hidden information. Output: music.mp3.txt
the bit stream file music.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 3416]Avg slots/frame = 417.837; b/smp = 2.90; br = 127.963 kbps
Decoding of "music.mp3" is finished
The decoded PCM output file name is "music.mp3.pcm"
```


得到 music.mp3.txt 文件，打开，得到base64编码的数据

music.mp3.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
U2ltQ1RGe01QM19NUDNfc2RmZHNmfQ==
```

解码，得到 flag

利用MP3Stego加密

```
encode.exe -E key.txt -P 123456  test.wav  test.mp3  #将key.txt写入test.wmv中，密码为123456，最后胜出test.mp3文
```

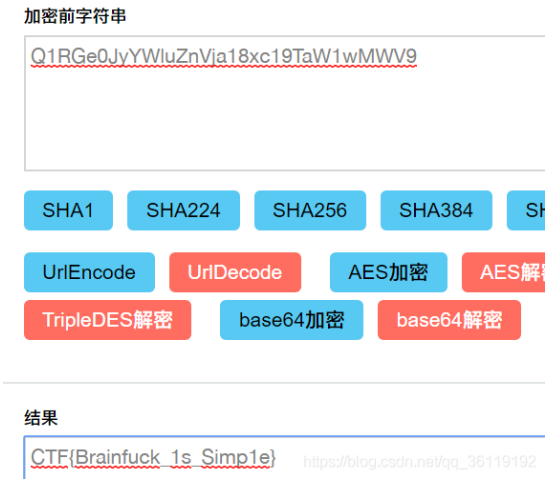
Bftools(Brainfuck)

```
PS D:\Cracer渗透工具包v3.0\Cracer渗透工具包v3.0\Cracer渗透工具\隐写术\bftools> .\bftools.exe -h

Command name not recognized.

Available commands are:

    run         - Run the given Brainfuck program.
    encode      - Encode input file using one of the languages.
    decode      - Decode input image using one of the languages.
    enlarge     - Enlarge an image by a given factor.
    reduce      - Shrink an image by a given factor.

    help <name> - For help with one of the above commands
```


示例：http://www.shiyanbar.com/ctf/1821

```
PS D:\Cracer渗透工具包v3.0\Cracer渗透工具包v3.0\Cracer渗透工具\隐写术\bftools> .\bftools.exe decode braincopter .\doge.jpg --output 1.jpg
PS D:\Cracer渗透工具包v3.0\Cracer渗透工具包v3.0\Cracer渗透工具\隐写术\bftools> .\bftools.exe run .\1.jpg
Q1RGe0JyYWluZnVjal8xc19TaW1wWVV9
```

然后将其进行Base64解码，得到flag



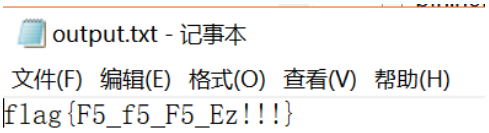
F5-steganography-master

示例：http://www.shiyanbar.com/ctf/1938

下载该图片 123456.jpg ，放入F5-steganography-master文件夹中，运行该命令

```
java Extract 123456.jpg -p 123456   #-p指定密码
```

得到 output.txt文件，打开，得到 flag

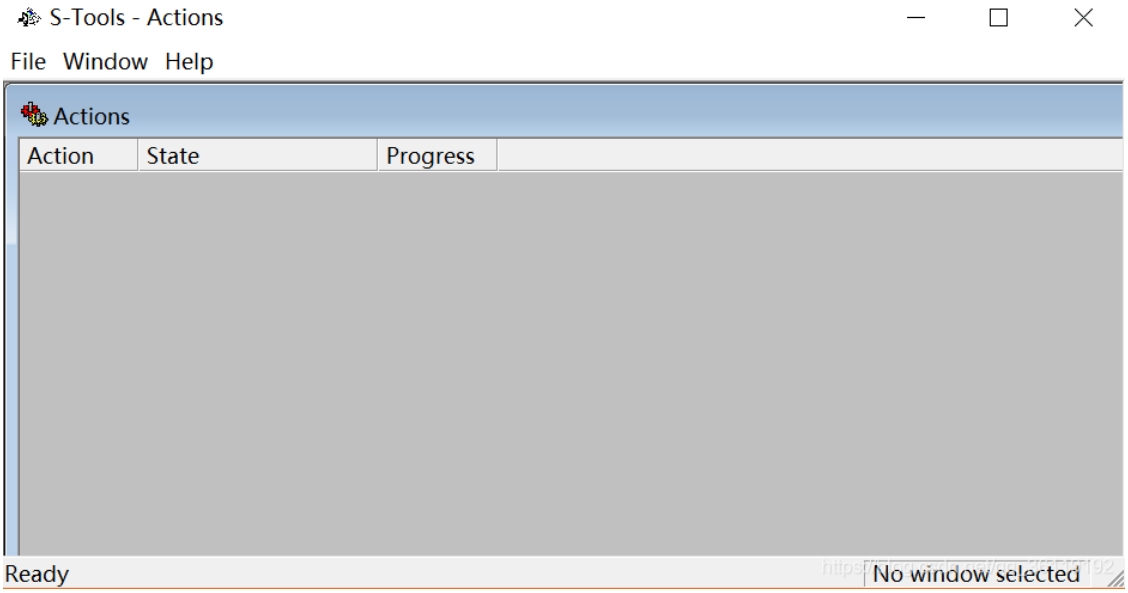


S-Tools

S-tools是一个时(空)域数字水印软件，支持 WAV 格式的音频文件、GIF和BMP格式的图像文件

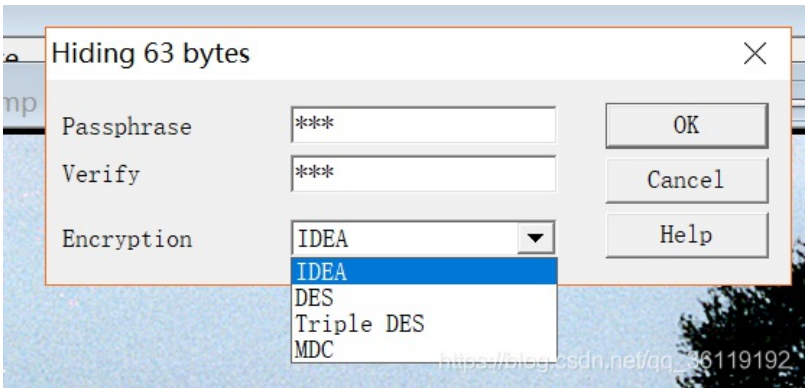
示例：我们现在有一个 original.bmp 的图片，我们现在要利用S-tools将一个密码文件key.txt隐藏到该图片中

打开S-tools，界面是这样的。

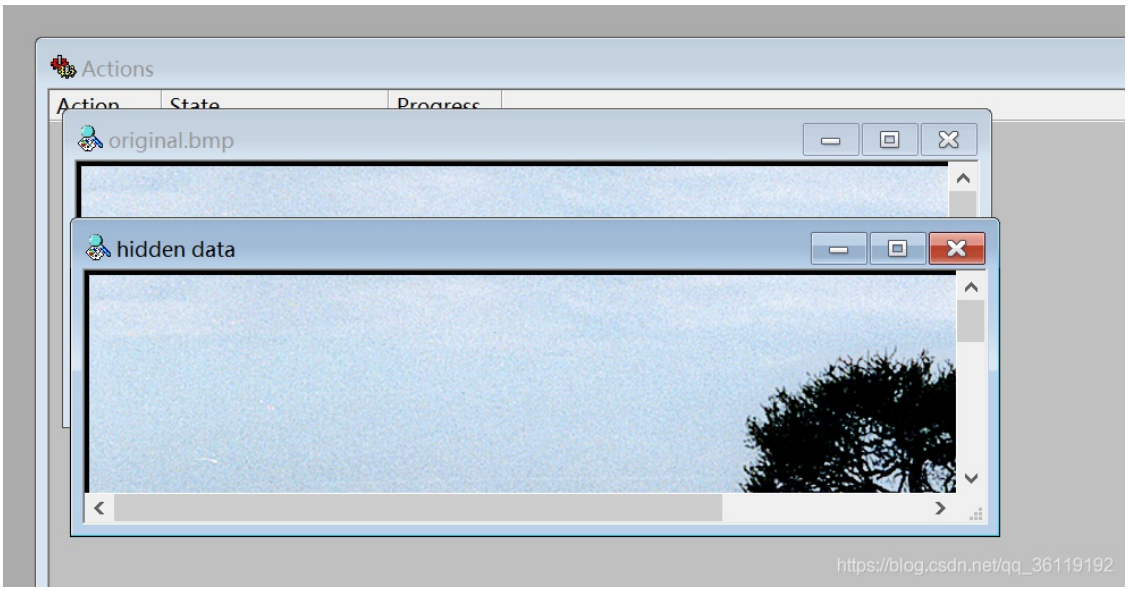


我们现在将图片拖进去

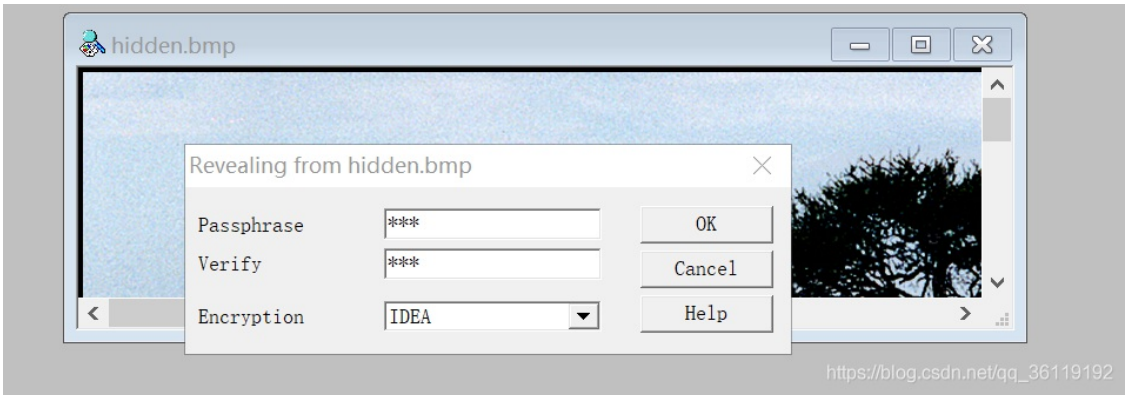
然后将我们的密码文件 key.txt 也拖进去 ， 此时会叫我们输入密码，我们输入密码，点击ok。这里还可以选择加密算法



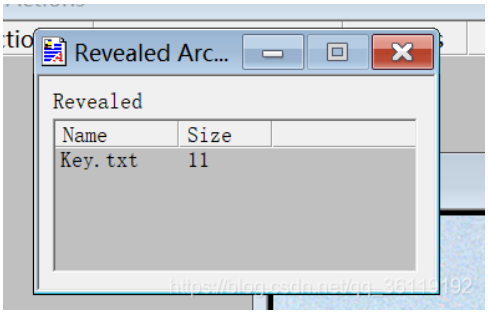
然后会生成我们写入隐藏数据后的图片，右键，save as 保存名为 hidden.bmp ！ 此时key.txt已经写入图片中了，并且图片可以正常打开，是看不到任何不一样的。



那么我们如何解密呢？把 hidden.bmp 拖入S-Tools中，右键->Reveal ，然后输入我们的密码。密码和加密算法必须和我们加密时候的一样



然后就显示我们写入的 key.txt 文件了



我们右键，Save as就可以把我们写入的文件保存下来了！

CTF隐写术的一些思路

当我们做CTF题，拿到图片之后

1. 第一步：先右键查看属性——>详细信息看有没有隐藏东西。如果没发现东西，
2. 第二步：用Stegsolve打开，看图层是否隐藏了东西。如果还是没有，
3. 第三步：则用Winhex打开图片，查看图片十六进制数据中是否隐藏了东西，有时候还需要修改图片的十六进制数据。传送门：https://www.jianshu.com/p/262397d9610b

当我们做CTF题，拿到压缩包之后

- 先解压缩，看能不能解压缩出来
- 拿binwalk看一下压缩包里面有没有隐藏东西

本文中的工具：链接: https://pan.baidu.com/s/1EJW0lEDZBi94TbRCsi74CQ 提取码: zyvr

相关文章：隐写数据