ctfshow-misc入门-31-50

原创

V3geD4g 于 2021-04-04 22:01:08 发布 689 收藏 2

分类专栏： wp 文章标签： python 安全

本文链接：https://blog.csdn.net/xczzhf/article/details/115434194

misc31

老套路，同misc24，算出bmp像素数再算下宽度改了即可，这里是487202/3/150，最后宽度1082，除不尽没事，应该是后面故意多加了几个字节

想了解bmp的可以看看这篇文章https://zhuanlan.zhihu.com/p/260702527

也贴一个最普通的爆破bmp宽度的脚本，爆破完得自己手动找(￣▽￣；)…

```
import struct
import zlib
f = open('./misc31/misc31.bmp','rb')
c = f.read()
width = c[18:22]
height = c[22:26]
# 爆破bmp宽度
for i in range(900,1100):
    f1 = open('./bpout/'+str(i)+'.bmp','wb')
    # print(struct.pack('>i',i)[::-1])
    img = c[:18]+struct.pack('>i',i)[::-1]+c[22:]
    f1.write(img)
    f1.close()
```

misc32

老套路，爆破png宽度，网上随便搜一堆，不讲了，贴个腊鸡脚本

```
import struct
import zlib
for i in range(4096):
    for j in range(4096):
        c = bytes.fromhex('49484452000003840000009608020000000')#ihdr段不包括crc
        ihdr = c[:4]+struct.pack('>i',i)+struct.pack('>i',j)+c[12:]
        crc = 0xE14A4C0B
        if zlib.crc32(ihdr) == crc:
            print(hex(i),hex(j))
            exit(0)
```

misc33

同上，只是把高也改了，crc32还是正确的

```
import struct
import zlib
for i in range(4096):
    for j in range(4096):
        c = bytes.fromhex('49484452000003840000009608020000000')
        ihdr = c[:4]+struct.pack('>i',i)+struct.pack('>i',j)+c[12:]
        crc = 0x5255A798
        if zlib.crc32(ihdr) == crc:
            print(hex(i),hex(j))
            exit(0)
```

misc34

IHDR块的CRC改成错误宽度的crc了，所以上面方法不适用，直接暴力爆破宽度，手动找

```
import struct
import zlib
#爆破png宽度
f = open(r'misc34.png','rb')
c = f.read()
width = c[16:20]
height = c[20:24]
for i in range(900,1200):
    f1 = open('./bpout/'+str(i)+'.png','wb')
    # print(struct.pack('>i',i)[::-1])
    img = c[:16]+struct.pack('>i',i)+c[20:]
    f1.write(img)
    f1.close()
```

misc35

这题先把高度改高一点，这里我改的高度是\x02\x96，如果你不一样，下面脚本记得改，flag在下面，然后去爆破jpg的宽度，原理都差不多

```
import struct
import zlib
#爆破jpg宽度
f = open('./misc35/misc35.jpg','rb')
c = f.read()
# width = c[c.index(b'\x00\x96\x03\x84')+2:c.index(b'\x00\x96\x03\x84')+4]
# height = c[c.index(b'\x00\x96\x03\x84'):c.index(b'\x00\x96\x03\x84')+2]
for i in range(900,1000):
    f1 = open('./bpout/'+str(i)+'.jpg','wb')
    # print(struct.pack('>i',i)[::-1])
    img = c[:c.index(b'\x02\x96\x03\x84')+2]+struct.pack('>h',i)+c[c.index(b'\x02\x96\x03\x84')+4:]
    f1.write(img)
    f1.close()
```

misc36

这题也先把gif的高度拉高，记得修改不止一处，然后再去爆破宽度，当然不用脚本也行，这题给了范围，手动试试就好

```
import struct
import zlib
#爆破gif宽度
f = open('./misc36/misc36.gif','rb')
c = f.read()
# width = c[c.index(b'\x00\x96\x03\x84')+2:c.index(b'\x00\x96\x03\x84')+4]
# height = c[c.index(b'\x00\x96\x03\x84'):c.index(b'\x00\x96\x03\x84')+2]
for i in range(920,950):
    f1 = open('./bpout/'+str(i)+'.gif','wb')
    # print(struct.pack('>i',i)[::-1])
    img = c[:c.index(b'\x00\x00\x84\x03\x96\x02')+2]+struct.pack('>h',i)[::-1]+c[c.index(b'\x00\x00\x84\x03\x96\
x02')+4:]
    f1.write(img)
    f1.close()
```

misc37

在9、14、21、31、34帧里，直接stegsolve看就行

misc38

是一张apng图片，即动态的png图片，用浏览器安装个插件就可以看到动态效果了（好像有些浏览器默认支持），这里推荐用apngdisassembler直接分离每一帧，找找就好了

misc39

gif帧数间隔隐写，kali里用命令`identify -format "%T " misc39.gif > misc39.txt`提取帧数，转01再七位一组转字符即可（为啥是七位就要问套神了□）

```
s = '37 37 36 36 36 37 37 37 37 37 36 37 36 36 37 37 36 36 37 37 36 37 37 37 36 36 37 37 37 37 36 37 36 36 36 37
 37 36 37 37 37 37 37 37 37 36 37 37 37 37 37 37 37 36 37 37 36 37 37 36 37 36 37 36 37 37 36 36 37 36 36 37 37
37 36 36 36 36 37 37 36 36 36 37 36 37 37 36 36 37 36 37 37 36 36 37 37 36 37 37 36 36 37 37 36 36 37 37 37 36 3
6 37 36 37 37 37 36 36 37 36 37 37 36 37 36 37 37 37 36 36 37 37 36 37 37 36 36 36 37 36 36 37 37 36 37 37 37 37
 37 36 36 36 37 36 37 37 36 36 37 36 37 36 37 37 36 36 37 36 36 37 37 36 37 37 36 36 37 37 37 36 36 36 37 37 36
36 37 36 36 36 37 37 37 36 36 37 36 37 37 36 37 37 36 36 37 37 36 36 37 37 37 37 36 36 36 37 36 37 37 37 36 3
6 37 37 37 36 36 37 36 37 37 37 36 36 36 37 36 37 37 36 36 36 37 37 37 37 36 36 36 36 37 36 37 37 36 36 36 36
 37 37 36 37 36 36 36 37 37 36 37 36 37 36 37 37 37 36 36 37 37 37 37 37 37 36 37'
s = s.replace('37','1').replace('36','0').replace(' ','')
# s = s[1:]
print(s)
print(len(s))
for i in range(len(s)//7):
    try:
        print(chr(int(s[i*7:i*7+7],2)),end='')
    except:
        pass
```

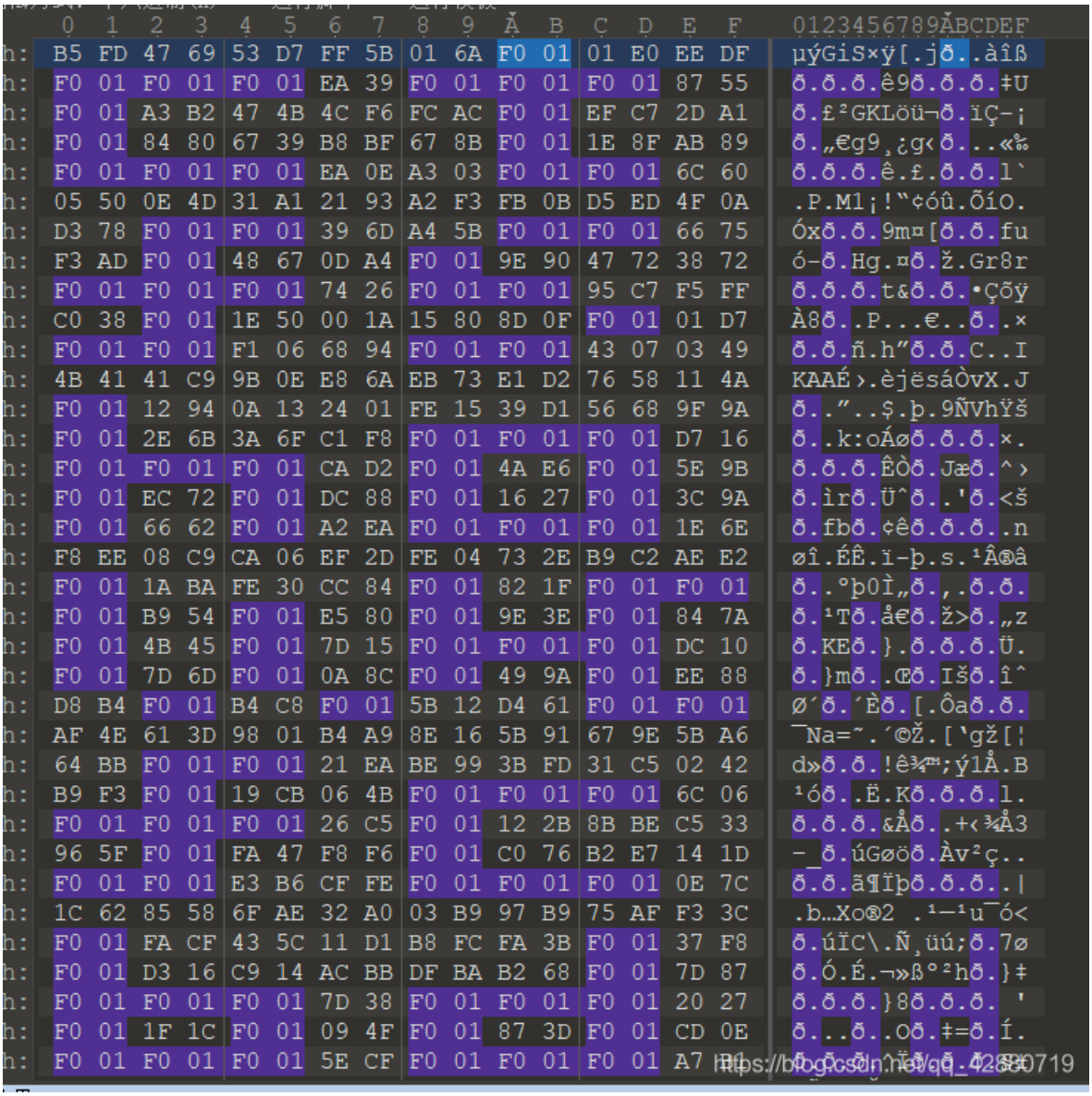
misc40

原理跟上题差不多，只不过换成了apng的帧时间间隔隐写，用apngdisassembler分解的时候会带一个时间间隔的txt，所以脚本很好写，还有前面多了些干扰数据，所以从28个开始

```
#apng时间隐写
for i in range(28,69):
    f = open('./misc40/apngframe'+str(i)+'.txt')
    c =f.read()
    print(chr(int(c.split('/')[0].split('=')[1])),end='')
```

misc41

愚人节专属题目，在010里面直接搜F001看高亮部分即可（不要问为什么，问就是八神师傅脑洞太大了）(直接偷个mumuzi的图)



misc42

IDAT块长度隐写，拖到tweakpng就很容易看出来，当然前面也多了些干扰数据，从99，116开始

```
s = [99,116,102,115,104,111,119,123,48,55,56,99,98,100,48,102,57,99,56,100,51,102,50,49,53,56,101,55,48,53,50,57
,102,56,57,49,51,99,54,53,125]
for i in s:
    print(chr(i),end='')
```

misc43

CRC32隐写，一拖入tweakpng就一堆报错，每块IDAT的CRC32都是错的，使用pngdebugger提取错误的crc，再转hex转字符就好了

```
file-size=4560 bytes

0x00000000      png-signature=0x89504E470D0A1A0A

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


0x00000021      chunk-length=0x00000180 (384)
0x00000025      chunk-type='IDAT'
0x000001A9      crc-code=0xE59387E5
>> (CRC CHECK)  crc-computed=0x8385F691          =>      CRC FAILED


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


0x00000339      chunk-length=0x00000180 (384)
0x0000033D      chunk-type='IDAT'
0x000004C1      crc-code=0x74667368
>> (CRC CHECK)  crc-computed=0x4462C3A1          =>      CRC FAILED


0x000004C5      chunk-length=0x00000180 (384)
0x000004C9      chunk-type='IDAT'
0x0000064D      crc-code=0x6F777B36
>> (CRC CHECK)  crc-computed=0x397611E1          =>      CRC FAILED


0x00000651      chunk-length=0x00000180 (384)
0x00000655      chunk-type='IDAT'
0x000007D9      crc-code=0x65623235
>> (CRC CHECK)  crc-computed=0x4F02AFA2          =>      CRC FAILED
```

```
et    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
000   09 DA D1 61 E5 93 87 E5  93 A6 2E 63 74 66 73 68  .诸a鍝囧摝.ctfsh
010   6F 77 7B 36 65 62 32 35  38 39 66 66 66 66 35 65  ow{6eb2589ffff5e
020   33 39 30 66 65 36 62 38  37 35 30 34 64 62 63 30  390fe6b87504dbc0
030   38 39 32 7D                                       892}
```

misc44

也是一种CRC32隐写，错误的CRC32和正确的CRC32分别代表着01，再8位一组转字符，劝大家不要拖进tweakpng，因为会有几百个弹窗。。。

至于如何提取正确和错误的CRC32，我的做法是用PCRT识别再放入txt，再写个脚本，就比较容易

PCRT命令如下

```
python PCRT.py -y -v -i misc44.png > 123.txt
```

python脚本如下

```
f = open('123.txt')
res = ''
while 1:
    c = f.readline()
    if c:
        if 'chunk crc' in c:
            # print(c)
            res+='0'
        elif 'Correct IDAT CRC' in c:
            res+='1'
    else:
        break

print(res)
print(len(res))
for i in range(len(res)//8):
    a = res[i*8:i*8+8]
    try:
        print(chr(int(a,2)),end='')
    except:
        pass
```

misc45

不会，菜鸡完全看不出来□
4.22更新
问bit师傅问出来的，直接把png转bmp，再binwalk一下就出来了，具体原因的话，就是在bmp的中间0x10000的地方藏了个gzip，不binwalk很难看出来，png直接读像素啥也看出不来是因为bmp的像素是从左下角开始bgr这样读取的，所以要转换为bmp再binwalk，也是题目提示转换格式的意义所在，只能说八神师傅tql！！！

misc46

比较有意思的一个题，用gif每一帧的偏移量作为坐标来画图即可，这里gif的偏移量我是用identify命令直接获取的

```
identify misc46.gif > 233.txt
```

再写个脚本画图即可

```
from PIL import Image
import matplotlib.pyplot as plt
f = open('233.txt')
pp = []
while 1:
    c = f.readline()
    if c:
        s = eval(c.split('+')[1]+','+c.split('+')[2][:2])
        pp.append(s)
        print(s)
        # print(c)
    else:
        break

img = Image.new('RGB',(400,70),(255,255,255))
for i in pp:
    new = Image.new('RGB',(1,1),(0,0,0))
    img.paste(new,i)
plt.imshow(img)
plt.show()
```

misc47

跟上一题差不多，只不过图片格式变成了apng，apng每一帧的偏移提取稍微有点麻烦，我是自己在hex定位后写脚本提取的

```
import struct
from PIL import Image
import matplotlib.pyplot as plt
f = open('misc47.png','rb')
c = f.read()
c = c[c.index(bytes.fromhex('6663544C00000001')):]
pp = []
for i in range(1,1124,2):
    start = c.index(bytes.fromhex('6663544C0000')+struct.pack('>h',i))
    # start = c.index(bytes.fromhex('6663544C000000'+hex(i)[2:]))
    # print(start)
    fc = c[start:start+30]
    print(fc[18:20],fc[22:24])
    print(struct.unpack('>h',fc[18:20])+struct.unpack('>h',fc[22:24]))
    pp.append(struct.unpack('>h',fc[18:20])+struct.unpack('>h',fc[22:24]))
    # print(fc.index(b'\xb6'),fc.index(b'\x34'))
# print(c[:100])
img = Image.new('RGB',(400,70),(255,255,255))
for i in pp:
    new = Image.new('RGB',(1,1),(0,0,0))
    img.paste(new,i)
plt.imshow(img)
plt.show()
```

misc48

脑洞题，问八神师傅才知道的

根据DQT块的提示，叫我们统计FF的数量再减1，一开始我以为是统计所有的，完全没找到思路，后来八神师傅说是统计每两个有意义块之间的FF的数量再减一，因为每两个有意义块之间插入数据好像是不太影响的，至于要减一是因为这段中的最后一个FF是下一个有意义块的开头，取前32个段即可，会发现每个段长度都不超过16，所以直接转为16进制就是flag

[外链图片转存失败,源站可能有防盗链机制,建议将图片保存下来直接上传(img-9c6PslON-1617544812055)(http://blog.v3ged4g.top/2021/04/04/ctfshow-misc%E5%85%A5%E9%97%A8-31-50/misc48.png)]

就是上图种紫色的FF段，取前32段的长度减一转16进制即可（不要忘记第一个），最后统计出来长度如下

```
0 12 11 0 7 10 13 13 9 0 9 13 0 13 6 0 10 9 2 1 0 1 10 8 11 5 12 7 2 2 3 10
```

```
s = '0 12 11 0 7 10 13 13 9 0 9 13 0 13 6 0 10 9 2 1 0 1 10 8 11 5 12 7 2 2 3 10'
d = '0123456789abcdef'
for i in s.split(' '):
    print(d[int(i)],end='')
```

misc49

脑洞题，也是问八神师傅才知道的

拖到010会发现开头有好多FFE?的块，后面还跟着一些奇怪的设备名，一开始我统计了下FFE0-FFEF块的数量，发现刚好32个，然后就没思路了，后来问了下八神师傅，原来E后面的那位就是flag，最后按照这些块的顺序把E后面那位合起来就行了，最后如下

```
0c618671a153f5da3948fdb2a2238e44
```

misc50

直接stegsolve看就有了