

HGAME 2022 Week1 writeup by Halo

HGAME 2022 Week1 writeup by Halo

MISC
摆烂

MISC

摆烂

使用 foremost 分离 bai.zip (binwalk 无法成功分离)

分离后得到一张 png 图片和一个正常的压缩包。在使用 TweakerPNG 查看图片的块时，发现其中有大约一半的块为普通 PNG 数据，另一半如下图。

00000016.png (C:\Users\Halo\Downloads\hgame 2022\week4\bai2\)

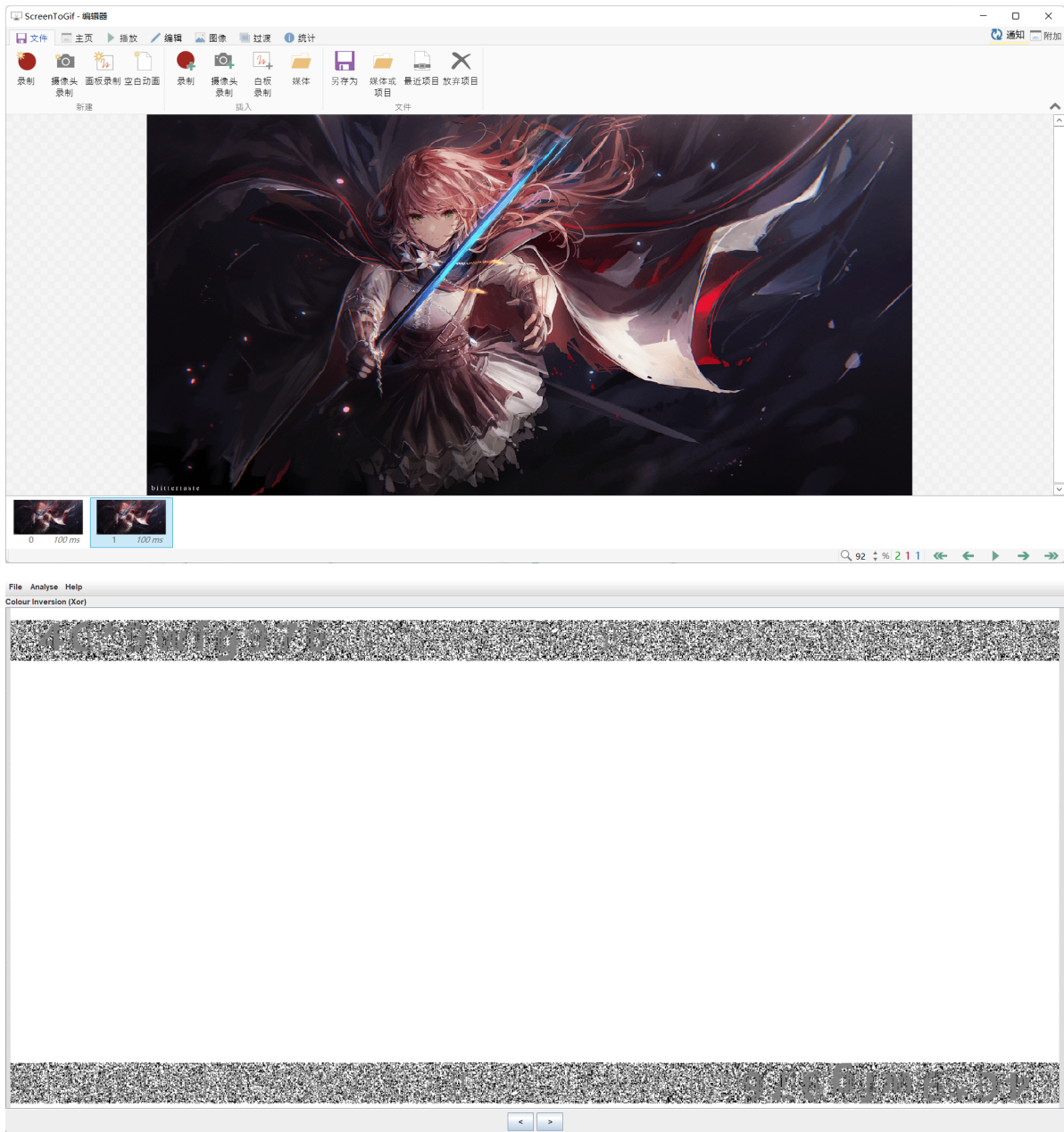
TweakPNG

Chunk	Length	CRC	Attributes	Contents
IDAT	8192	6c6c4...	critical	PNG image data
IDAT	8192	fb3d...	critical	PNG image data
IDAT	8192	8b1d3...	critical	PNG image data
IDAT	8192	8efea2...	critical	PNG image data
IDAT	8192	a88cd...	critical	PNG image data
IDAT	8192	b4e82...	critical	PNG image data
IDAT	8192	b1e79...	critical	PNG image data
IDAT	8192	08fda...	critical	PNG image data
IDAT	5926	e06e1...	critical	PNG image data
fcTL	26	b903f...	ancillary, private, u...	APNG frame control, seq#=1, 1501x748+0+0, delay=0.100s, dispose=none, blend=source
fdAT	8196	fd7fd...	ancillary, private, u...	APNG frame data, seq#=2
fdAT	8196	ebd17...	ancillary, private, u...	APNG frame data, seq#=3
fdAT	8196	8ac0b...	ancillary, private, u...	APNG frame data, seq#=4
fdAT	8196	5e496...	ancillary, private, u...	APNG frame data, seq#=5
fdAT	8196	01f14e...	ancillary, private, u...	APNG frame data, seq#=6
fdAT	8196	2c36b...	ancillary, private, u...	APNG frame data, seq#=7
fdAT	8196	96156f...	ancillary, private, u...	APNG frame data, seq#=8
fdAT	8196	29c0d...	ancillary, private, u...	APNG frame data, seq#=9
fdAT	8196	b9428...	ancillary, private, u...	APNG frame data, seq#=10
fdAT	8196	a16da...	ancillary, private, u...	APNG frame data, seq#=11
fdAT	8196	2431f4...	ancillary, private, u...	APNG frame data, seq#=12
fdAT	8196	f4a17c...	ancillary, private, u...	APNG frame data, seq#=13
fdAT	8196	7cbe4...	ancillary, private, u...	APNG frame data, seq#=14
fdAT	8196	bd3a5...	ancillary, private, u...	APNG frame data, seq#=15
fdAT	8196	c31a5...	ancillary, private, u...	APNG frame data, seq#=16
fdAT	8196	c744c...	ancillary, private, u...	APNG frame data, seq#=17
fdAT	8196	5698b...	ancillary, private, u...	APNG frame data, seq#=18
fdAT	8196	f4174c...	ancillary, private, u...	APNG frame data, seq#=19

PNG file size: 3563832 bytes

搜索后得知 APNG 格式。APNG 是一种动图格式，向下兼容普通 PNG。大致格式为：第一帧为普通 PNG 格式，后续帧为带有特殊标识的 PNG，即 APNG。

使用 screenToGif 软件分离出两张 PNG 图片。两张图肉眼看似相同，实则有细微的不同。据去年 hgame 的经验，使用 盲水印 提取出图中的水印。



(调整图片参数让字符更好认一些)

得到的字符串为压缩包密码。解压得到四张图片。拼成一张大的二维码。



CTF.png



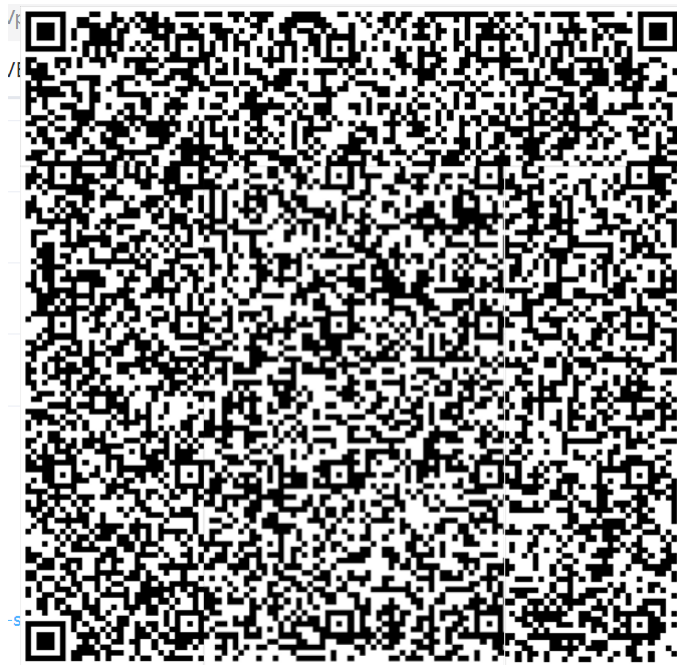
啊.png



好.png



难.png



之后卡了好久好久。在尝试获取这个二维码的版本信息之类时，发现有些扫码软件会报错。又研究了一阵，发现扫出来的结果中包含一些 `unicode` 无宽度字符。

找了一个python的代码扫码，成功得到包含无宽字符的结果。

```
#_*_coding:utf-8*_  
from pyzbar import pyzbar  
import matplotlib.pyplot as plt  
import cv2  
import re  
  
def save_to_file(file_name, contents):  
    fh = open(file_name, 'w', encoding='utf-8')  
    fh.write(contents)  
    fh.close()  
  
#条形码定位及识别  
def decode(image, barcodes):  
    #循环监测条形码  
    for barcode in barcodes:  
        #提取条形码边界框位置  
        #画出图中条形码的边界框  
        (x,y,w,h)=barcode.rect#获得这个图吗的x,y坐标和宽和高区域  
        cv2.rectangle(image, (x,y), (x+w,y+h), (255,0,0), 5)#把它框起来用蓝色，线粗5  
  
        #条形码数据为字节对象，所以如果想在输出图像上  
        #画出来，就需要先将它转换为字符串  
        barcodeData=barcode.data.decode("utf-8")#将barcode的数据识别出来  
        barcodeType=barcode.type#类型也直接识别出来了  
  
        #绘制出图像上条形码的数据和条形码的类型  
        text="{ } ({} )".format(barcodeData , barcodeType)  
        cv2.putText(image, text, (x,y-10), cv2.FONT_HERSHEY_SCRIPT_SIMPLEX, 8,  
        (255,0,0), 2) # cv2.putText(image, text, (x,y-10)  
        #像终端打印条形码数据和条形码类型  
        # 去除可见字符，实际可以不用  
        # barcodeData=re.sub('[\u4e00-\u9fa5]', '', barcodeData)
```

```
# barcodeData=barcodeData.replace(' ', '').replace('。',  
'').replace('CTF', '').replace('.', '').replace(' ', '')  
print("{}\n".format(barcodeData))  
save_to_file('QRdata.txt', barcodeData)  
#plt.figure(figsize=(10,10))  
#plt.imshow(image)  
plt.show()  
  
#二维码  
image=cv2.imread('Snipaste_2022-02-11_00-05-01.png')  
bacodes=pyzbar.decode(image)  
decode(image,bacodes)
```

使用该网站得到隐写内容，获得flag。(就这个我找了好久)

<https://www.mzy0.com/ctftools/zerowidth1/>