攻防世界Misc高手进阶区第一页WriteUp

原创

D-R0s1 于 2019-09-04 17:03:48 发布 2878 收藏 3

分类专栏： CTF WriteUp 文章标签： 攻防世界 ctf 杂项

本文链接：https://blog.csdn.net/CliffordR/article/details/100541301

文章目录

0x01 写这篇博客的目的

在上一篇杂项新手进阶区的WriteUp写完以后，自己也觉得起到了一个复习强化的作用。只有把简单的事情做好做精重复做才会熟能生巧，刷题–经验–在经验基础上继续刷题–继续获得经验----------形成自己的一套风格。同样，本着互联网分享的精神把自己的解题思路和解题步骤都记录下来，方便大家查看，节约时间，提高效率。师傅们有更好的解题思路也欢迎在评论区留言，咱们一起交流，共同进步。

0x02 解题步骤

1.easycap

a.拿到手的是一个流量包
b.Ctrl+F搜索flag，然后右键追踪流就会发现flag

2.Avatar

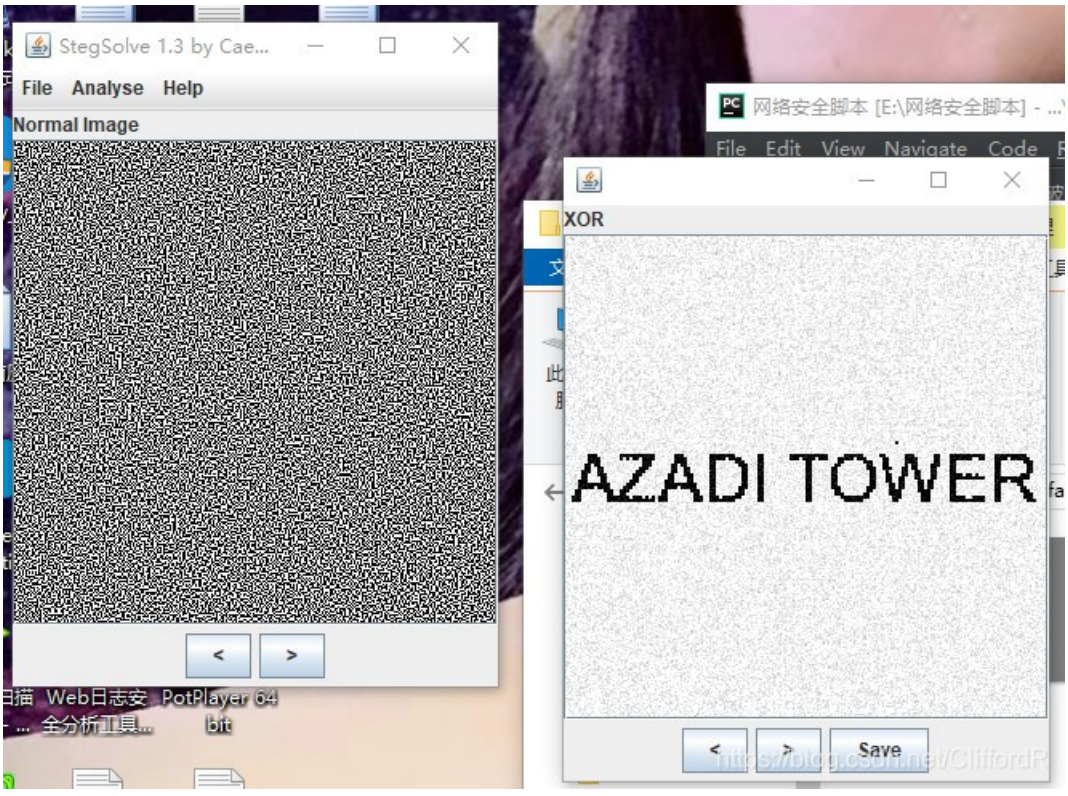
a.到手是一张照片
b.经过一番属性、winhex等工具查看没有什么异常
c.使用outguess工具可以分离出信息
d.

```
outguess -r '/root/ctf/a21da602f6674d79b10018930d61a558.jpg'  -t 123.txt
```

e.本题也就考察outguess工具的使用

3.What-is-this

a.到手是一个没有后缀的文件，首先就要搞清楚这是一个什么文件
b.把它放到kali的虚拟机中就会自动的识别这个是个什么文件
c.放进虚拟机中发现是一个压缩包，然后回到win10中解压，然后又是个没有后缀的文件，同样的方法，最后解压出来是两张差不多的照片
d.看见两张差不多的照片我就习惯先把他放进stegsolve中比较一下



4.签到题

a.Z2dRQGdRMWZxaDBvaHRqcHRfc3d7Z2ZoZ3MjfQ== base64解密ggQ@gQ1fqh0ohtjpt_sw{gfhgs#}
b.花括号的位置不对，栅栏解密试试ggqht{ggQht_gsQ10jsf#@fopwh}，这是第四栏，括号前面五位应该这就是flag的格式
c.凯撒解密ssctf{ssCtf_seC10ver#@rabit}得到flag

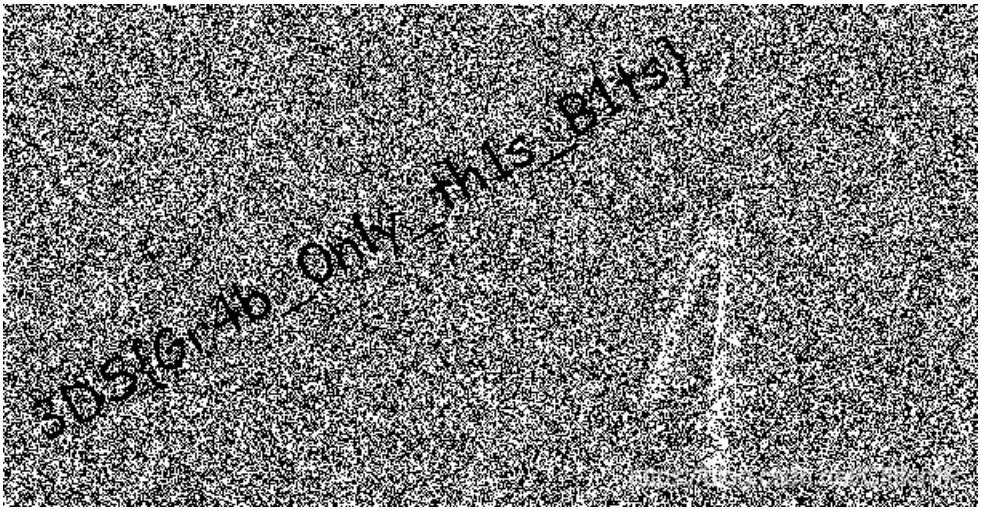
5.Training-Stegano-1

a.notepad++打开直接出现flag

```
: passwd:steganoI
```

6.Excaliflag

stegsolve打开出现flag



7.get-the-key.txt

a.这个直接非常规解法在notepad++打开就能发现

```
NUL NUL NUL NUL NUL NUL NUL NUL NUL
NUL NUL NUL NUL NUL NUL NUL NUL NUL
NULSECCON{kdt#RUd"KWsvMxT`.'8h
```

8.glance-50

这个题目在之前发表过博客

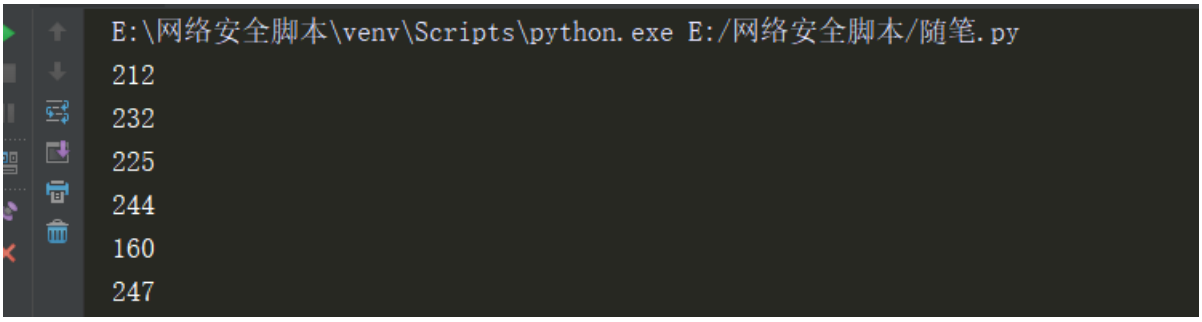
9. 4-2

这个题目是一个词频分析的题目，考点就是词频分析，也是比较基础，但是考虑到CTF竞赛中不让连接外网的情况，想找一个离线的也能解决此类问题的的工具或者方法。还希望师傅们不吝赐教。
这个题目在线解决的话可以再这个网站进行词频分析https://quipqiup.com/

10. misc1

a.一串字符最大的是由0~9和a ~ f，首先想到是十六进制转换
b.常规的十六进制转换不成功
c.把十六进制转换成十进制看一下
这里贴出一个脚本

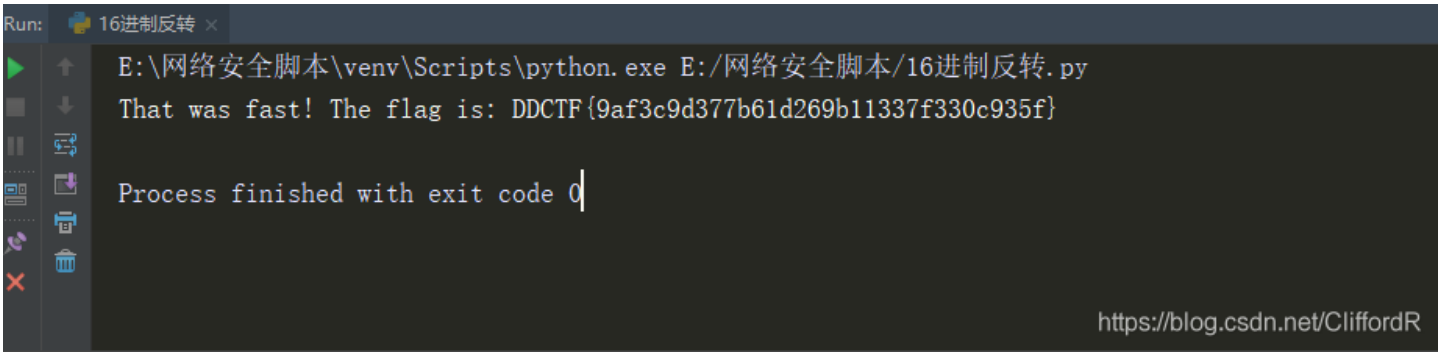
```
s = "d4e8e1f4a0f7e1f3a0e6e1f3f4a1a0d4e8e5a0e6ece1e7a0e9f3baa0c4c4c3d4c6fbb9e1e6b3e3b9e4b3b7b7e2b6b1e4b2b6b9e2b1b1b3b3b7e6b3b3b0e3b9b3b5e6fd"
for i in range(int(len(s)/2)):
   print(int(s[(i*2):(i+1)*2],16))
```



d.发现212,232之类的数字，ascii码一共是128位，这个很明显超过了最大限度。把所有二进制减128试一下
e.在这里贴出脚本

```
string="d4e8e1f4a0f7e1f3a0e6e1f3f4a1a0d4e8e5a0e6ece1e7a0e9f3baa0c4c4c3d4c6fbb9e1e6b3e3b9e4b3b7b7e2b6b1e4b2b6b9e2b1b1b3b3b7e6b3b3b0e3b9b3b5e6fd"
flag=""
for i in range (0,len(string),2):
    s = "0x" + string[i] + string[i+1]
    flag += chr(int(s,16) -128)
print(flag)
```

这个脚本是把十六进制转换为十进制然后减128在把它转化位ascii码
得到结果

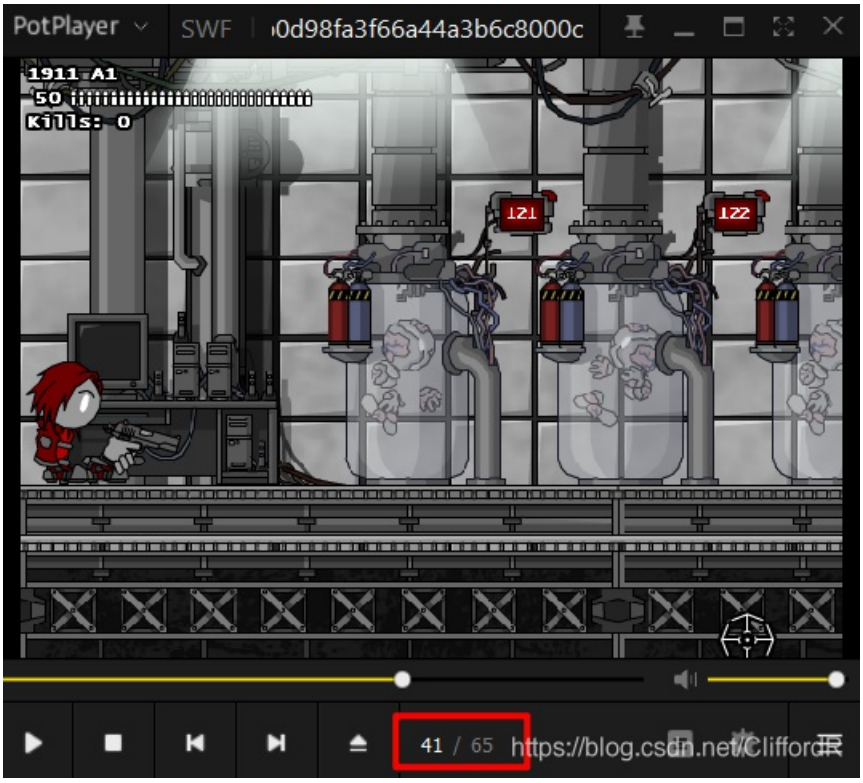


11.embarrass

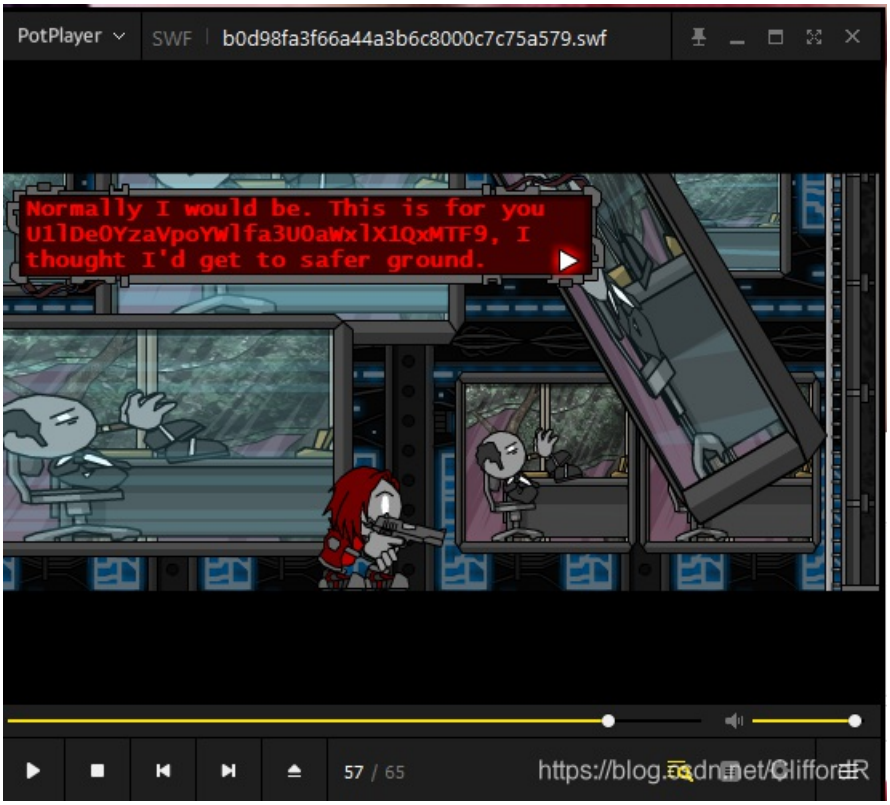
a.这个题目下载下来是一个流量包，一般来说流量包要放入wireshark中进行流量分析
b.我一般习惯先搜索一下他的字符串，说不定啥时候有神奇的收获
c.在kali中进行字符串的搜索
strings ‘/var/run/vmblock-fuse/blockdir/CAkzoC/misc_02.pcapng’ | grep flag
d.结果

12.肥宅快乐题

a.这个题目下载到手是一个.swf后缀的文件
b.用到一个特殊的工具，叫做potplayer
c.把这个.swf文件使用这个工具打开
d.根据提示注意对话就好
e.点击这里可以一步一步往下走



他自己会播放，不用自己打



在第57帧的时候发现一串base64字符，解密base64即可

13.Cephalopod

a.拿到手是一个流量包，习惯性先把他扔进kali中分析一下字符串

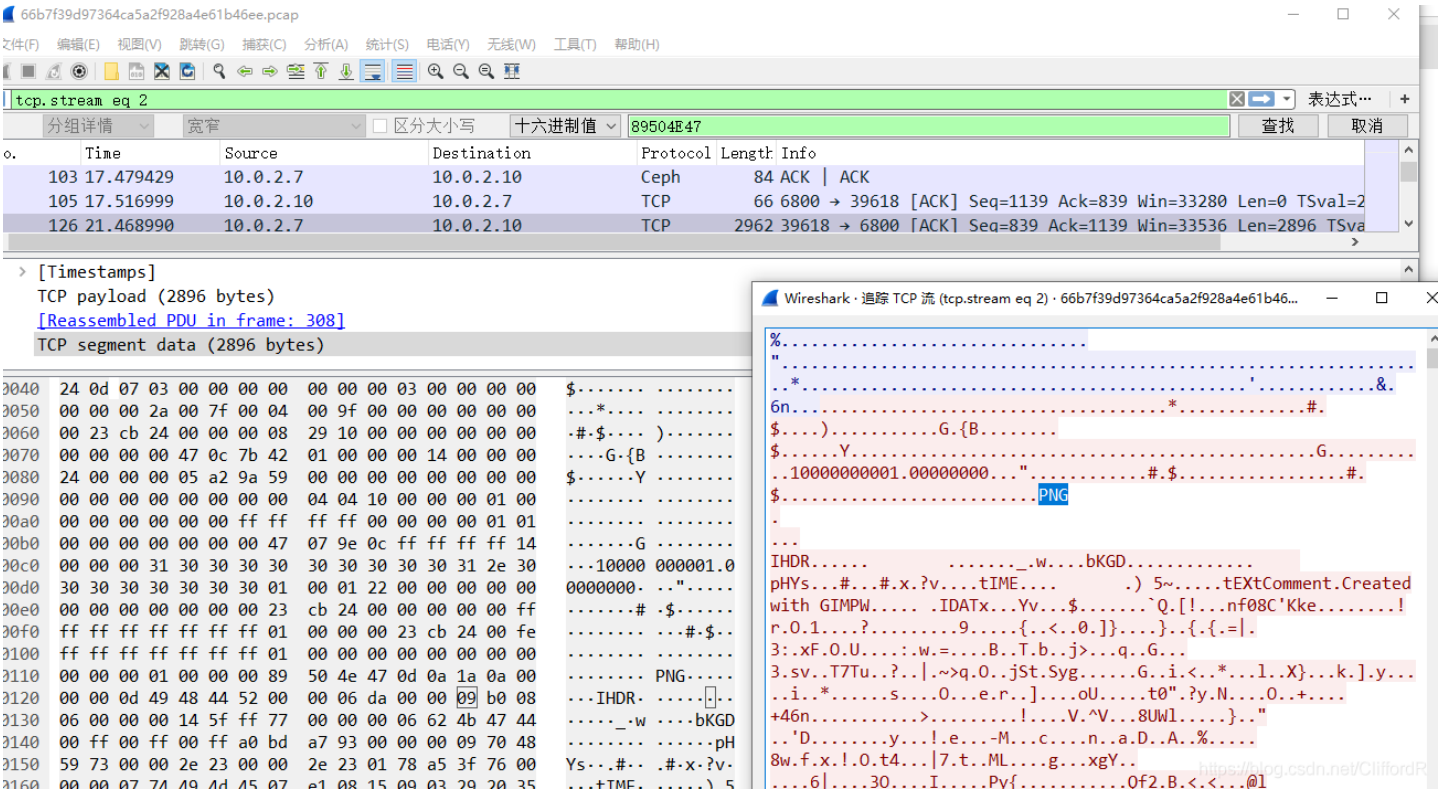
```
root@D-R0s1:~# strings '/var/run/vmblock-fuse/blockdir/aBxzE0/66b7f39d97364ca5a2
f928a4e61b46ee.pcap'  | grep flag
flag.png
flag.png
flag.png
```

b.

发现其中存在flag.png，那么下一步的任务就是先把这个png给分离出来

c.为了查找更加快速，这里我并没有去搜索flag的字符串而是去搜索png的十六进制文件头，89504E47

d.定位到了存在png文件头的数据流，然后追踪一下



e.为了方便提取，把数据流转换成原始数据，这样根据png的文件头就可以比较精确的分离出png

f.把提取出的数据粘贴到winhex中注意以ascii-hex粘贴进去然后保存改后缀为.png就可以了

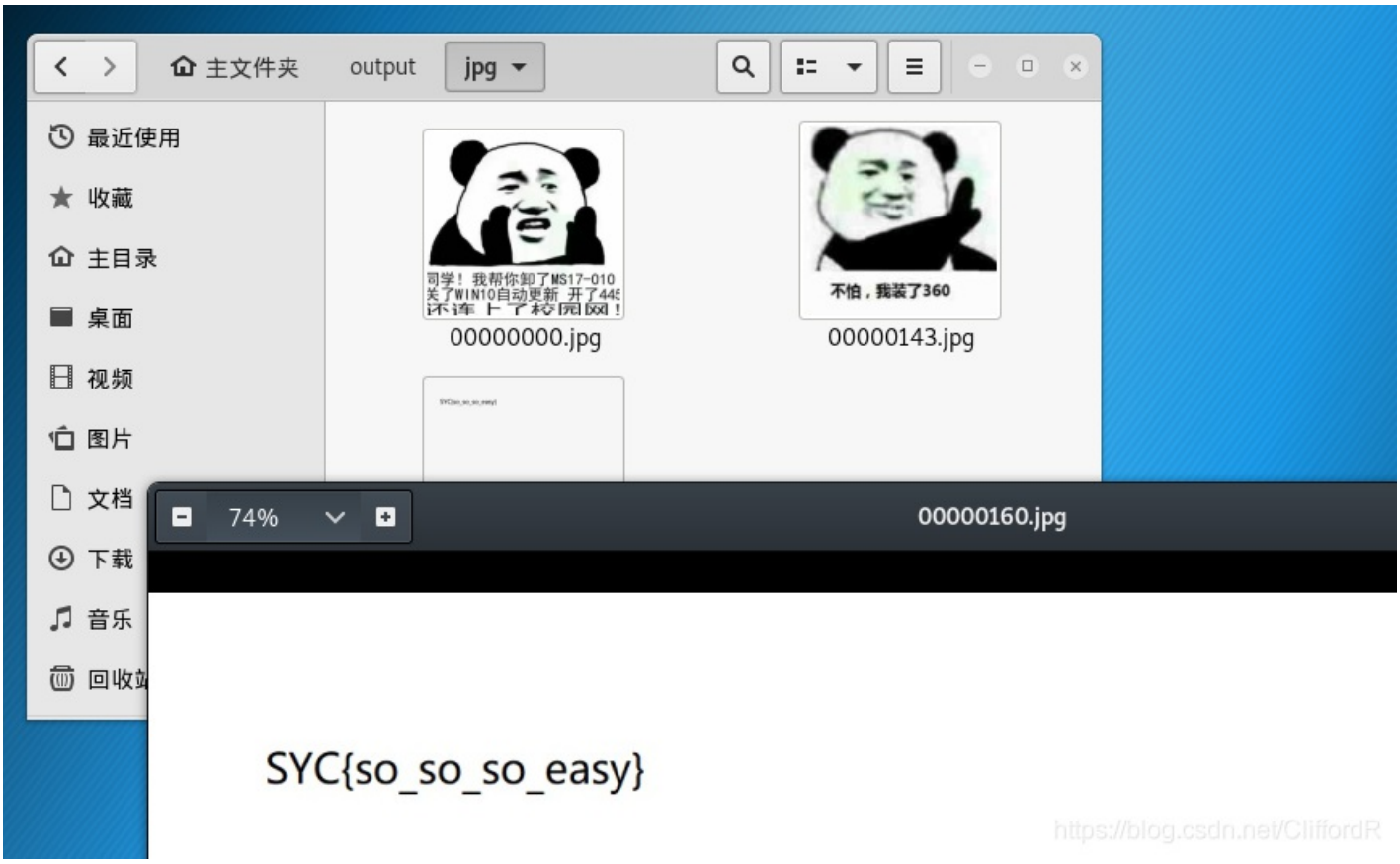
14.小小的pdf

a.放入kali中binwalk一下

```
root@D-R0s1:~# binwalk '/root/ctf/7e5ab2e7587d4a4abf9c705dfb935a92.pdf'

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PDF document, version: "1.4"
452           0x1C4           JPEG image data, JFIF standard 1.01
73254         0x11E26         JPEG image data, JFIF standard 1.01
81606         0x13EC6         Zlib compressed data, default compression
82150         0x140E6         JPEG image data, JFIF standard 1.01
104469        0x19815         Zlib compressed data, default compression
105134        0x19AAE         Zlib compressed data, default compression
```


b.foremost一下，把里面东西分离出来



15.hit-the-core

a.使用strings 查看一下字符串，找到其中这一段代码
b.看到A L E之类的而且每个字母之间隔了4个字母

```
cvqAeqacLtqazEigwiXobxrCrtuiTzahfFreqc{bnjrKwgk83kgd43j85ePgb_e_rw
qr7fvbmHjklo3tews_hmkogooyf0vbnk0ii87Drfgh_n kiwutfb0ghk9ro987k5tf
b_hjiouo087ptfcv}
```

c.写个脚本提取一下

```
data = 'cvqAeqacLtqazEigwiXobxrCrtuiTzahfFreqc{bnjrKwgk83kgd43j85ePgb_e_rwqr7fvbmHjklo3tews_hmkogooyf0vbnk0ii87D
rfgh_n kiwutfb0ghk9ro987k5tfb_hjiouo087ptfcv}'
flag = ''
for i in range(3, len(data), 5):
    flag += data[i]

print(flag)
```

16.pure_color

a.在winhex中没有发现什么收获，在kali中使用binwalk分析一下没有什么东西
b.使用stegsolve得到flag

17.2-1

a.下载到手是一个图片，然后打开的时候提示图片错误，扔进winhex中查看一下
b.发现文件头是错误的，修改文件头为89 50 （png文件头89504E47）

t	0	1	2	3	4	5	6	7
00	89	50	4E	47	0D	0A	1A	0A
10	00	00	00	00	00	00	02	F8

c.修改以后也是照片无法打开，然后发现这里

00	89	50	4E	47	0D	0A	1A	0A	00
10	00	00	00	00	00	00	02	F8	08
20	6B	00	00	00	04	67	41	4D	41

也就是代表图片的宽度是0
d.先详细解释一下png的文件头：

- （固定）八个字节89 50 4E 47 0D 0A 1A 0A为png的文件头
- （固定）四个字节00 00 00 0D（即为十进制的13）代表数据块的长度为13
- （固定）四个字节49 48 44 52（即为ASCII码的IHDR）是文件头数据块的标示（IDCH）
- （可变）13位数据块（IHDR)
  - 前四个字节代表该图片的宽
  - 后四个字节代表该图片的高
  - 后五个字节依次为：
    Bit depth、ColorType、Compression method、Filter method、Interlace method
- （可变）剩余四字节为该png的CRC检验码，由从IDCH到IHDR的十七位字节进行crc计算得到。
  也就是说我们可以通过爆破来得到高度和宽度

e.使用tweakpng这个工具可以直接得到校验码

f.使用脚本

```
import struct
import binascii
import os

m = open("misc4.png","rb").read()
for i in range(1024):
    c = m[12:16] + struct.pack('>i', i) + m[20:29]
    crc = binascii.crc32(c) & 0xffffffff
    if crc == 0x932f8a6b: #自己根据情况改
        print(i)
```

得到宽度为709，修改宽度为709得flag