

攻防世界Misc新手练习区WriteUp大全

原创

D-R0s1 于 2019-09-02 14:24:26 发布 3205 收藏 12

分类专栏: [CTF WriteUp](#) 文章标签: [攻防世界](#) [misc](#) [ctfwriteup](#)

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/CliffordR/article/details/100187010>

版权



[CTF WriteUp](#) 专栏收录该内容

28 篇文章 3 订阅

订阅专栏

文章目录

写这篇博客的目的

解题过程

1. this is flag

2.ext3

3.give_you_flag

4.pdf

5.SimpleRAR

6.坚持60s

7.gif

8.掀桌子

9.如来十三掌

10.base64stego

11.功夫再高也怕菜刀

12.stego

总结

写这篇博客的目的

对于CTF中的Misc来说, 做题经验显得十分重要, 而做题经验的获得很大一部分取决于刷题量。为了避免大家在刷题过程到处搜WriteUp浪费时间, 现在把我的一些做题方法分享出来, 希望对大家有帮助。当然, 大家有更好的解决方法欢迎在评论区留言, 互相学习, 共同进步。

解题过程

1. this is flag

点开这个题目直接显示flag

2.ext3

a.把文件放进虚拟机，使用命令

```
strings linux | grep flag
```

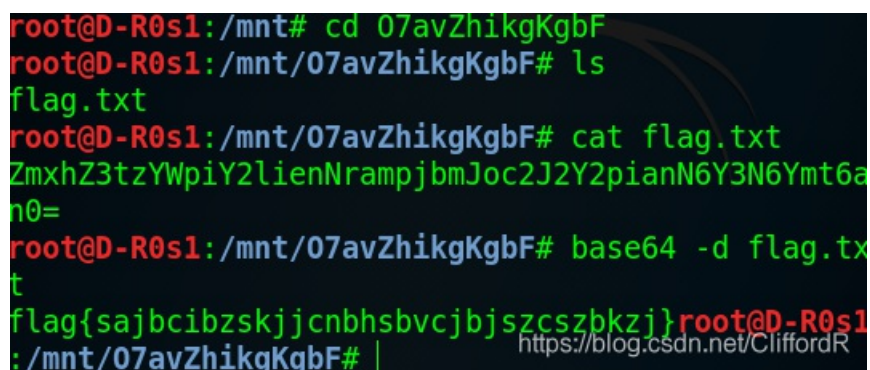
在linux这个文件中搜索flag字符串



发现存在flag.txt文件，ext3文件是一种Linux日志文件，所以把它挂载到linux系统上
把这个文件挂载到mnt目录，然后进入此目录，使用ls列出所有文件，查看flag

```
mount linux /mnt
cd /mnt
ls
cd 07avZhikgbf/
cat flag.txt
```

linux操作截图



b.得到flag.txt中是base64编码的。使用

```
base64 -d flag.txt
```

解码

3.give_you_flag

- a.附件为一个gif
- b.放入stegsolve中一帧一帧的查看一下（stegsolve中analyse下的frame browser）
- c.发现二维码



4.pdf

- a.根据提示说图片下面什么都没有，怀疑图片下面隐藏着另一个图层
- b.把pdf放进linux中，隐藏的部分鼠标会有变化，把它选中
- c.flag就出现了



5.SimpleRAR

- a.得到压缩包后打开得知压缩包内用文件头损坏

simp.rar (评估版本)

文件(F) 命令(C) 工具(S) 收藏夹(O) 选项(N) 帮助(H)

添加 解压到 测试 查看 删除 查找 向导 信息 扫描病毒 注释 保护 自解压格式

↑ simp.rar - RAR 4.x 压缩文件, 解包大小为 16 字节

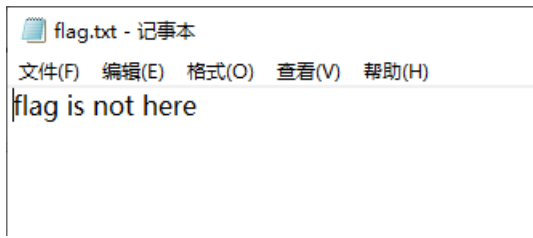
名称	大小	压缩后大小	类型	修改时间	CRC32
..			文件夹		
flag.txt	16	16	文本文档	2017/10/14 23:27	366788C7

WinRAR:诊断信息

信息	压缩文件
文件头已损坏: secret.png	simp.rar (F:\ctf文件\simp.rar)

的是一个图片，并且发现里面还有一个flag.txt

b.打开这个txt



并没有搞头，目光聚集到这个损坏的png上

c.查资料得知被压缩后的png头为A8 3C 74，在winhex中查找一下这个png头，看看有没有改变

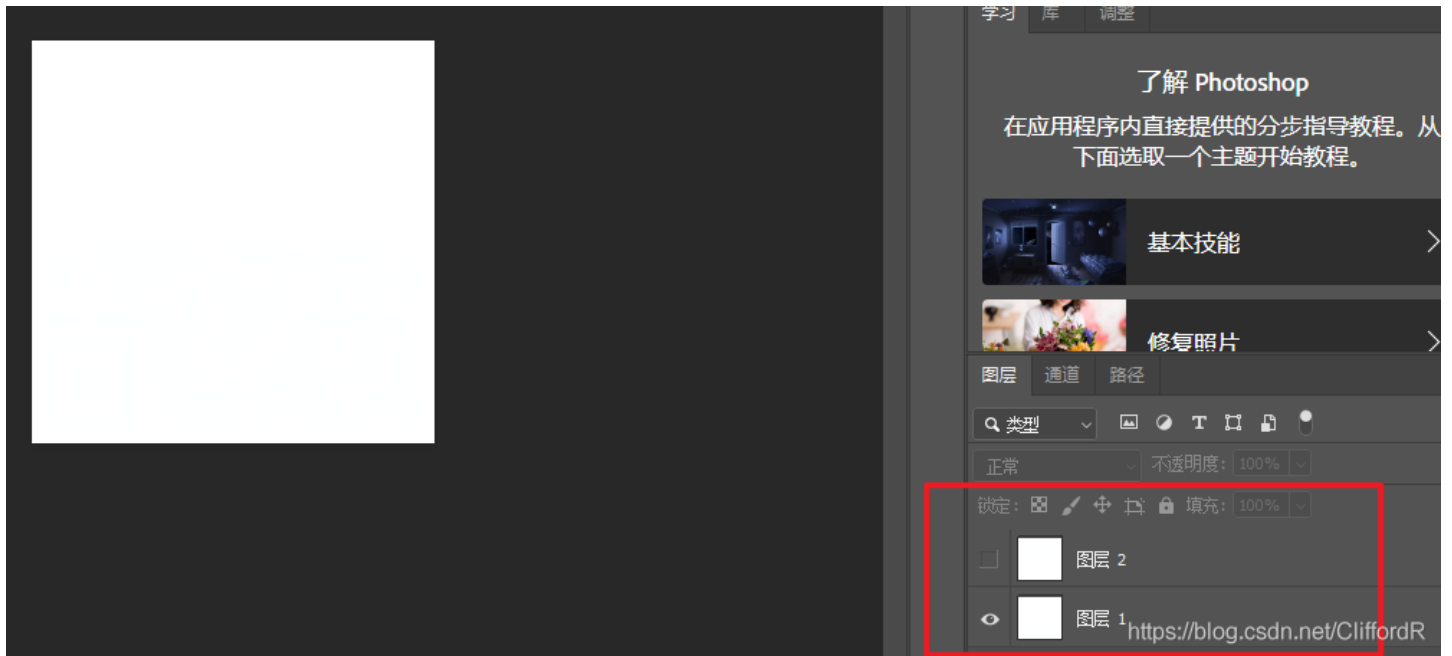
d.发现压缩包中文件头是A8 3C 7A

```
:0 43 66 6C 61 67 20 69 73 20 6E
:0 65 38 3C 7A 20 90 2F 00 3A 15
:0 02 BC E9 8C 2F 6E 84 4F 4B 1D
:0 00 73 65 63 72 65 74 2F 70 6E
```

e.修改7A为74。

f.成功解压出图片，但是发现是一个白色的图片。根据提示是ps双图层，把它放入ps中分离图层，后缀为png但是拖入ps的过程中被提示这不是一个png文件。在2345看图王中打开发现存在两个帧，那么这就是一个gif了，改后缀为gif

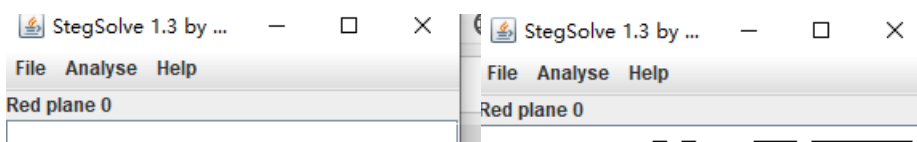
g.然后把它拖入ps中

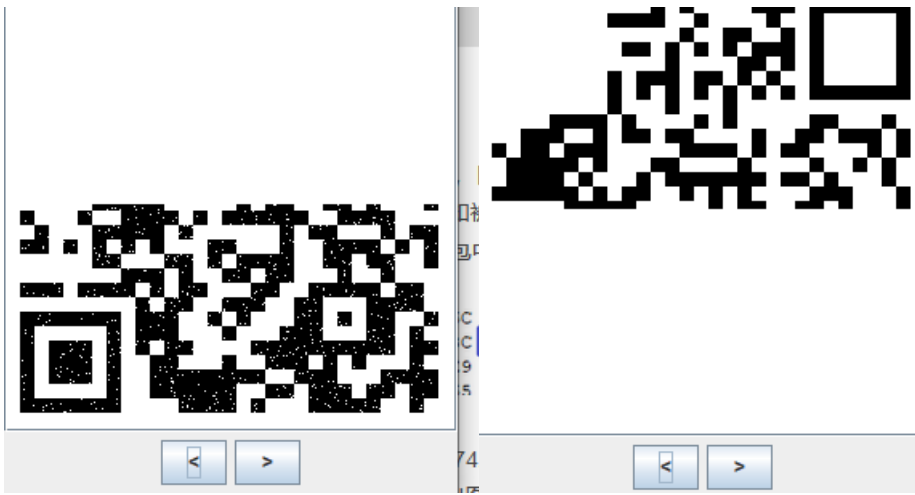


把这两个图层分别保存

f.保存后把这个纯白的图片放入stegsolve中。发现两个图中都可以发现半个二维码，剩下的工作就是把两个二维码拼接成一个二维码

g.





但是第二块二维码定位点缺失，只要第一个二维码中的定位点补过去就可以
f.补图+扫描



6. 坚持60s

a.下载下来是一个.jar

b.看文件类型，这就需要看出这个程序的基本的对象，就要用到java反编译器，这里使用jd-gui

c.



7.gif

a.下载附件后打开发现是一些黑白图片

b.一共104块， $104 \div 8 = 13$ ，也就刚好是13个八位的二进制数

c.白色为0，黑色为1，依靠手输肯定是行不通的。附上一个python脚本

```
from PIL import Image
flag_dic=""
for i in range(0,104):
    img = Image.open("F:\\ctf文件\\b3ba561080fb4a9d9c1f019e298e218b\\gif\\%d.jpg"%i)
    clr = img.getcolors() #clr 包含 [("num of occurrences", "color"), ...]
    if clr == [(46656, (12, 12, 0))]:
        flag_dic += "1"
    else:
        flag_dic += "0"
print(flag_dic)

flag = ""
for q in range(int(len(flag_dic)/8)):
    flag += chr(int(flag_dic[q*8:(q+1)*8],2))
print(flag)
```

脚本的话是自己写的，比较菜，师傅们可以交流一下，咱们共同进步。代码逻辑比较简单，有疑问的可以评论区留言。利用这个python脚本可以直接获得flag。

原理就是黑色代表1，白色代表0，然后是8个为一组的二进制。转换为字符串就是flag了。

```
1 from PIL import Image
2 flag_dic=""
3 for i in range(0,104):
4     img = Image.open("F:\ctf文件\b3ba561080fb4a9d9c1f019e298e218b\gif\%d.jpg"%i)
5     clr = img.getcolors() #clr 包含 [("num of occurrences", "color"),...]
6     if clr == [(46656, (12, 12, 0))]:
7         flag_dic += "1"
8     else:
9         flag_dic += "0"
10    print(flag_dic)
11
12    flag = ""
13    for q in range(int(len(flag_dic)/8)):
14        flag += chr(int(flag_dic[q*8:(q+1)*8],2))
15    print(flag)
16
17
18
```

识别白色黑色

```
E:\网络安全脚本\venv\Scripts\python.exe E:/网络安全脚本/识别白色黑色.py
011001100110110001100001011001110111011010001100111010100111001011110110011101101001010001100111101
flag {FuN_giF}
Process finished with exit code 0
```

<https://blog.csdn.net/CliffordR>

8.掀桌子

- a.初步判定是十六进制直接转换
- b.发现不行
- c.将ascii码减去128转换

```
string="c8e9aca0c6f2e5f3e8c4efe7a1a0d4e8e5a0e6ece1e7a0e9f3baa0e8eafae3f9e4eafae2eae4e3eaebfaebe3f5e7e9f3e4e3e8eaf9eaf3e2e4e6f2"
flag=""
for i in range(0,len(string),2):
    s = "0x" + string[i] + string[i+1]
    flag += chr(int(s,16) - 128)
print(flag)
```

9.如来十三掌

- a.看字符直接与佛论禅
- b.得到这一串
MzkuM3gvMUAwnzuvn3cgozMIMTuvqzAenJchMUAeqzWenzEmLJW9
- c.由如来十三掌想到rot13，然后进行rot13解码，最后base64解码得到flag

10.base64stego

这个题目说简单也不简单，说难也不难，我对这个题目的理解还没有到达分享的地步，在这里分享一个师傅的博客。

在这里只把python的脚本贴出来做一个保存。

```
import base64

b64chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'
with open('F:\\ctf文件\\rebuilt.787798f51ce441dd9961405c1aff8893\\stego.txt', 'rb') as f:
    flag = ''
    bin_str = ''
    for line in f.readlines():
        stegb64 = str(line, "utf-8").strip("\n")
        rowb64 = str(base64.b64encode(base64.b64decode(stegb64)), "utf-8").strip("\n")
        offset = abs(b64chars.index(stegb64.replace('=', '')[-1]) - b64chars.index(rowb64.replace('=', '')[-1]))
        equalnum = stegb64.count('=') # no equalnum no offset
        if equalnum:
            bin_str += bin(offset)[2:].zfill(equalnum * 2)
            # flag += chr(int(bin(offset)[2:].zfill(equalnum * 2), 2))
            # print(flag) 这样写得不出正确结果
    print([chr(int(bin_str[i:i + 8], 2)) for i in range(0, len(bin_str), 8)])
```

11.功夫再高也怕菜刀

a.在流量包中搜索flag

b.追踪几个tcp数据流后发现存在FFD8/FFD9，这是png图片文件的开头和结尾

The screenshot shows Wireshark interface with a search filter 'flag' applied. The packet list pane shows several TCP and HTTP packets. The packet details pane for packet 1144 shows an application/javascript content type. The packet bytes pane shows a hex dump and ASCII representation of the data, including the text 'text/html' and a URL 'http://192.168.25.128:80/1144/1148/'.

c.把FFD8和FFD9之间的数据复制写到winhex中，注意以十六进制数据写入，然后改为png打开



12.stego

a.在google上安装插件PDF Viewer

b.控制台输入document.documentElement.textContent，得到一串AB编码而成的字符串，将A变为.，B变为-，摩斯解密为CONGRATULATIONSnullFLAGnull1NV151BL3M3554G3

c.即flag为1NV151BL3M3554G3

总结

本文优点仁者见仁智者见智

本文缺点：

本文缺点之一就是单纯的就题论题，并没有做一些拓展和归纳总结。

希望在以后自己题量上去以后好好地分类总结一下，更加深入的理解每一种解题方法和思路。