



【MISC】图片隐写与zip压缩包

原创

[Sunlight_316](#)  已于 2022-03-20 08:21:05 修改  539  收藏 1

分类专栏: [MISC](#) 文章标签: [开发语言](#)

于 2021-11-27 15:00:57 首次发布

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/weixin_51614272/article/details/121576450

版权



[MISC 专栏收录该内容](#)

2 篇文章 0 订阅

订阅专栏

目录

图片隐写解题思路1：按照工具划分

- 1.拿010分析一下二进制，查看图片由什么组成
- 2.拿stegsolve分析
- 3.tweakpng工具
- 4.kali工具分析
- 5.其他工具

图片隐写解题思路2：按照题目类型划分

- 1.基本操作和信息附加
 - 2.文件头分析
 - 3.lsb隐写
 - 4.IDAT块隐写
- 异常IDAT块导出后，生成一个二维码
- 5.CRC32隐写

png: 前四位是宽，后四位是高

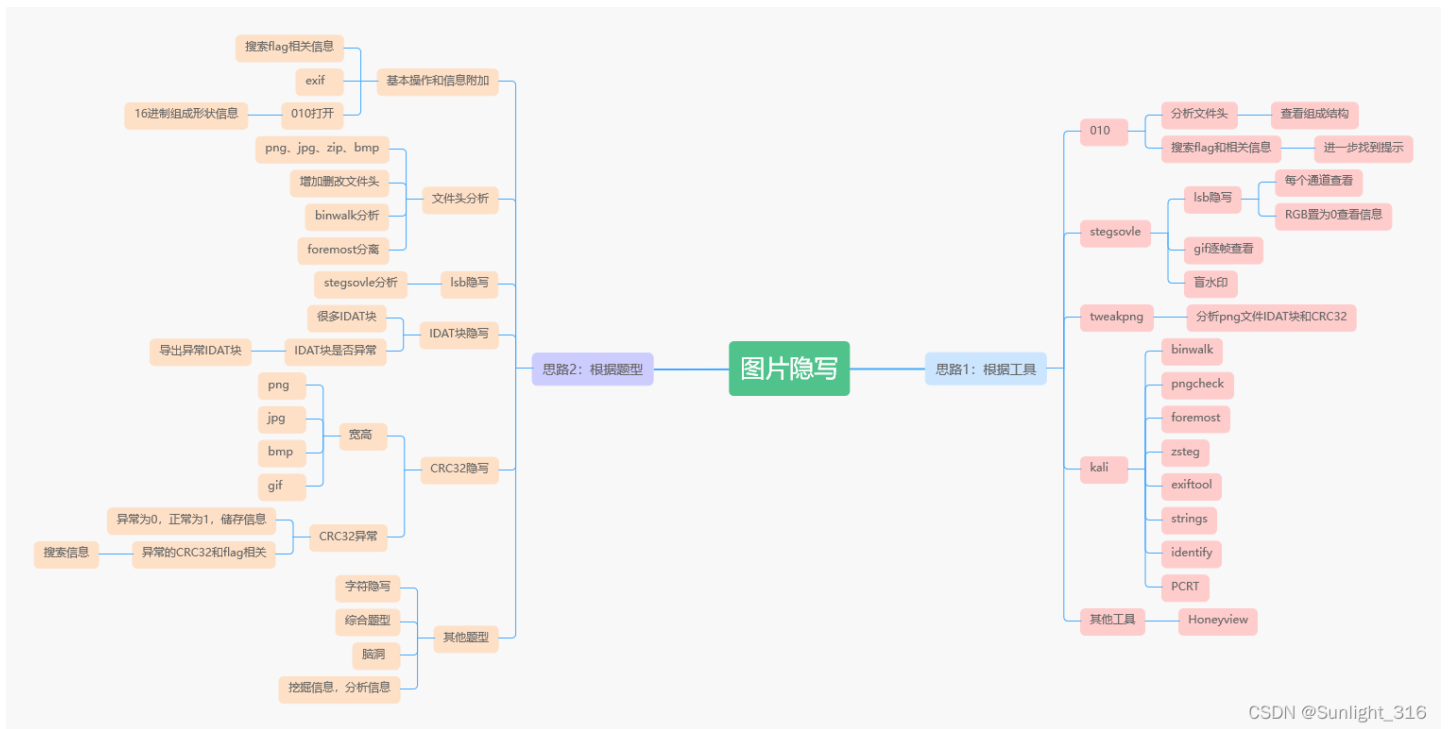
jpg: 前两位是高，后两位是宽

bmp:前四位是宽，后四位是高，但是是倒着写

- 6.其他题型

zip压缩包解题思路

- 1.拿010分析一下二进制，查看压缩包由什么组成
- 2.有密码，无思路：伪加密，拿zipop解一下
- 3.暴力破解，一般都是纯数字，或者低于7位的数字字母组合，否则时间太长
- 4.查看属性，查看注释
- 5.出现大于12kb的文本文档，明文攻击
- 6.crc一样——>CRC32碰撞
- 7.和字符隐写相结合，若删除时删好几下没反应，若出现莫名空白符，放在txt，word看看
- 8.持续更新中~~



图片隐写解题思路1：按照工具划分

1.拿010分析一下二进制，查看图片由什么组成

- 分析出是否包含zip文件，查看16进制文件头，查看有没有和**flag**相关的信息。
- 如果查出来了，可以分离出来，或者用binwalk
- 如果文件头类似png, jpg, 该补全的要补全文件头
- CRC是否正常，不正常，修改png文件的第二行第七列宽高

文件(F) 编辑(E) 搜索(S) 视图(V) 格式(O) 脚本(I) 模板(L) 调试(D) 工具(T) 窗口(W) 帮助(H)

起页码 flag.png flaggg.png x

```

0000h: 89 50 4E 47 0D 0A 1A 0A 00 00 00 00 49 48 44 52 %PNG.....IHDR
0010h: 00 00 03 45 00 00 02 61 08 02 00 00 00 2A 4D F3 ...E...a....*Mó
0020h: B2 00 00 00 01 73 52 47 42 00 AE CE 1C E9 00 00 ?...sRGB@f.é..
0030h: 00 04 67 41 4D 41 00 00 8F 0B FC 11 05 00 00 ..gAMA..±.Ua...
0040h: 00 09 70 48 59 73 00 00 1B 87 00 00 1D 87 01 8F .pHYs...±.±.±.
0050h: E5 F1 65 00 00 04 21 49 44 11 54 78 5E ED DD EB ãñ. @!IDAT^iYè
0060h: 61 A3 C8 12 06 D0 8D 6B 02 72 3C 8E C6 C9 38 98 aEè..D.k.r<ZèE8~
0070h: 59 21 DE D0 40 81 24 BB 5A 73 FC E7 EE 1D B7 50 Y!Pð@.$~Zsüçi..P
0080h: 71 AA 6D 3E 37 0F FD F7 D7 17 01 02 04 08 10 20 q>^m>7.y~x.....
0090h: 40 80 40 CD 02 FF D5 5C BC DA 09 00 20 40 80 00 @€@f.yò\WU.. @€.
00A0h: 01 02 04 FE CA 73 26 01 01 02 04 08 10 20 40 A0 ..pÈs&..... @
00B0h: 6E 01 79 AE EE FE A9 9E 00 01 02 04 08 10 20 20 n.y@ipøz.....
00C0h: CF 99 03 04 08 10 20 40 80 00 81 BA 05 E4 B9 BA I^m.... @€..°.@i°
00D0h: FB A7 7A 02 04 08 10 20 40 80 00 3C 67 0E 10 20 ùš<g... @€.
00E0h: 40 80 00 01 02 04 EA 16 90 E7 EA EE 9F EA 09 10 @€...è..çèiYè..
00F0h: 20 40 80 00 01 02 F2 9C 39 40 80 00 01 02 04 08 @€...òæ9@€.....
0100h: 10 A8 5B 40 9E AB BB 7F AA 27 40 80 00 01 02 04 .["@ž«»..°@€.....
0110h: 08 C8 73 E6 00 01 02 04 08 10 20 40 A0 6E 01 79 .Èsà..... @ n.y
0120h: AE EE FE A9 9E 00 01 02 04 08 10 20 20 CF 99 03 @ipøz..... I^m.
0130h: 04 08 10 20 40 80 00 81 BA 05 E4 B9 BA FB A7 7A ... @€...@i°ùšz
0140h: 02 04 08 10 20 40 80 80 3C 67 0E 10 20 40 80 00 ... @€<g... @€.
0150h: 01 02 04 EA 16 90 E7 EA EE 9F EA 09 10 20 40 80 ...è..çèiYè.. @€
0160h: 00 01 02 F2 9C 39 40 80 00 01 02 04 08 10 A8 5B ..òæ9@€.....["@ž«
0170h: 40 9E AB BB 7F AA 27 40 80 00 01 02 04 08 C8 73 @ž«»..°@€.....Ès
0180h: E6 00 01 02 04 08 10 20 40 A0 6E 01 79 AE EE FE a..... @ n.y@ip
0190h: A9 9E 00 01 02 04 08 10 20 20 CF 99 03 04 08 10 @ž..... I^m....
01A0h: 20 40 80 00 81 BA 05 E4 B9 BA FB A7 7A 02 04 08 @€..°.@i°ùšz...
01B0h: 10 20 40 80 80 3C 67 0E 10 20 40 80 00 01 02 04 . @€<g... @€.
01C0h: EA 16 90 E7 EA EE 9F EA 09 10 20 40 80 00 01 02 @è..çèiYè.. @€.
01D0h: F2 9C 39 40 80 00 01 02 04 08 10 A8 5B 40 9E AB òæ9@€.....["@ž«
01E0h: BB 7F AA 27 40 80 00 01 02 04 08 C8 73 E6 00 01 »..°@€.....Èsà...
01F0h: 02 04 08 10 20 40 A0 6E 01 79 AE EE FE A9 9E 00 ... @ n.y@ipøz.
0200h: 01 02 04 08 10 20 20 CF 99 03 04 08 10 20 40 80 ... I^m.... @€
0210h: 00 81 BA 05 E4 B9 BA FB A7 7A 02 04 08 10 20 40 ..°.@i°ùšz... @
0220h: 80 80 3C 67 0E 10 20 40 80 00 01 02 04 EA 16 90 €è<g... @€...è..
  
```

输出

```

执行模板 'D:\Users\下载\SweetScape\010 Templates\Repository\PNG.bt' 于 'C:\Users\12751\Desktop\flag.png'...
*ERROR: CRC Mismatch @ chunk[0]: in data: 6bb6add1; expected: 7a5eb365
  
```

输出 查找结果 多文件中查找 比较 直方图 校验和 进程

*ERROR: CRC Mismatch @ chunk[0]: in data: 6bb6add1; expected: 7a5eb365 址: 010h 值: 137 89h 大小: 16.709 十六进制(H) ANSI 小端 W 清除

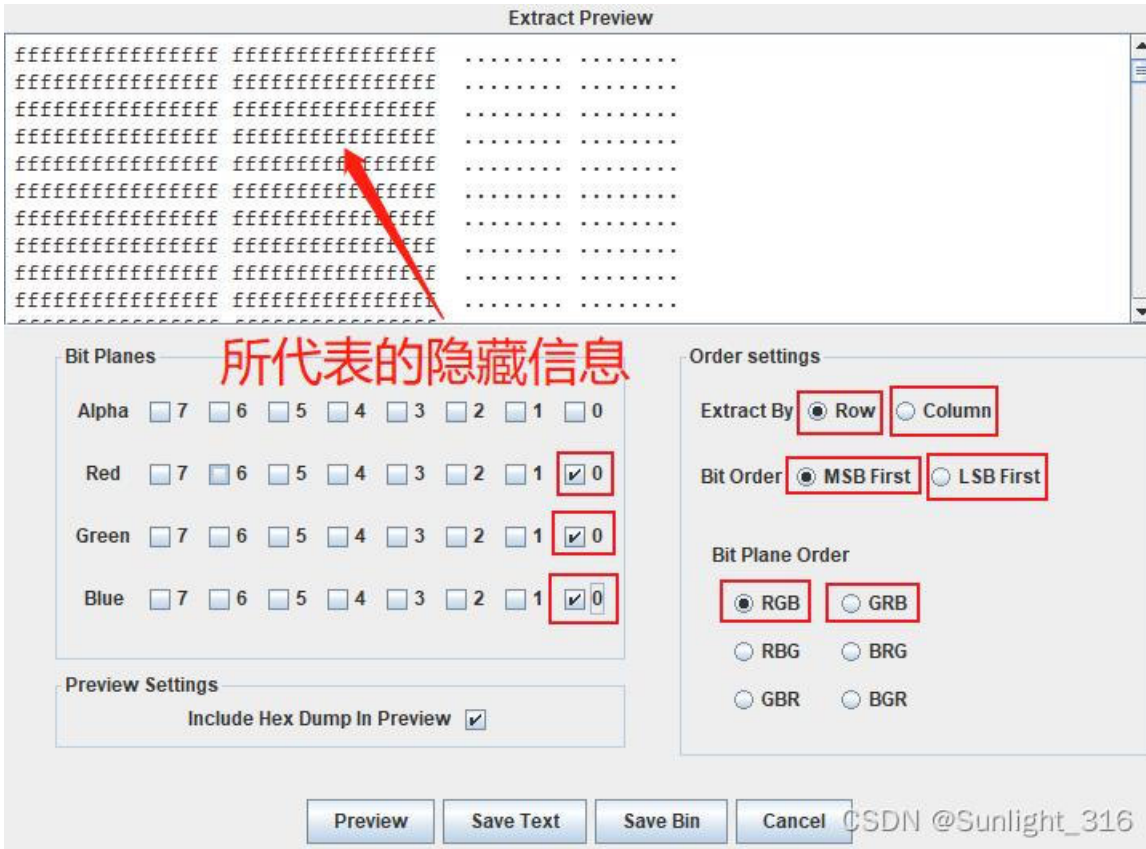
CSDN @Sunlight 316

A	B	C	D	E	F	G
1	扩展名	文件头	文件尾	描述	扩展名	Magic Number
2	png	89 50 4E 47 0D 0A 1A 0A	ae426082	Adobe Illustrator	.ai	25 50 44 46 [%PDF]
3	gif	474946383961(GIF89a)	003b	Bitmap graphic	.bmp	42 4D [BM]
4	zip	504b0304	504b	Class File	.class	CA FE BA BE
5	rar	52617221		IPEG graphic file	.jpg	FF D8
6				IPEG 2000 graphic file	.ip2	000000C6A5020200D0A [...iP...]
7				GIF graphic file	.gif	47 49 46 38 [GIF89]
8				TIF graphic file	.tif	49 49 [II]
9				PNG graphic file	.png	89 50 4E 47 .PNG
10				WAV audio file	.wav	52 49 46 46 [RIFF]
11				ELF Linux EXE	.elf	7F 45 4C 46 .ELF
12				Photoshop Graphics	.psd	38 42 50 53 [8BPS]
13				Zlib File	.zlib	78 9C
14				Windows Meta File	.wmf	D7 CD C6 9A
15				MIDI file	.mid	4D 54 68 64 [MThd]
16				Icon file	.ico	00 00 01 00
17				MP3 file with ID3 identity tag	.mp3	49 44 33 [ID3]
18				AVI video file	.avi	52 49 46 46 [RIFF]
19				Flash Shockwave	.swf	46 57 53 [FWS]
20				Flash Video	.flv	46 4C 56 [FLV]
21				Mpeg 4 video file	.mp4	00 00 00 18 66 74 79 70 6D 70 34 32 [...ftvmp42]
22				MOV video file	.mov	6D 6F 6F 76 [...moov]
23				Windows Video file	.wmv	30 26 B2 75 8E 06 CF
24				Windows Audio file	.wma	30 26 B2 75 8E 06 CF
25				PKZip	.zip	50 4B 03 04 [PK]
26				GZip	.gz	1F 8B 08
27				Tar file	.tar	75 73 74 61 72
28				Microsoft Installer	.msi	D0 CF 11 E0 A1 B1 1A E1
29				Object Code File	.obi	4C 01
30				Dynamic Library	.dll	4D 5A [MZ]
31				CAB Installer file	.cab	4D 53 43 46 [MSCF]
32				Executable file	.exe	4D 5A [MZ]
33				RAR file	.rar	52 61 72 21 1A 07 00 [Rar!..]
34				SYS file	.sys	4D 5A [MZ]
35				Help file	.hlp	3F 5F 03 00 [? ...]
36				VMWare Disk file	.vmdk	4B 44 4D 56 [KDMV]
37				Outlook Post Office file	.pst	21 42 44 4E 42 [!BDM]
38				PDF Document	.pdf	25 50 44 46 [%PDF]
39				Word Document	.doc	D0 CF 11 E0 A1 B1 1A E1
40				RTF Document	.rtf	7B 5C 72 74 66 31 [! tfl]
41				Excel Document	.xls	D0 CF 11 E0 A1 B1 1A E1
42				PowerPoint Document	.ppt	D0 CF 11 E0 A1 B1 1A E1
43				Visio Document	.vsd	D0 CF 11 E0 A1 B1 1A E1
44				DOCX (Office 2010)	.docx	50 4B 03 04 [PK]
45				XLSX (Office 2010)	.xlsx	50 4B 03 04 [PK]
46				PPTX (Office 2010)	.pptx	50 4B 03 04 [PK]
47				Microsoft Database	.mdb	53 74 61 6E 64 61 72 64 20 4A 65 74

CSDN @Sunlight 316

2. 拿stegsolve分析

- 1.把所有通道看一遍
- 2.最低位置零查看，一般都是后三位
- 3.如果确定是LSB隐写，那就把所有选项都试试
- 4.多尝试、多尝试、多尝试!!!



- 还有逐帧查看Frame Browser
- 盲水印隐写 Image Combiner

3.tweakpng工具

- 可以直接对各个IDAT块进行操作的工具

4.kali工具分析

```
sudo apt-get update //更新安装程序
sudo apt-get install XXX //安装工具包
//方向左键> -->当前灰色的指令
//方向上键^ -->上一条指令
//方向下键v-->下一条指令
```

```
biwalk misc17.png
step1:----binwalk -e misc17.png //如果binwalk分离不出来,用dd
step2:----dd if=misc17.png of=1.png skip=3892 bs=1
//if是目标文件//skip是起始位置//of是接收取出来的文件,看具体取出来是什么文件

pngcheck -v misc17.png //查看IDAT块信息,检测有没有异常IDAT块

foremost misc17.png
//直接将图片文件中包含的所有文件分离,输出到一个output文件夹中

step1:----zsteg misc17.png //用于检测被隐写在png, bmp图片里的数据。
step2:----zsteg -E "extradata:0" misc17.png > 1.txt
step3:----binwalk -e 1.txt

exiftool //apt-get install exiftool 读写和处理图像、音视频和PDF等文件的元数据
exiftool -ThumbnailImage -b misc22.jpg > 1.jpg //缩略图隐写

strings misc5.png | grep ctfshow //查找字符串ctfshow

identify -format "%T " misc39.gif > misc39.txt //提取gif的帧数

python PCRT.py -y -v -i misc44.png > 666.txt //自动化检测修复PNG损坏的取证工具
```

5.其他工具

- Honeyview, 比win10自带的图片查看其更加方便,也可以打开更多不同格式的图片

图片隐写解题思路2: 按照题目类型划分

1.基本操作和信息附加

- 直接搜索flag或者和flag有关的信息
- exif查看文件具体信息,和点开属性差不多
- 很多16进制信息组成一个特别的形状——>flag

2.文件头分析

- 要熟悉各种文件头, png、jpg、zip、bmp
- 有的是增加删改文件头,有的是根据文件头,打开010保存
- 用binwalk分析组成结构更快一点点
- 也可以用foremost直接分离出来

3.lsb隐写

- stegsolve分析

4.IDAT块隐写

- 对于有很多IDAT块的png，可以尝试删除几个后查看
- 分析IDAT块是否异常，没有65524就直接进入下一个IDAT块
- 分析IEDN块的长度是否是0，若不是，则说明有其他的隐藏信息

sctf.png (C:\Users\nihao\Desktop) - TweakPNG

Chunk	Length	CRC	Attributes
IHDR	13	5871e019	critical
sRGB	1	aece1ce9	ancillary, unsafe to
gAMA	4	0bfc6105	ancillary, unsafe to
pHYs	9	952b0e1b	ancillary, safe to c
IDAT	65445	3c52e386	critical
IDAT	65524	21a250d6	critical
IDAT	65524	dd582fbe	critical
IDAT	65524	939f6ecf	critical
IDAT	65524	cc5b8b36	critical
IDAT	65524	34e41cee	critical
IDAT	65524	526d60fe	critical
IDAT	65524	e5c2ad0c	critical
IDAT	65524	7c5eafb4	critical
IDAT	65524	87f6163d	critical
IDAT	65524	00a5e59f	critical
IDAT	65524	0df4a0fa	critical
IDAT	65524	2ab5b183	critical
IDAT	65524	b07349a3	critical
IDAT	65524	41bd8f1f	critical
IDAT	65524	251f6df9	critical
IDAT	65524	8787049d	critical
IDAT	65524	c1cf4fce	critical
IDAT	65524	6dd6c304	critical
IDAT	65524	c5bd9fb1	critical
IDAT	65524	1e255491	critical
IDAT	45027	68958fcd	critical
IDAT	138	d9cfa5a8	critical
IEND	0	ae426082	critical

异常IDAT块导出后，生成一个二维码

代码如下：

1.使用zlib进行压缩

```
import zlib

import binascii

IDAT = "十六进制代码块:4ca45a".decode('hex')

#print IDAT

result = binascii.hexlify(zlib.compress(IDAT))

print (result.decode('hex'))

print (len(result.decode('hex')))
```

2.用python来生成一个25*25的二维码

```

from PIL import Image
from zlib import *

MAX = 25
pic = Image.new("RGB", (MAX,MAX))
str = "111111100010000110111111100000"

i=0
for y in range(0,MAX):
    for x in range(0,MAX):
        if(str[i] == '1'):
            pic.putpixel([x,y],(0,0,0))
        else:pic.putpixel([x,y],(255,255,255))
        i = i+1
pic.show()
pic.save("flag.png")

```

5.CRC32隐写

用tweakpng打开发现CRC校验错误，说明要改宽高了

1.修改宽高，先打开属性，确定目前宽高数据，再转换成十六进制在010中找，就很容易找到

- 用脚本直接爆破出png，jpg正确的宽高
- 原理就是根据CRC32算出宽高来

png: 前四位是宽，后四位是高

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	01	F4	00	00	01	A4	08	06	00	00	00	CB	D6	DF	ó ¤ EöB
00000020	8A	00	00	00	09	70	48	59	73	00	00	12	74	00	00	12	! pHYs t
00000030	74	01	DE	66	1F	78	00	00	0A	4D	69	43	43	50	50	68	t ðf x MiCCPPh
00000040	6F	74	6F	73	68	6F	70	20	49	43	43	20	70	72	6F	66	otoshop ICC prof
00000050	69	6C	65	00	00	78	DA	9D	53	77	58	93	F7	16	3E	DF	ile xÜ SwX!÷ >B

jpg: 前两位是高，后两位是宽

FFC0 是固定值的位置，后三位是数据长度和精度。再后四位就是高和宽了

I SOF0, Start of Frame, 帧图像开始

u 标记代码 2字节 固定值0xFFC0

u 包含9个具体字段:

- ① 数据长度 2字节 ①~⑥六个字段的总长度
即不包括标记代码, 但包括本字段
- ② 精度 1字节 每个数据样本的位数
通常是8位, 一般软件都不支持 12位和16位
- ③ 图像高度 2字节 图像高度 (单位: 像素), 如果不支持 DNL 就必须 >0
- ④ 图像宽度 2字节 图像宽度 (单位: 像素), 如果不支持 DNL 就必须 >0
- ⑤ 颜色分量数 1字节 只有3个数值可选
1: 灰度图; 3: YCrCb或YIQ; 4: CMYK
而JFIF中使用YCrCb, 故这里颜色分量数恒为3
- ⑥ 颜色分量信息 颜色分量数*3字节 (通常为9字节)

bmp:前四位是宽, 后四位是高, 但是是倒着写

03 B6——>写成B6 03

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ANSI	ASCII
00000000	42	4D	50	87	06	00	00	00	00	00	36	00	00	00	28	00	BMP+	6 (
00000016	00	00	B6	03	00	00	96	00	00	00	01	00	18	00	00	00	█	-
00000032	00	00	1A	87	06	00	12	0B	00	00	12	0B	00	00	00	00	‡	
00000048	00	00	00	00	00	00	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿ	
00000064	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿ	
00000080	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿ	
00000096	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿ	
00000112	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿ	

2.CCRC错误

- PCRT工具
- 不是很熟悉这种题型

6.其他题型

- 图片隐写可能和字符隐写相结合
- 以上只是基本题型, 现实题目会根据各种不同题型进行混合
- 主要还是分析出怎么藏信息, 为什么可以这样藏信息
- 不断尝试不断试探各种解法去寻找, 开脑洞
- 不断挖掘出隐藏的信息和提示, 在进行下一步分析

zip压缩包解题思路

这类题目经常和图片隐写相结合, 互相运用, 融会贯通

- 1.拿010分析一下二进制, 查看压缩包由什么组成
- 2.有密码, 无思路: 伪加密, 拿zipop解一下
- 3.暴力破解, 一般都是纯数字, 或者低于7位的数字字母组合, 否则时间太长

4.查看属性，查看注释

5.出现大于12kb的文本文档，明文攻击

6.crc一样——>CRC32碰撞

7.和字符隐写相结合，若删除时删好几下没反应，若出现莫名空白符，放在txt，word看看

8.持续更新中~~