## 实验吧关于隐写术的writeUp(二)



## 0x01 Black Hole

1.下载文件后,发现打不开,放到kali中。用命令file分析一下文件

root@trial:~/Documents# file blackhole.img blackhole.img: Linux rev 1.0 ext3 filesystem data, UUID=d85407b9-7cfc-4443-8464-dfe46da616d6 (errors)

可以发现这是一个linux ext3系统文件,我们创建一个目录,挂载文件

root@trial:~/Documents# mkdir blackhole
root@trial:~/Documents# mount blackhole.img blackhole

2.从文件名为???的文件夹中获得一张图片,图片打不开。

3.用hex workshop打开,发现文件头很奇怪,文件尾也很奇怪。

4.然后看其他人的才明白要进行异或运算

Masked: 76 88 B1 A7 0D 1A 50 4C PNG: 89 50 4E 47 0D 0A 1A 0A XOR: FF D8 FF E0 00 10 4A 46

5.异或后再文件末尾发现了这八个字节。至于后面的为什么要用最后64字节进行异或没有想通。

用python写个脚本,就得到了正确的图片

```
mask_in = open('masked_key.png','rb')
key = bytearray(mask_in.read(983040-64))
mask= bytearray(mask_in.read(64))
for x in range(983040-64):
    key[x] ^= mask[x%64]
key_out = open('unmasked_key.png','wb')
key_out.write(key)
key_out.close()
```



## 0x02 stegas 250

- 1.提示说警告信息就藏在图盘中,警告信息的MD5就是flag
- 2.用stegsolve分析,在蓝色图层0通道发现左上角有东西



## 3,数据提取

<u>\$</u>	- 0
Extract Preview	
6d6f6465726e7761 7266617265000000 modernwa rfare.	
00000000000000 00000000000000	
00000000000000 00000000000000	
00000000000000 00000000000000	
00000000000000 000000000000000000000000	••
00000000000000 00000000000000	••
00000000000000 000000000000000000000000	
	••
	••
Bit Planes	Order settings
Alpha 7 6 5 4 3 2 1 0	Extract By 🖲 Row 🔾 Column
Red 7 6 5 4 3 2 1 0	Bit Order (  MSB First   LSB First
Green 7 6 2 5 4 3 2 1 0	
	Bit Plane Order
Blue 7 6 5 4 3 2 1 0	🔍 RGB 🛛 🔾 GRB
	0.000 0.000
	○ RBG ○ BRG
Preview Settings	
Include Hex Dump In Preview 🔽	
Provine Save Text Save Pin Cancel	

4.然后MD5就得到flag了

转载于:https://www.cnblogs.com/Triomphe/p/7668231.html