Hgame 2024 week 1 wp

Si1ence

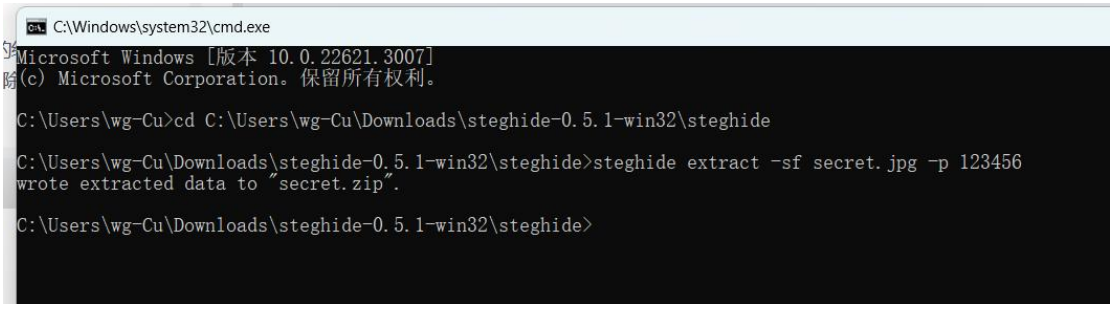
Misc

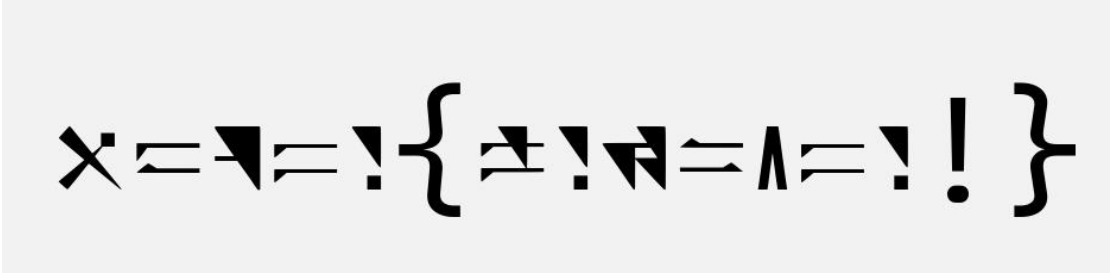
1. SignIn

修改图片尺寸，把宽拉大。

2. 来自星尘的问候

题目描述里提到了六位弱加密，应该就是用的 steghide 用六位密码搞的隐写，所以放进去分离一下，爆破密码之前试了 123456 结果直接就成功了。



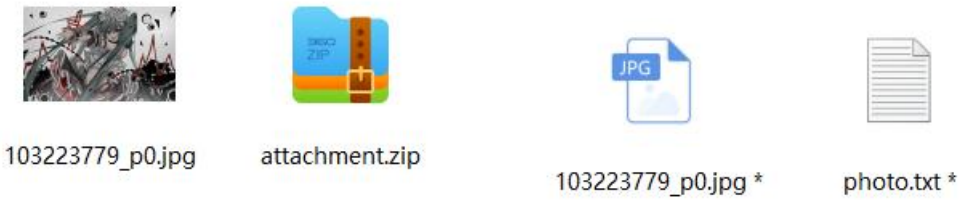


再根据题目里说的官网上找相关文字，就可以去一个个对应翻译。但更方便的是直接搜网上对这个文字的分析。然后又发现网上有人指出可以去官网扒 woff2 文件，跟大小写字母和数字逐一对应就好了。

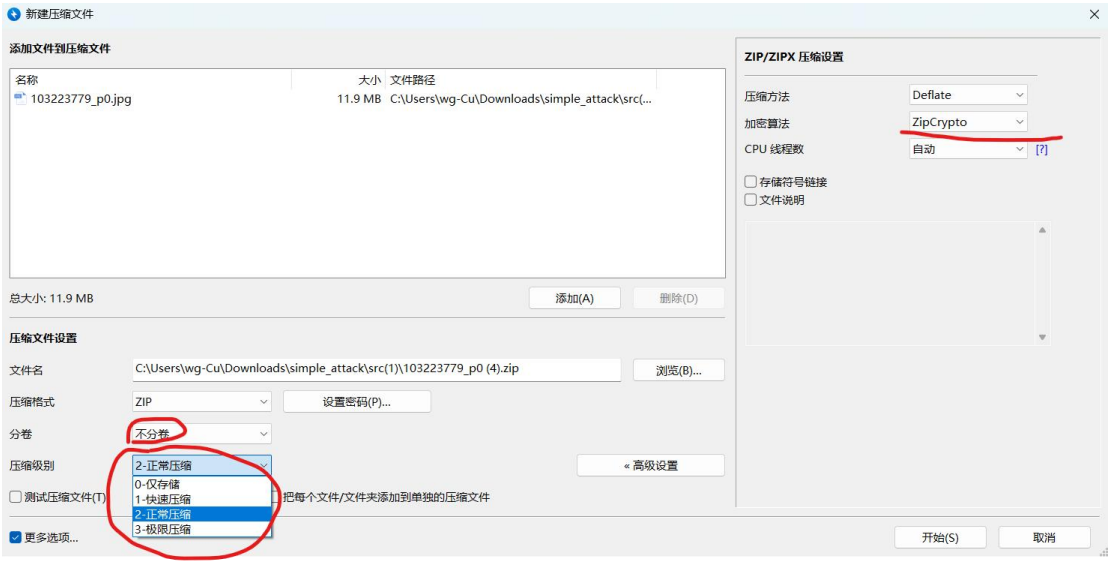
3. simple_attack

压缩包解密题，里面一张图片加另一个压缩包，包中包里有一个加密过的跟外面一样的图片（放 Bandizip 里看 crc 一致且命名一致）和加密过的 txt，压

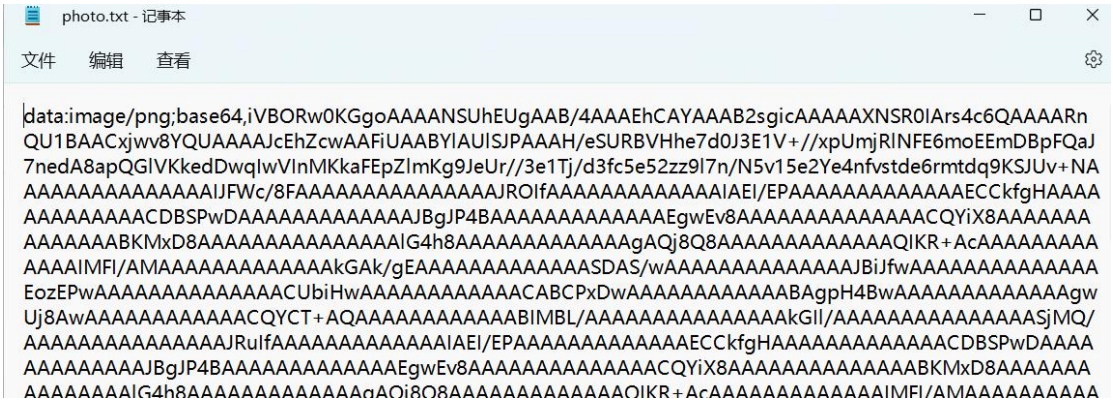
缩算法都是 ZipCrypto，那基本就是明文攻击的题型了。



所以把外面的那张未加密图片按压缩算法 ZipCrypto 压缩，其他项也要与压缩包内加密的图片一致，压缩级别逐个试过来就是正常压缩，然后放到 ARCHPR 里明文攻击，得到解密后的 attachment 压缩包。

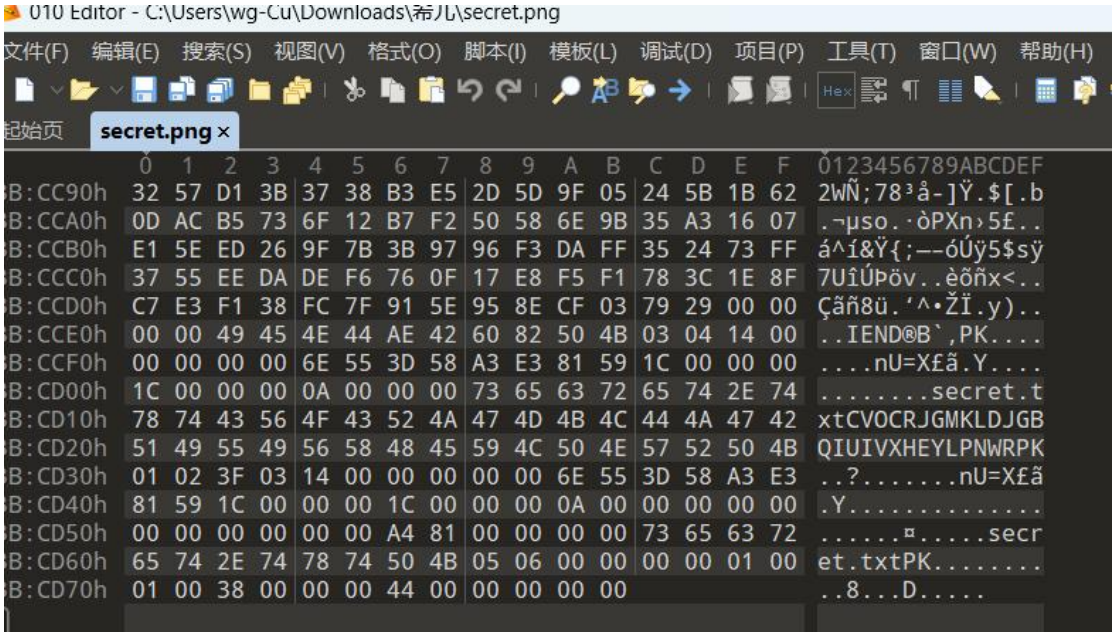


打开 photo.txt 里面是 Data URI scheme 的格式，放到浏览器地址栏里就可以查看图片，即 flag。



4. 希儿希儿希尔

拿到手是一个不正常显示的图片，题目说需要修复，所以在 010Editor 检查了一下，图片格式没有问题，但在末尾有 secret.txt 和 PK，说明图里藏有压缩包，直接改了拓展名，拿到压缩包里的 txt 文件。但没法直接解密，暂时还不知道有什么用。



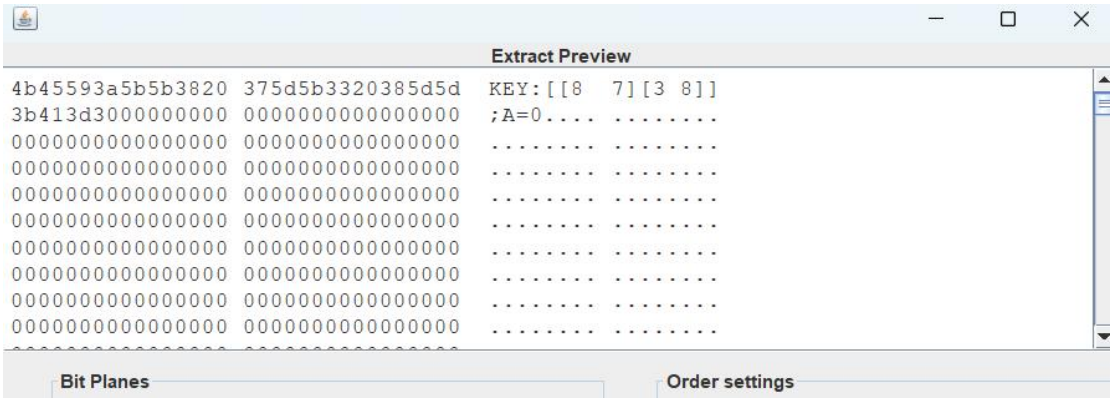
然后想把它放进 Stegsolve 里看看，结果打不开，看来应该是宽高被修改，需要 crc 校验。于是到 python 脚本里跑出正常宽高并在 010Editor 修改，得到希儿的正常图片。

```python
import os
import binascii
import struct

crcbp = open("secret.png", "rb").read()    #打开图片
for i in range(10000):
    for j in range(10000):
        data = crcbp[12:16] + \
            struct.pack('>i', i)+struct.pack('>i', j)+crcbp[24:29]
        crc32 = binascii.crc32(data) & 0xffffffff
        if(crc32 == 0x121B804D):    #图片当前CRC
            print(i, j)
            print('hex:', hex(i), hex(j))
```

```
misc计算图片正确尺寸脚本
D:\Python\venv\Scripts\python.exe D:\Python\misc计算图片正确尺寸脚本.py
1394 1999
hex: 0x572 0x7cf
```

接下来就可以在 Stegsolve 里正常打开，顺便检查了一下属性没什么问题，

但发现了 LSB 隐写藏着可疑的数据。



因为一开始没有好好看题，结果始终不知道这个到底是怎么用的，甚至后来以为这里只是我多想了转而去尝试其他隐写，直到我又看了一遍题："希儿希儿希尔"最后一个是"希尔"且"不过他似乎忘了这个加密的名字不是希儿了"，也就是说题目已经给出提示，去网上一搜"希尔加密"还真有，然后在线解密网站解决。

5. 签到

真真真·签到

Pwn

1. EzSignIn

连上去就完了《真真真·签到》

Reverse

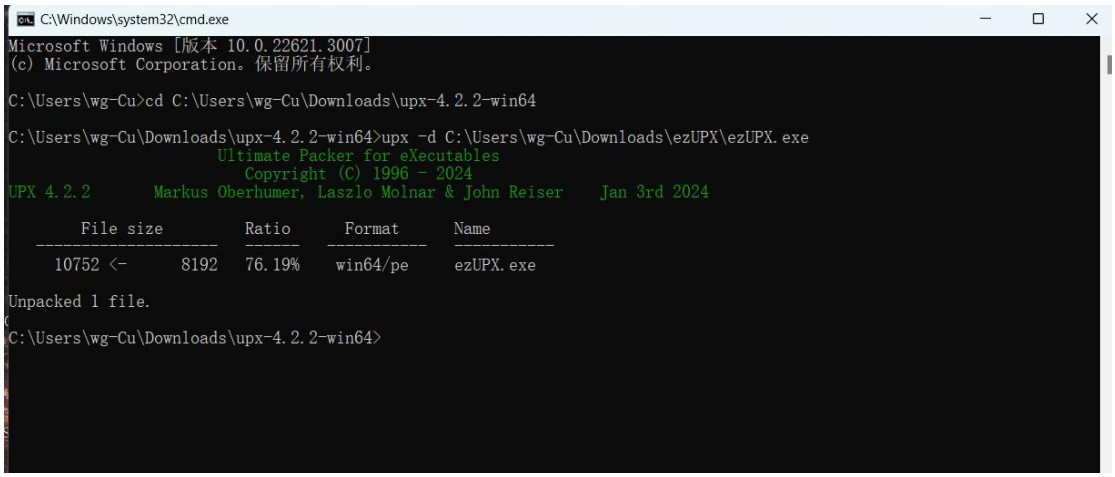
1. ezASM

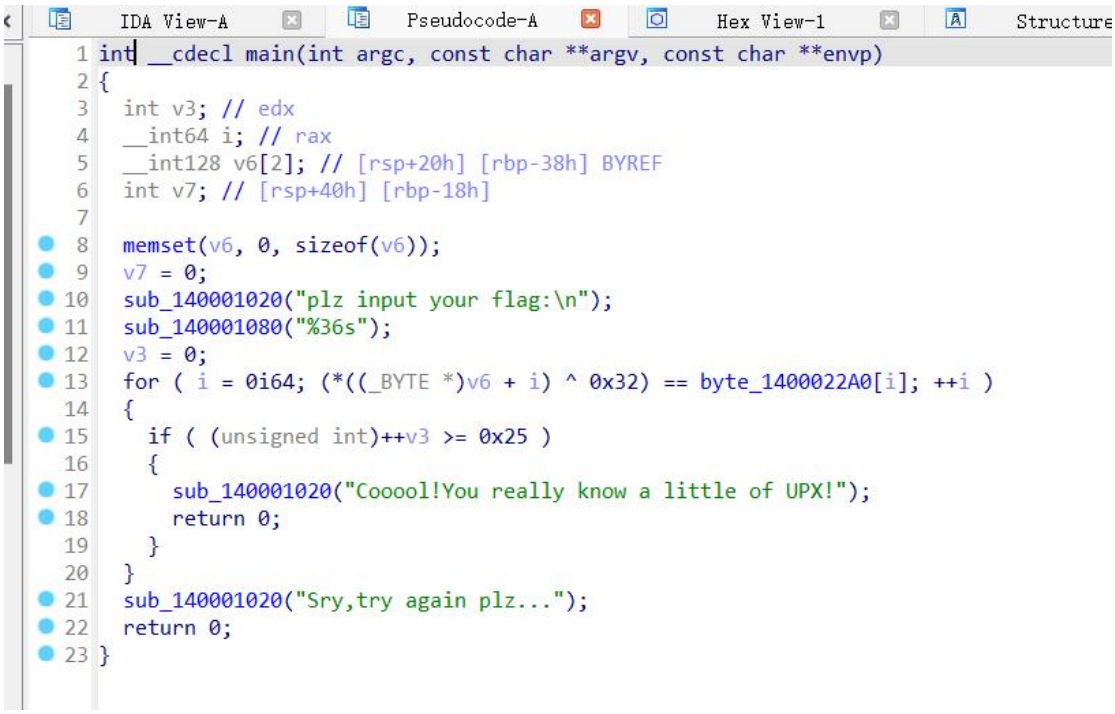
一开始去临时学习了一下汇编知识，后来感觉没必要，像 C 语言理解应该就可以，于是把上面 c 里的数和 0x22 异或一下，ASCII 转文字就是 flag 了。

```
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)                       //定义了33个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                      //将4的内容移至eax(累加寄存器)，移动后4的内容仍在，eax的内容被4覆盖。
    mov ebx, 1                      //(基址寄存器)
    mov ecx, format                 //(计数寄存器)
    mov edx, 20                     //(数据寄存器)
    int 0x80      //int 表示中断,编号为0x80是中断号在 Linux 中,0x80中断处理程序是内核,用于其他程序对内核进行系统调用。

    ; 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
    cmp al, byte [c + esi]
    jne failure_check
```

2. ezUPX

如题，是个 UPX，所以先用 upx 去壳即可





输入的 flag 要与 0x32 异或后等于 byte_1400022A0 的内容，所以找到并异或回去就得到 flag。（新学的快捷键 shift+e 导出这些文本）


```python
key='2'
pwd='d{vs`Ie]E\x13k\x02GmY\\\x02Em\x06m^\x03FF^\x01m\x02Tmgbj\x1302'
#解密
result=[]
#pwd为密文
for j in range(len(pwd)):
    result.append(chr(ord(pwd[j])^ord(key))) #跟KEY异或回去就是原明文
result=''.join(result)
print(result)
```

3. ezIDA

一进 IDA 差不多就看到了。

Web

1. Bypass it

一开始想偏了，以为要绕过别的什么，但其实就是绕过对注册的阻止就行，查看 html，可以看到注册相关的页面地址。

```
        </li>
        <li>
            <label> </label>
            <input type="submit" name="login" value="登录" />
            <a href="/register_page.php">注册</a>
        </li>
    </ul>
```

```
<html>
<head>
    <meta charset="utf-8">
    <title>用户注册</title>
    <link rel="stylesheet" href="/css/bootstrap.min.css">
    <script src="/js/jquery.min.js"></script>
    <script src="/js/bootstrap.min.js"></script>
</head>
<body>
<div class="container">
    <form action="register.php" method="post">
        <fieldset>
            <legend>用户注册</legend>
            <ul>
                <li>
                    <label>用户名:</label>
                    <input type="text" name="username" />
                </li>
                <li>
                    <label>密  码:</label>
                    <input type="password" name="password" />
                </li>
                <li>
                    <label> </label>
                    <input type="submit" name="register" value="注册" />
                </li>
            </ul>
        </fieldset>
    </form>
<script language='javascript' defer>alert('很抱歉，当前不允许注册');top.location.href='login.html'</script></div>
</body>
</html>
```

然后直接向 register.php 传参就好了 username=1&password=1&register=注册。然后登录。

2. ezHTTP

先是“请从 vidar.club 访问这个页面”，Referer=vidar.club 即可

再是“请通过 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 访问此页面”，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 即可

然后“请从本地访问这个页面”，我第一个想到的是 X-Forwarded-For，但是没反应，发现响应头里有提示“Hint：Not XFF”，所以需要更换其他等效的字段，最终 X-Real-IP: 127.0.0.1 可以发挥作用。

于是“Ok, the flag has been given to you ^-^”，去看响应头有“Authorization”，放到 https://jwt.io/解码即可。