

ctf 图片

原创

[cha0ski11](#) 于 2021-06-01 21:27:11 发布 617 收藏 5

文章标签: [信息安全](#)

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_52612705/article/details/117402402

版权

文章目录

[图片里蕴含信息](#)

[改后缀](#)

[16进制](#)

[属性](#)

[binwalk分离](#)

[藏在另一张图](#)

[手撸](#)

[thumbnail隐写](#)

[藏在时间里](#)

[高度](#)

[宽度](#)

[同时爆破宽度和高度](#)

[藏在动态图中](#)

[特殊字符组成](#)

[IDAT](#)

[stegsolve](#)

图片里蕴含信息

这就得具体情况具体分析了。

改后缀

可能一开始给我们的不是图片，而是文档之类的，用记事本打开



如果像这一类的就是改PNG图片，

有个别需要改后缀名少见，

图片后缀很多，例如png、jpg、bmp、gif、tif、webp也有在线转化的网站哦。

但也有像bpg文件，无法直接查看。到网站<https://bellard.org/bpg/>下载

download

The following archive contains the source code of the `bpgenc`, `bpgdec` and `bpgview` components and the Javascript decoder.

Source code: [bpg-0.9.8.tar.gz](#)

Binary distribution for Windows (64 bit only): [bpg-0.9.8-win64.zip](#)

Official [Github mirror](#).

For Mac users, the BPG utilities are available in the `libbpg` [Homebrew](#) formula.

把你查看的bpg文件放到你下载工具的目录下，打开命令符，输入**bpgview.exe** 文件名就行了

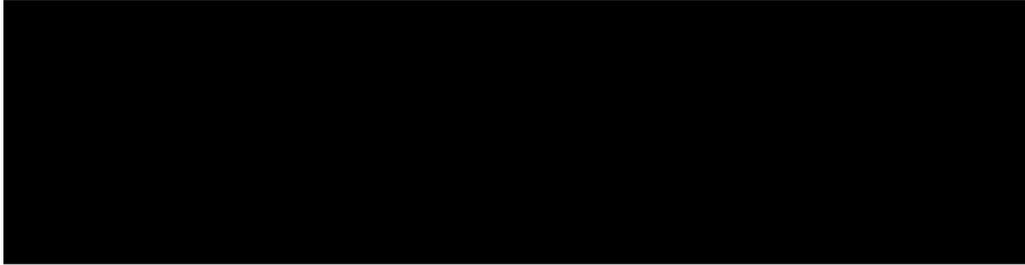
doc	2021/5/29 22:18	文件夹	
html	2021/5/29 22:18	文件夹	
bpgdec.exe	2018/4/21 17:40	应用程序	135 KB
bpgenc.exe	2018/4/21 17:40	应用程序	12,956 KB
bpgview.exe	2018/4/21 17:40	应用程序	139 KB
ChangeLog	2018/4/21 17:40	文件	2 KB
libgcc_s_seh-1.dll	2018/4/21 17:40	应用程序扩展	486 KB
libjpeg-62.dll	2018/4/21 17:40	应用程序扩展	371 KB
libpng16-16.dll	2018/4/21 17:40	应用程序扩展	217 KB
libstdc++-6.dll	2018/4/21 17:40	应用程序扩展	8,425 KB
libtiff-5.dll	2018/4/21 17:40	应用程序扩展	428 KB
libwinpthread-1.dll	2018/4/21 17:40	应用程序扩展	57 KB
misc3.bpg	2021/2/4 17:21	BPG 文件	7 KB
README	2018/4/21 17:40	文件	10 KB
SDL.dll	2018/4/21 17:40	应用程序扩展	345 KB
SDL_image.dll	2018/4/21 17:40	应用程序扩展	61 KB
stderr.txt	2021/5/30 18:56	文本文档	0 KB
stdout.txt	2021/5/30 18:56	文本文档	0 KB
zlib1.dll	2018/4/21 17:40	应用程序扩展	89 KB

```
C:\Windows\System32\cmd.exe
```

```
:\网安工具系列\bpg-0.9.8-win64>bpgview.exe misc3.bpg
```

```
:\网安工具系列\bpg-0.9.8-win64>
```

```
bpgview [1 of 1] - misc3.bpg
```



v{aade771916df7cde3009c0e631f

有些需要先去在线网站把图片格式转化，才能直接binwalk提取就能得到flag.png了。

例如png格式转为bmp格式，bmp格式下，中间的位置插入了一个gzip的数据，直接肉眼很难看出来，至于为什么原本的png格式下，binwalk得不到结果呢？大师傅们的解释是png和bmp像素点的读取方式不一样。

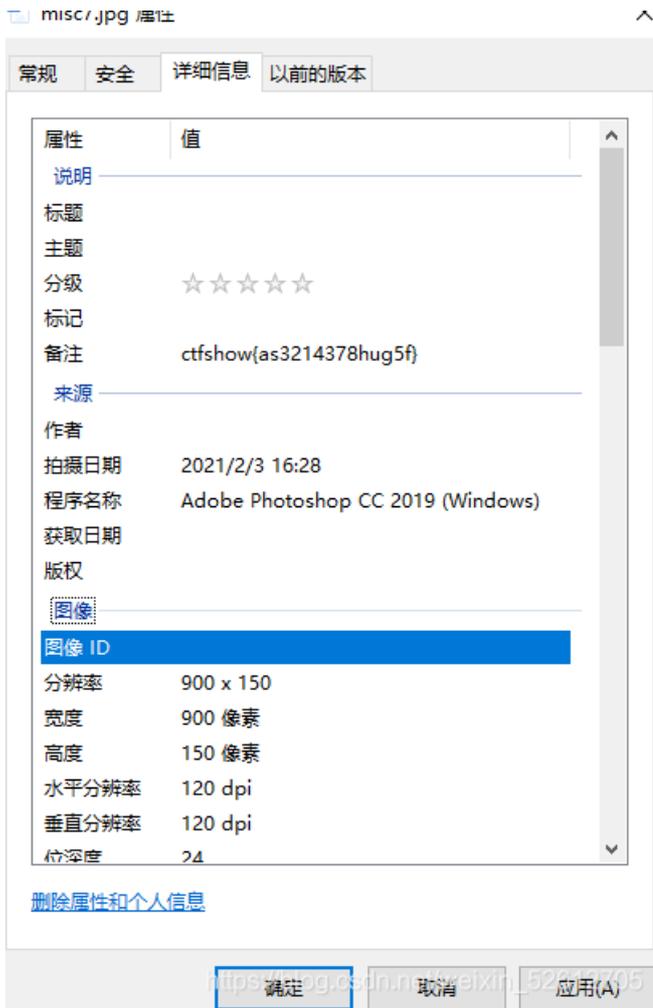
16进制

这里我用的是winhex

```
2E 30 00 38 42 49 4D 04 25 00 00 00 00 00 10 00 .0 8BIM %
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 38 BIM : * 8
42 49 4D 04 3A 00 00 00 00 00 D7 00 00 00 10 00 printCu
00 00 01 00 00 00 00 00 0B 70 72 69 6E 74 4F 75 tput PstS
74 70 75 74 00 00 00 05 00 00 00 00 50 73 74 53 bool Inteenu
62 6F 6F 6C 01 00 00 00 00 49 6E 74 65 65 6E 75 m Inte Img
6D 00 00 00 00 49 6E 74 65 00 00 00 00 49 6D 67 ctfshow{d5e
20 00 00 00 0F 63 74 66 73 68 6F 77 7B 64 35 65 937aefb091d38e70
39 33 37 61 65 66 62 30 39 31 64 33 38 65 37 30 d927b80ele2ea}
64 39 32 37 62 38 30 65 31 65 32 65 61 7D 00 01 printProof
00 00 00 00 0F 70 72 69 6E 74 50 72 6F 6F 66 SetupObjc h!h
53 65 74 75 70 4F 62 6A 63 00 00 00 05 68 21 68 7<% n proof
37 8B BE 7F 6E 00 00 00 00 00 0A 70 72 6F 6F 66 Setup Blt
53 65 74 75 70 00 00 00 01 00 00 00 00 42 6C 74 nenum builtin
6E 65 6E 75 6D 00 00 00 0C 62 75 69 6C 74 69 6E
```

属性

有些信息隐藏在属性中



有的和你说在图片属性中，但是右键找不到，右键的信息比较少，
https://exif.tuchong.com/信息比较全

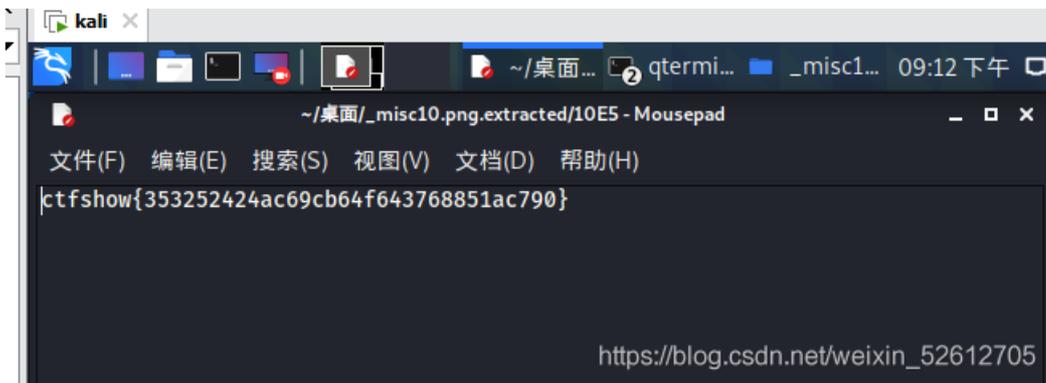
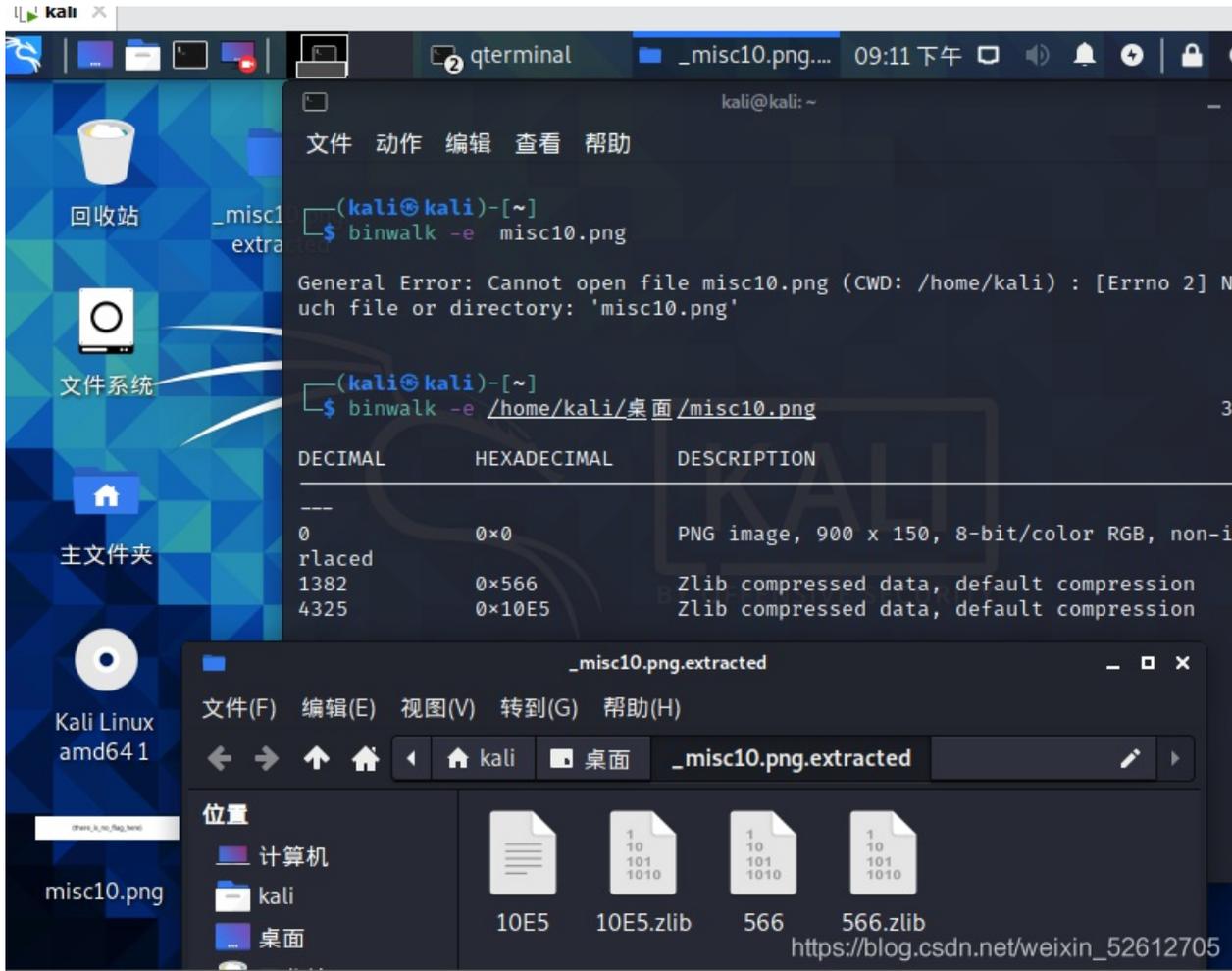
名称	值	开始	大小	颜色	注释
> struct PNG_SIGNATURE sig		0h	8h	Fg: Bg: [color swatch]	
> struct PNG_CHUNK chunk[0]	IHDR (Critical, Public, Unsafe to Copy)	8h	19h	Fg: Bg: [color swatch]	
> struct PNG_CHUNK chunk[1]	pHYs (Ancillary, Public, Safe to Copy)	21h	15h	Fg: Bg: [color swatch]	
> struct PNG_CHUNK chunk[2]	iTXt (Ancillary, Public, Safe to Copy)	36h	528h	Fg: Bg: [color swatch]	
> struct PNG_CHUNK chunk[3]	tEXt (Ancillary, Public, Safe to Copy)	55Eh	3Dh	Fg: Bg: [color swatch]	
uint32 length	49	55Eh	4h	Fg: Bg: [color swatch]	
> union CTYPE type	tEXt	562h	4h	Fg: [color swatch] Bg: [color swatch]	
> struct PNG_CHUNK_TEXT text	Warning = ctfshow[5c5e819508a3ab1fd823f11e83e93c75]	566h	31h	Fg: Bg: [color swatch]	
uint32 crc	6A940E9h	597h	4h	Fg: [color swatch] Bg: [color swatch]	
> struct PNG_CHUNK chunk[4]	IDAT (Critical, Public, Unsafe to Copy)	59Bh	B7Fh	Fg: Bg: [color swatch]	
> struct PNG_CHUNK chunk[5]	IEND (Critical, Public, Unsafe to Copy)	111Ah	Ch	Fg: Bg: [color swatch]	

在模板块中也能找到

binwalk分离

而有些东西需要分离才能查到

用binwalk -e



藏在另一张图

这是一张图。

{there_is_no_flag_here}

https://blog.csdn.net/weixin_52612705

放入tweakpng中，删除一个IDAT块，保存为新的图片。得到flag

misc11_1.png (G:\网络攻防\ctf.show\MISC\11\misc11 (1) - tweakPNG)

Chunk	Length	CRC	Attributes	Contents
IHDR	13	09dad...	critical	PNG image header: 900x150, 8 bits/sample,
IDAT	2931	c464a...	critical	PNG image data
IDAT	7541	228b6...	critical	PNG image data
IEND	0	ae426...	critical	end-of-image marker

https://blog.csdn.net/weixin_52612705

ctfshow{44620176948fa759d3eeafeac99f1ce9}

https://blog.csdn.net/weixin_52612705

手撸

binwalk分析发现有额外数据，直接binwalk -e或者foremost分离不出来，无奈手撸。

从选中的数据开始，复制到结尾，新建为一个jpg文件。

000007E0	00 01 00 01 1B 00 00 00	00 00 01 00 02 00 00 02 01 00 04 00	(
00000800	28 00 03 00 00 00 01 00	02 00 00 02 01 00 04 00	;
00000810	00 00 01 00 00 01 A6 02	02 00 04 00 00 00 01 00	
00000820	00 04 D5 00 00 00 00 00	00 00 48 00 00 00 01 00	õ
00000830	00 00 48 00 00 00 01 FF	D8 FF E0 00 10 4A 46 49	H
00000840	46 00 01 01 01 00 78 00	78 00 00 FF DB 00 43 00	yøÿà JFI
00000850	02 01 01 02 01 01 02 02	02 02 02 02 02 02 03 05	F x x yÛ C
00000860	03 03 03 03 03 06 04 04	03 05 07 06 07 07 06	

thumbnail隐写

用magicexif(<http://www.xue51.com/soft/23613.html>)打开

MagicEXIF 元数据编辑器 v1.07 (未注册) - misc22.jpg

文件(F) 编辑(E) 查看(V) 图像(I) 工具(T) 帮助(H)

新建 打开 保存 另存为 导入 导出 编辑项 增添项 删除项 JPEG段 原图重构 编辑向导 批处理 查找 注册产品

misc22.jpg
JPEG 图像

{there is no flag here}
ctfshow{dbf7d3f84b0125e833dfd3c80820a129}

项目	值	标签号	标签名	数据类型	组件数	字节
缩略图信息 (IFD1)						
123 压缩方案	JPEG压缩	0103	Compression	SHORT	1	2
17 水平分辨率	72	011A	XResolution	RATIONAL	1	8
17 垂直分辨率	72	011B	YResolution	RATIONAL	1	8
128 分辨率单位	英寸	0128	ResolutionUnit	SHORT	1	2

文件大小: 21.72 KB
图像大小: 900 × 150 像素
位深度: 24 位
压缩指纹: E28603EB (IJG)
字节序: Motorola (大端字节序)
创建时间: 2021-03-27 14:07:50
最后修改: 2021-03-27 14:09:20

拍摄信息 常规信息 GPS 信息 厂商注释 全部 EXIF 信息 未知原始性

G:\网络攻防\ctf.show\MISC入门\misc22\misc22.jpg JPEG 图像 共有 4 项 (1 个目录) https://blog.csdn.net/weixin_52612705

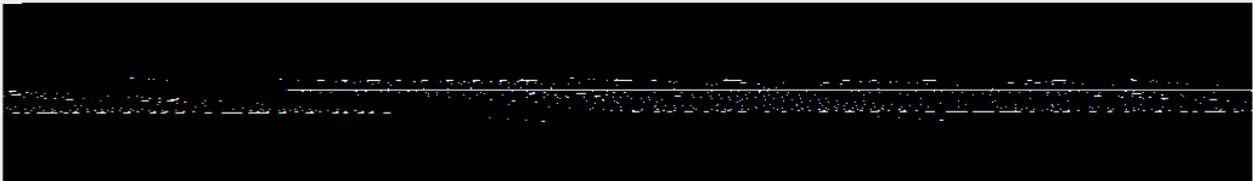


藏在时间里

我们一般不知道多少，而宽也没法像高度试试，所以复制了大佬代码。

同时爆破宽度和高度

```
filename = "misc32.png"
with open(filename, 'rb') as f:
    all_b = f.read()
    data = bytearray(all_b[12:29])
    n = 4095
    for w in range(n):
        width = bytearray(struct.pack('>i', w))
        for h in range(n):
            height = bytearray(struct.pack('>i', h))
            for x in range(4):
                data[x+4] = width[x]
                data[x+8] = height[x]
            crc32result = zlib.crc32(data)
            #替换成图片的crc
            if crc32result == 0xE14A4C0B:
                print("宽为: ", end = '')
                print(width, end = ' ')
                print(int.from_bytes(width, byteorder='big'))
                print("高为: ", end = '')
                print(height, end = ' ')
                print(int.from_bytes(height, byteorder='big'))
```



https://blog.csdn.net/weixin_52612705

ctfshow{685082227bcf70d17d1b39a5c1195aa9}

https://blog.csdn.net/weixin_52612705

藏在动态图中

有些GIF图片闪的快，可能就在某一帧中。

需要注意的是APNG图片也是动态图，如果直接用图片查看，看到的只是一张图片，如果用浏览器打开就是个动态图。需要用工具APNG Disassembler分离

不过也有是利用不同帧之间的间隔时间来隐写的。kali里用命令identify -format "%T " misc39.gif > misc39.txt提取帧数。

特殊字符组成

A20h:	02 8A 28 A0	02 8A 28 A0	0F 4D C9 4E	9D 58 55 D8	.Š(.Š(.MĚN.XUØ
A30h:	B5 FD 47 69	53 D7 FF 5B	01 6A F0 01	01 E0 EE DF	ųýGİS×ý[.jđ.àİß
A40h:	F0 01 F0 01	F0 01 EA 39	F0 01 F0 01	F0 01 87 55	đ.đ.đ.ê9đ.đ.đ.đ.đ.đ.
A50h:	F0 01 A3 B2	47 4B 4C F6	FC AC F0 01	EF C7 2D A1	đ.đ.đ.GKLöü-đ.đ.đ.đ.đ.
A60h:	F0 01 84 80	67 39 B8 BF	67 8B F0 01	1E 8F AB 89	đ.đ.đ.đ.đ.đ.đ.đ.đ.đ.
A70h:	F0 01 F0 01	F0 01 EA 0E	A3 03 F0 01	F0 01 6C 60	đ.đ.đ.đ.đ.đ.đ.đ.đ.đ.
A80h:	05 50 0E 4D	31 A1 21 93	A2 F3 FB 0B	D5 ED 4F 0A	.P.M1;!"cóú.ŌíŌ.
A90h:	D3 78 F0 01	F0 01 39 6D	A4 5B F0 01	F0 01 66 75	Óxđ.đ.đ.đ.đ.đ.đ.đ.đ.
AA0h:	F3 AD F0 01	48 67 0D A4	F0 01 9E 90	47 72 38 72	ó-đ.Hg.đ.đ.đ.đ.đ.đ.
AB0h:	F0 01 F0 01	F0 01 74 26	F0 01 F0 01	95 C7 F5 FF	đ.đ.đ.đ.đ.đ.đ.đ.đ.đ.
AC0h:	C0 38 F0 01	1E 50 00 1A	15 80 8D 0F	F0 01 01 D7	Á8đ.đ.đ.đ.đ.đ.đ.đ.
AD0h:	F0 01 F0 01	F1 06 68 94	F0 01 F0 01	43 07 03 49	đ.đ.đ.đ.đ.đ.đ.đ.đ.đ.
AE0h:	4B 41 41 C9	9B 0E E8 6A	EB 73 E1 D2	76 58 11 4A	KAAĚ.đ.đ.đ.đ.đ.đ.
AF0h:	F0 01 12 94	0A 13 24 01	FE 15 39 D1	56 68 9F 9A	đ.đ.đ.đ.đ.đ.đ.đ.đ.đ.
B00h:	F0 01 2E 6B	3A 6F C1 F8	F0 01 F0 01	F0 01 D7 16	đ.đ.đ.đ.đ.đ.đ.đ.đ.đ.
B10h:	F0 01 F0 01	F0 01 CA D2	F0 01 4A E6	F0 01 5E 9B	đ.đ.đ.đ.đ.đ.đ.đ.đ.đ.
B20h:	F0 01 EC 72	F0 01 DC 88	F0 01 16 27	F0 01 3C 9A	đ.đ.đ.đ.đ.đ.đ.đ.đ.đ.
B30h:	F0 01 66 62	F0 01 A2 EA	F0 01 F0 01	F0 01 1E 6E	đ.đ.đ.đ.đ.đ.đ.đ.đ.đ.
B40h:	F8 EE 08 C9	CA 06 EF 2D	FE 04 73 2E	B9 C2 AE E2	ŏí.ĚĚ.đ.đ.đ.đ.đ.đ.
B50h:	F0 01 1A BA	FE 30 CC 84	F0 01 82 1F	F0 01 F0 01	đ.đ.đ.đ.đ.đ.đ.đ.đ.đ.
B60h:	F0 01 B9 54	F0 01 E5 80	F0 01 9E 3E	F0 01 84 7A	đ.đ.đ.đ.đ.đ.đ.đ.đ.đ.
B70h:	F0 01 4B 45	F0 01 7D 15	F0 01 F0 01	F0 01 DC 10	đ.đ.đ.đ.đ.đ.đ.đ.đ.đ.
B80h:	F0 01 7D 6D	F0 01 0A 8C	F0 01 49 9A	F0 01 EE 88	đ.đ.đ.đ.đ.đ.đ.đ.đ.đ.
B90h:	D8 B4 F0 01	B4 C8 F0 01	5B 12 D4 61	F0 01 F0 01	ŏ'đ.đ.đ.đ.đ.đ.đ.đ.
BA0h:	AF 4E 61 3D	98 01 B4 A9	8E 16 5B 91	67 9E 5B A6	Na=".'@Ž.['gž[!
BB0h:	64 BB F0 01	F0 01 21 EA	BE 99 3B FD	31 C5 02 42	d»đ.đ.đ.đ.đ.đ.đ.đ.
BC0h:	B9 F3 F0 01	19 CB 06 4B	F0 01 F0 01	F0 01 6C 06	'óđ.đ.đ.đ.đ.đ.đ.đ.
BD0h:	F0 01 F0 01	F0 01 26 C5	F0 01 12 2B	8B BE C5 33	đ.đ.đ.đ.đ.đ.đ.đ.đ.đ.
BE0h:	96 5F F0 01	FA 47 F8 F6	F0 01 C0 76	B2 E7 14 1D	-_đ.đ.đ.đ.đ.đ.đ.đ.
BF0h:	F0 01 F0 01	E3 B6 CF FE	F0 01 F0 01	https://blog.csdn.net/qq_352612705	
C00h:	1C 62 85 58	6E AF 32 A0	03 B9 97 B9	75 AF E3 3C	b.Xo@2.'-!u.ó<

IDAT

用tweakpng打开图片，发现这七个数正好是ctfshow对应的ascii码

IDAT	99	d6
IDAT	116	af
IDAT	102	d7
IDAT	115	b5
IDAT	104	dc
IDAT	111	3c
IDAT	119	92

![在这里插入图片描述](https://img-blog.csdnimg.cn/20210601203607739.png)

更过分的是

用tweakpng打开，报了一堆错，然后使用pngdebugger分析，发现所有IDAT块的crc32值都是错误的，所以也可能藏在错误中，如果对错都多的话有可能对是0或1。

```
Microsoft Windows [版本 10.0.18363.418]
(c) 2019 Microsoft Corporation. 保留所有权利。

C:\网安工具系列\png-debugger-master\Debug>PNGDebugger.exe misc43.png
-----
file-path=misc43.png
file-size=4560 bytes

x00000000      png-signature=0x89504E470D0A1A0A

x00000008      chunk-length=0x0000000D (13)
x0000000C      chunk-type=' IHDR'
x0000001D      crc-code=0x09DAD161
> (CRC CHECK)  crc-computed=0x09DAD161          =>    CRC OK!

x00000021      chunk-length=0x00000180 (384)
x00000025      chunk-type=' IDAT'
x0000001A9     crc-code=0xE59387E5
> (CRC CHECK)  crc-computed=0x8385F691      =>    CRC FAILED

x000001AD      chunk-length=0x00000180 (384)
x000001B1      chunk-type=' IDAT'
x00000335      crc-code=0x93A62E63
> (CRC CHECK)  crc-computed=0x42434298      =>    CRC FAILED

https://blog.csdn.net/weixin_52612705
x00000339      chunk-length=0x00000180 (384)
```

还有每一个IDAT块前面都会有一个fcTL块，它其中就包含水平垂直偏移量。所以可能是坐标啥的。

stegsolve

这个我另一篇博客有写，但是考虑到比较懒的，就在这里在写一遍吧。

格式

File Format:文件格式

Data Extract:数据提取

Stereogram Solve:立体试图 可以左右控制偏移

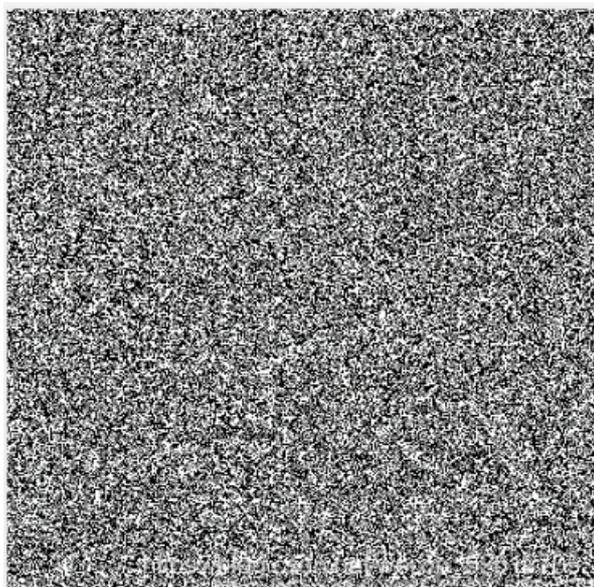
Frame Browser:帧浏览器

Image Combiner:拼图，图片拼接

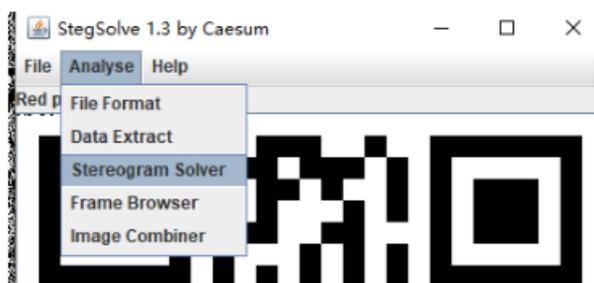
用法（使用场景）

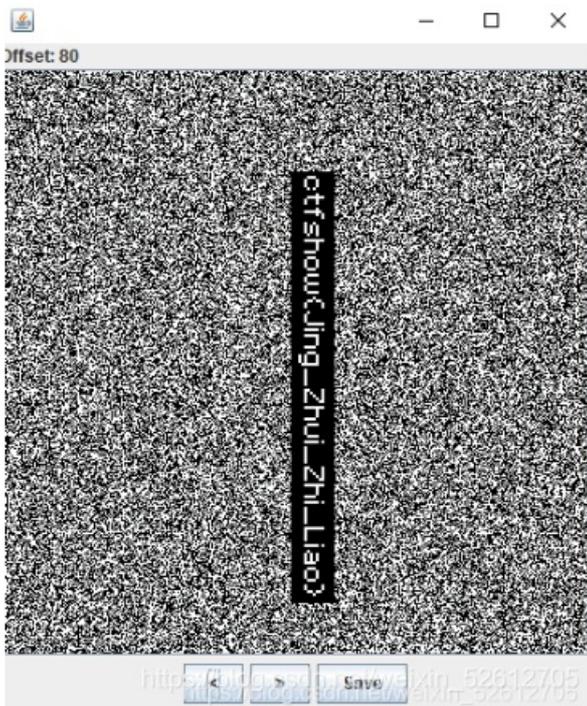
- 1.File Format:这里你会看见图片的具体信息有时候有些图片隐写的flag会藏在这里
- 2.Data Extract:好多涉及到数据提取的时候
- 3.Stereogram Solve:立体试图 可以左右控制偏移 可以放张图片试一下就知道这个是什么意思了
- 4.Frame Browser:帧浏览器 主要是对GIF之类的动图进行分解，把动图一帧帧的放
- 5.Image Combiner:拼图，图片拼接（意思显而易见）

直接用



stereogramsolver





Data Extract



