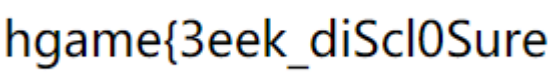
Hgame Week1 Writeup

> kevin

Web

谁吃了我的flag



打开页面看到一半的flag，看出seek disclosure的字样，fine，抓住泄露，用“ctf 泄露”搜了相关文章，试过了可能的.git .svn .hg还有其他的压缩包形式的源码泄露，全部返回404，思路在这里卡住了，后来先做别的题去了

然后尝试了各种扫描工具，包括御剑1.5还有github上一些脚本，也没扫到什么文件。

后来mki学长更新了hint，看到vim想到土土学长博客里的一篇文章，访问.index.html.swp,下载到文件，打开后获得flag：hgame{3eek_diScl0Sure_fRom+wEbsit@}

换头大作战

打开看到"do you want flag?"，submit以后根据提示，把method改为post，然后出现下面的提示

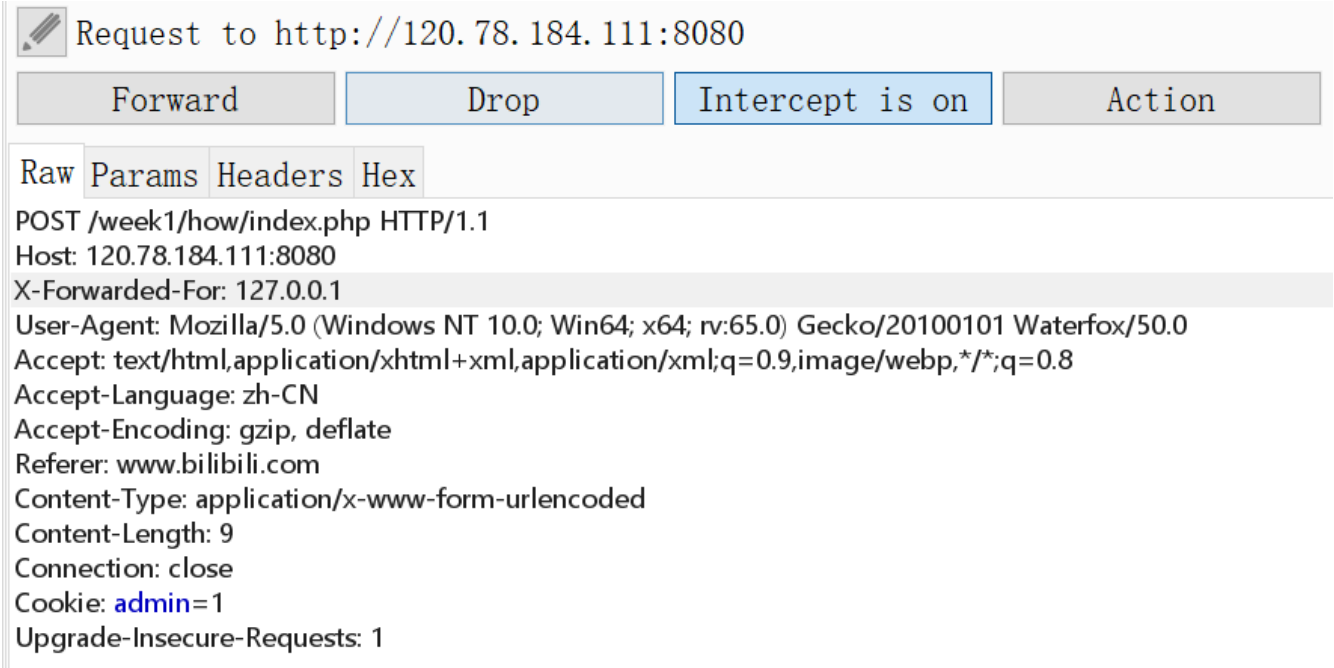
还贴心的给了网址学习，提示把请求IP改为本地ip，就用burp suite更改请求头，加上一行`X-Forwarded-For: 127.0.0.1`

然后下面提示要用水狐浏览器（噗），其实要伪装的话在浏览器里首选项里可以改，不过还是改请求头更快些，于是改`User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:65.0) Gecko/20100101 Waterfox/50.0`

又提示referer要为b站，于是改`Referer: www.bilibili.com`

然后根据提示改cookie`cookie: admin=1`

最后请求头改为下面这样



获得flag：hgame{hTTp_HeaDeR_iS_Ez}

very easy web

代码审计题，给出下面的php代码

```php
<?php

error_reporting(0);

include("flag.php");

if(strpos("vidar",$_GET['id'])!==FALSE)

  die("<p>干巴爹</p>");

$_GET['id'] = urldecode($_GET['id']);

if($_GET['id'] === "vidar")

{

  echo $flag;

}

highlight_file(__FILE__);

?>
```

查到urldecode函数的漏洞，把vidar转码两次后可以绕过，转码后输入

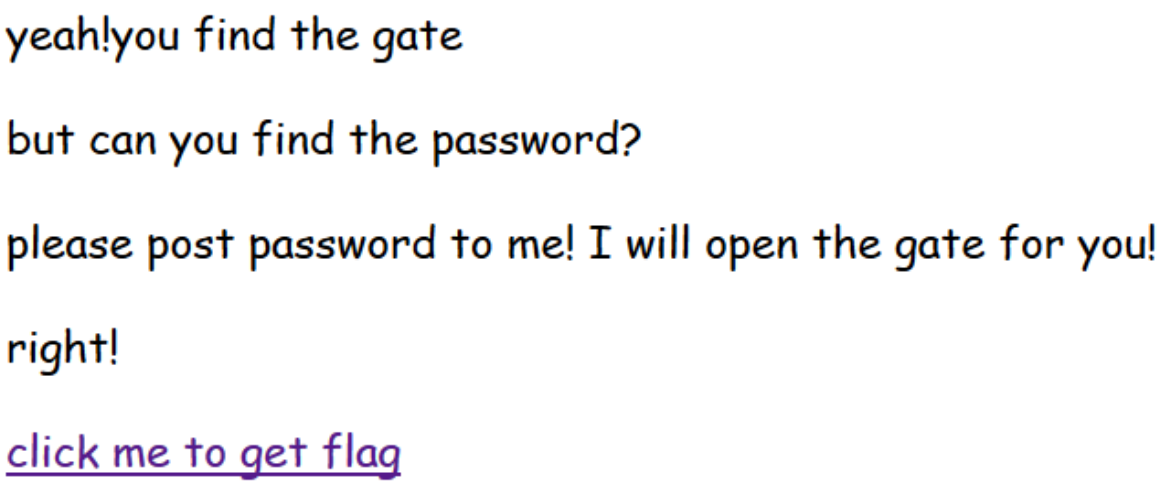
`http://120.78.184.111:8080/week1/very_ez/index.php?id=%2576%2569%2564%2561%2572`

获得flag: hgame{urlDecode_Is_GoOd}

can u find me?

日常f12,源码里找到链接，访问f12.php，在响应头里找到`password: woyaoflag`,便post `password=woyaoflag`

得到



结果页面直接跳转到tofast.php，看出这里有个302重定向，用burp suite抓包，响应包里找到：

flag：hgame{f12_1s_aMazIng111}

Re

brainfxxker

说实话这题是给了第一个hint以后才做出来的，直接读了brainfuck代码，然后根据理解大概改成这样的缩进，代码由下面的代码块重复组成，brainfxxk应该是在一个大数组上完成指针操作来实现的

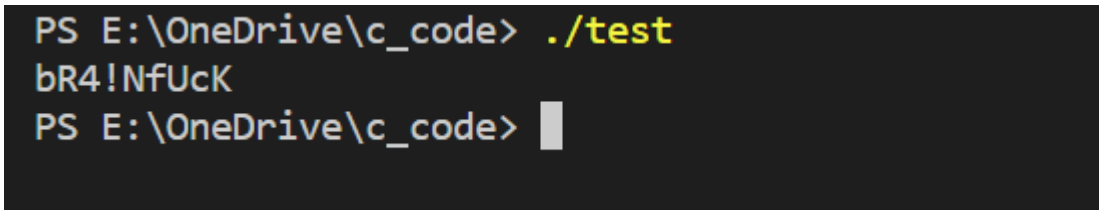
```
,
>++++++++++
[
  <----------
  >-
]
<++
[
  +.
]
```

先获得标准输入

第一排+是一个循环，数组第二项负责完成这个循环，数组第一项计数，[和]类似于c++中的{和}，然后减100再加2，所以一开始录入ascii码等于98的b，第二个循环的是数组第一项完成的，在第一项变为0之前一直在输出第一项所对应的char值

第一次做这道题我重复这个过程，查表得到flag（好蠢。。。） 写这篇writeup的时候想到，把代码改成下面这样，把flag输出就好了。。。

```
>++++++++++
[
  <++++++++++
  >-
]
<--
.
>>
```



flag： hgame{bR4!NfUcK}

Hello Re

搜索一下字符串得到flag：hgame{Welc0m3_t0_R3_World!}

Pro的Python教室(一)

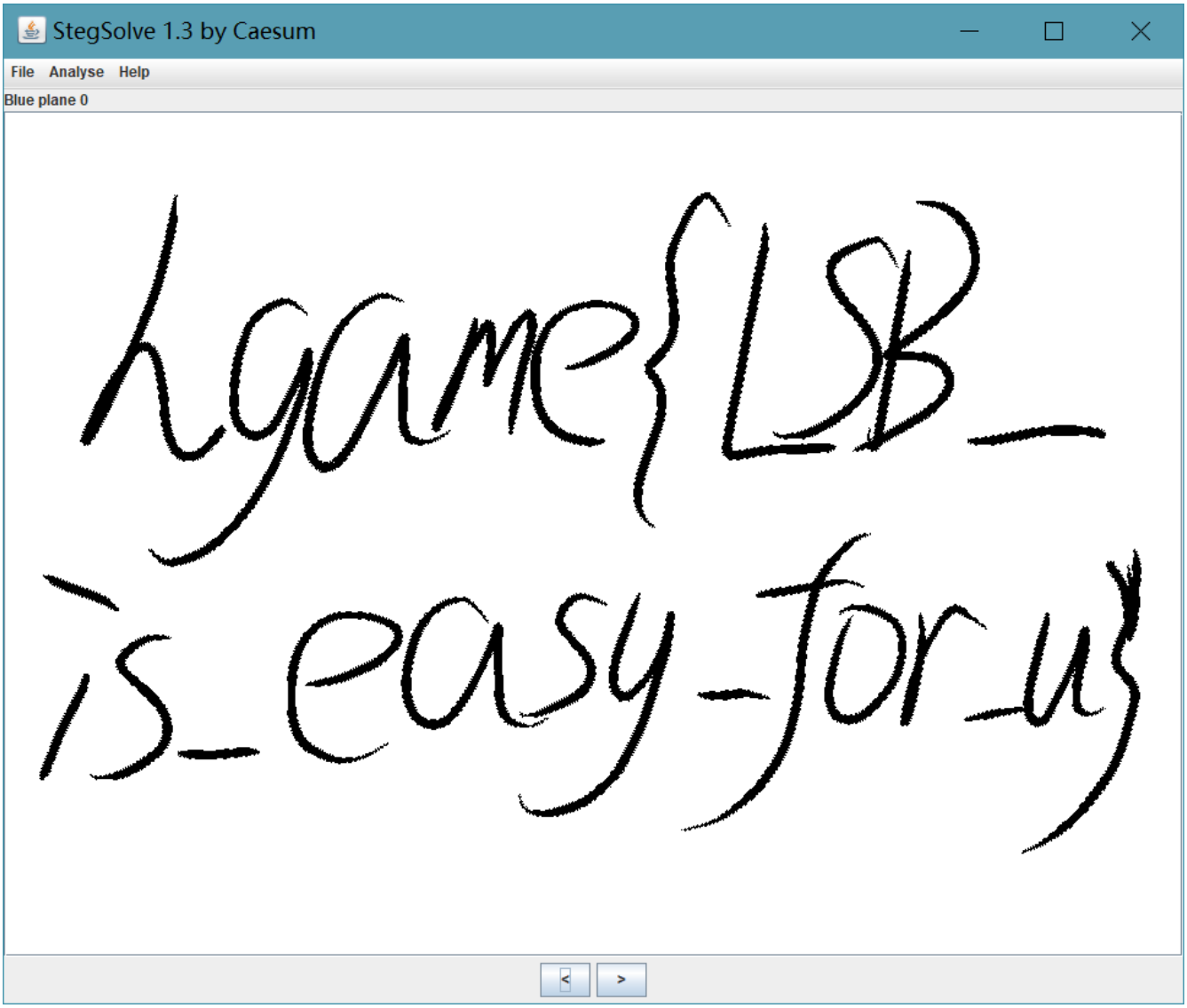
flag的第一段跟第三段都给了，第二段base32解码一下就可以了

flag：hgame{Here_1s_3asy_Pyth0n}

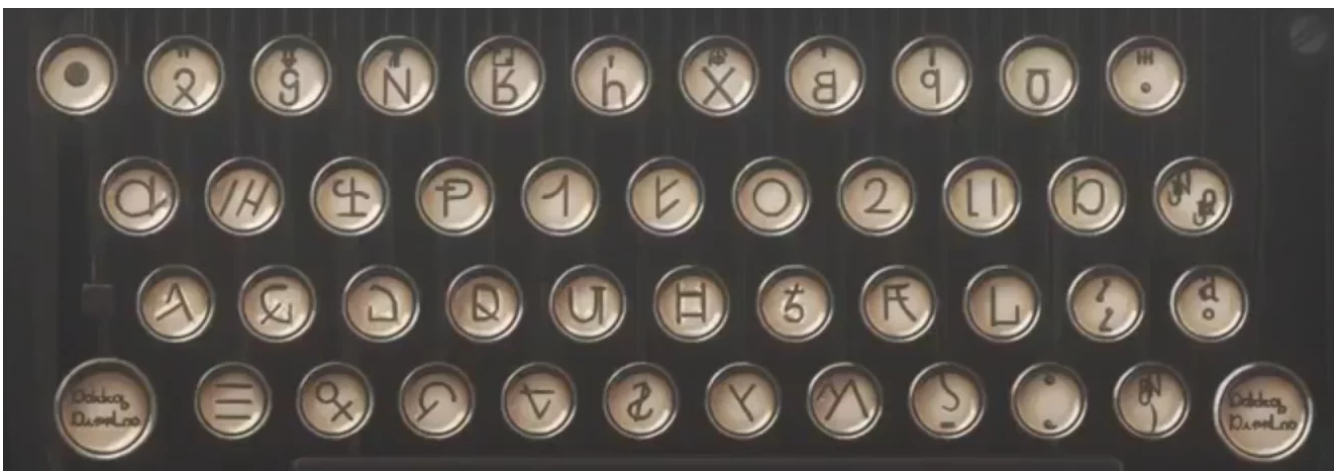
misc

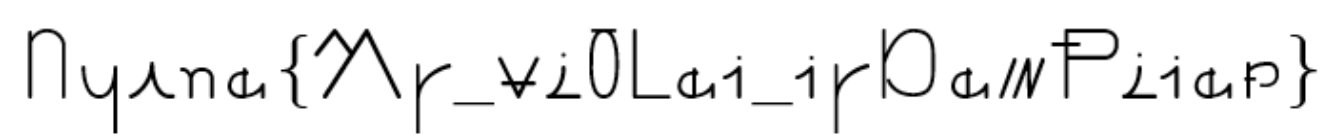
Hidden Image in LSB

刚看到到这道题的时候就想，先试试stegsolve吧😂，结果flag就找到了，后来给了hint也刚好是这种解法。。



打字机





对照自己键盘和打字机大致就能读懂，我是先猜出typewriter，然后紫罗兰就比较容易了

大小写的话，跟图中相同的是大写，图上没有的是小写

flag：hgame{My_vi0let_tyPewRiter}

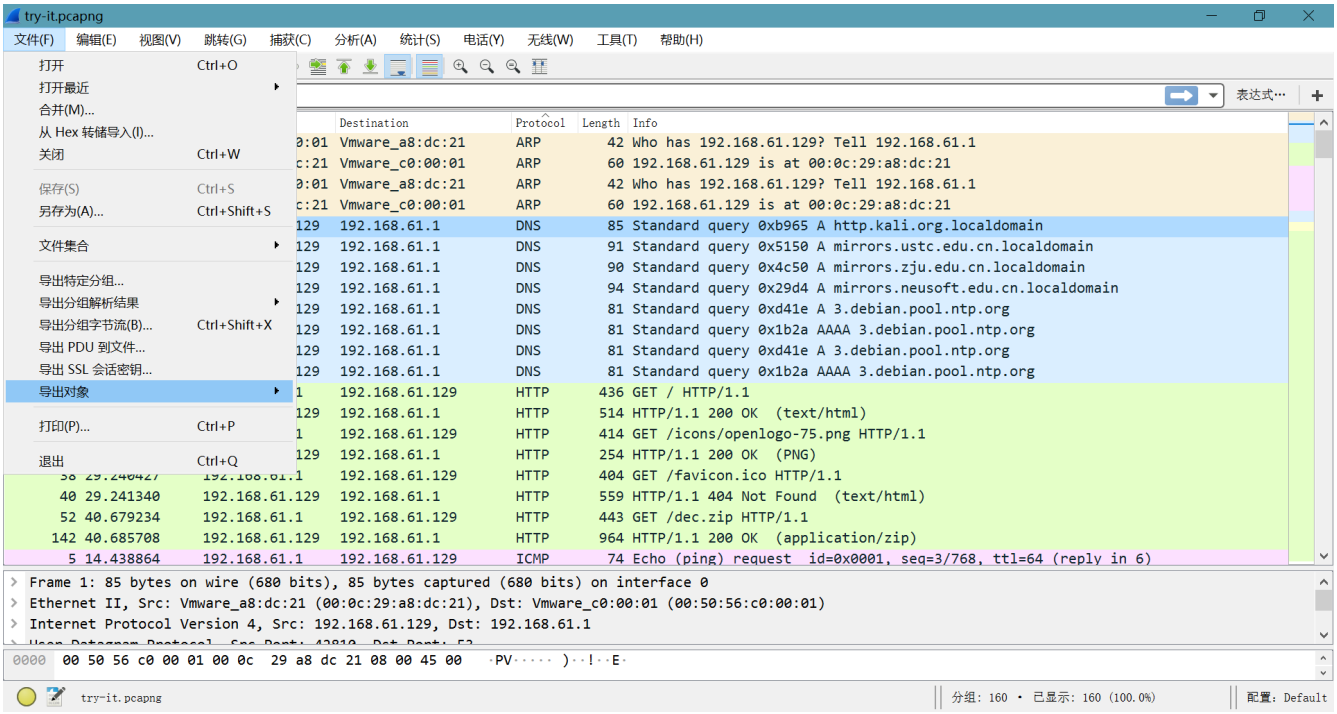
Broken Chest

先用binwalk扫一下，只发现了zip文件尾，大概猜到是文件头被改掉了，用winhex打开，果然是这样，改成50 4B 03 04保存一下，解压需要密码，密码在注释里，S0mETh1ng_U5efuL

解压后得到flag：hgame{Cra2y_D1aM0nd}

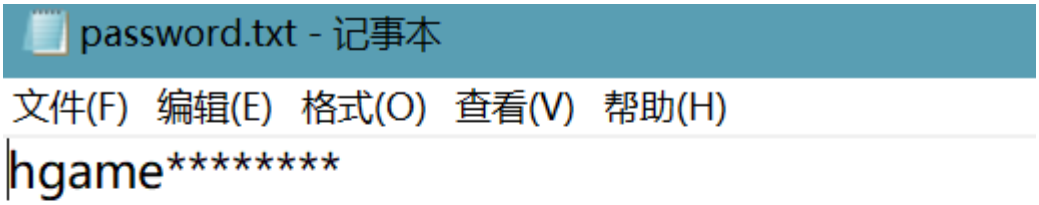
无字天书

发现是一个流量包，用wireshark打开，看到http包里有几个文件，导出一下文件



得到一个dec.zip

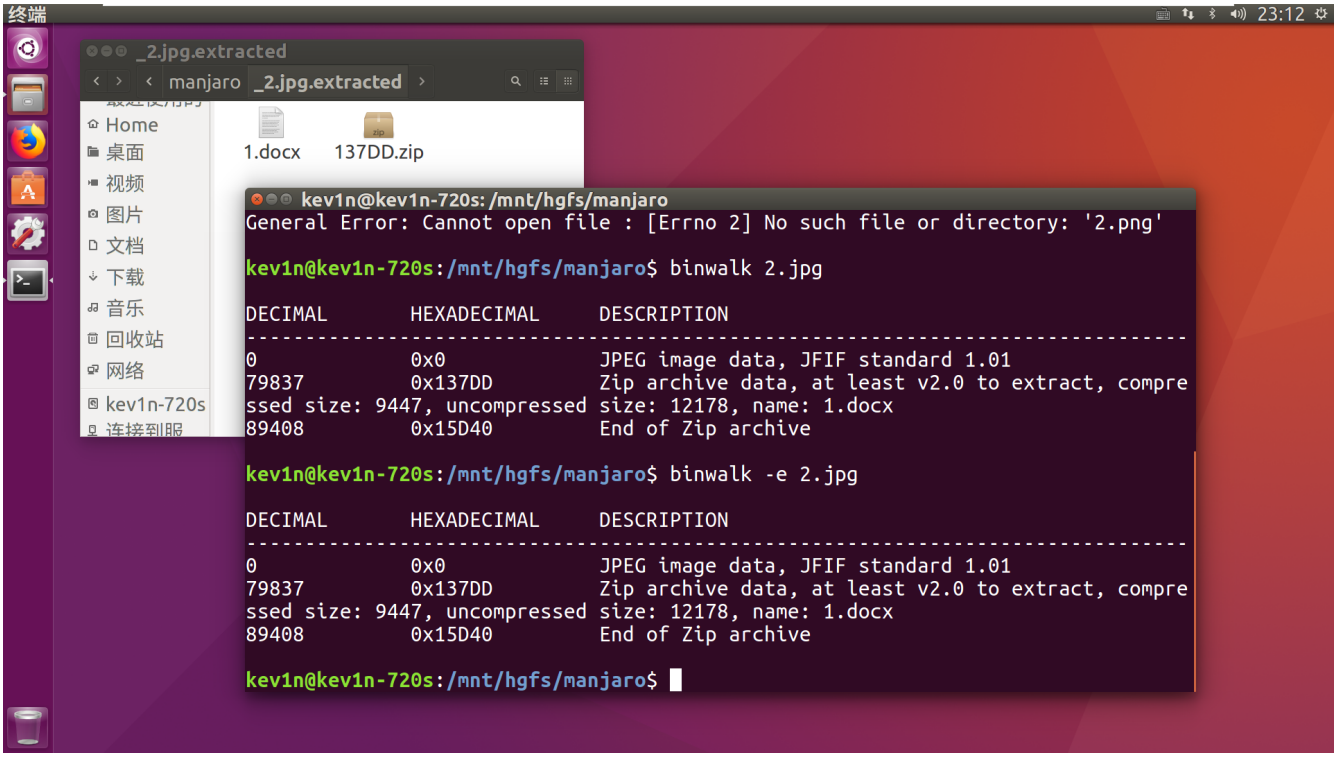
解压后有一个带密码的压缩包，和一个password.txt



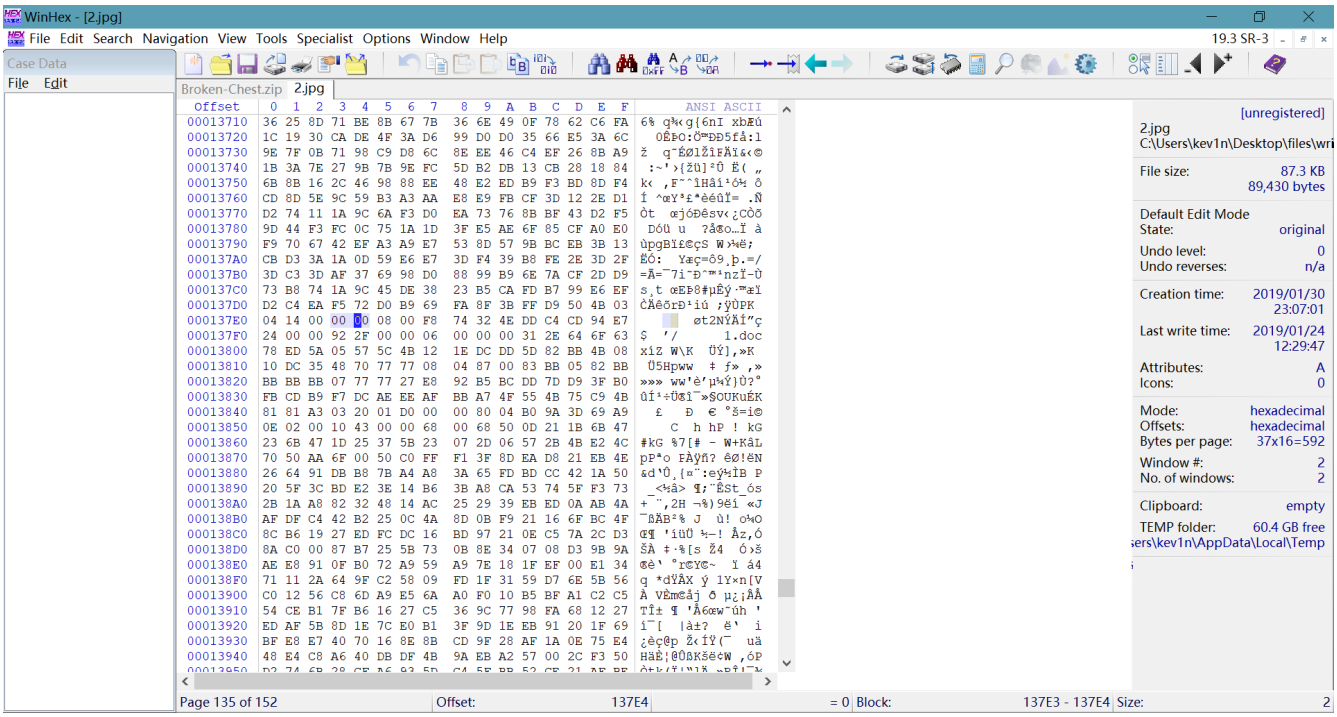
用archpr跑了一下字典，跑不出来，明文的要求也达不到

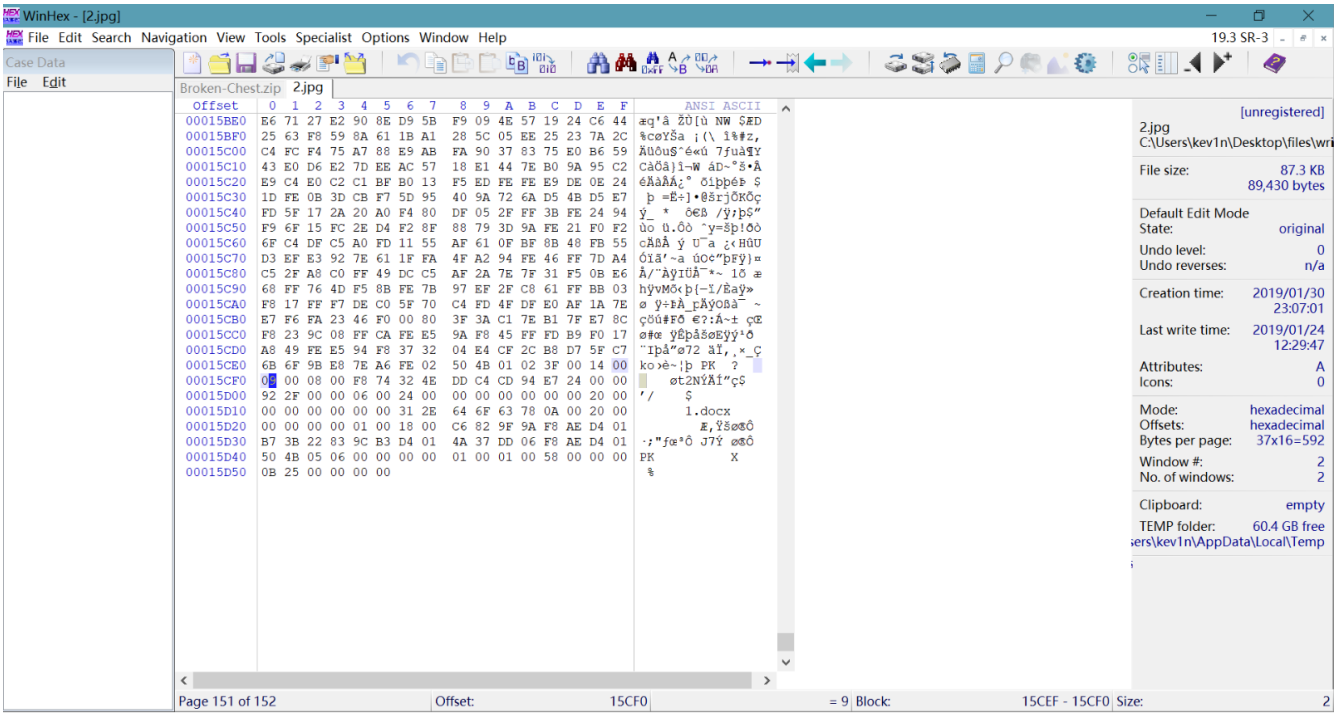
然后思路就断掉了，卡在这里两天，暴力的话也完全不可能跑出来，一边查资料一边试，最后终于发现掩码攻击，一方面也是没有经验，其实拿到这种密码，应该都跑一下弱密码试一下，最后跑出来密码是hgame25839421

解压得到这样一张图



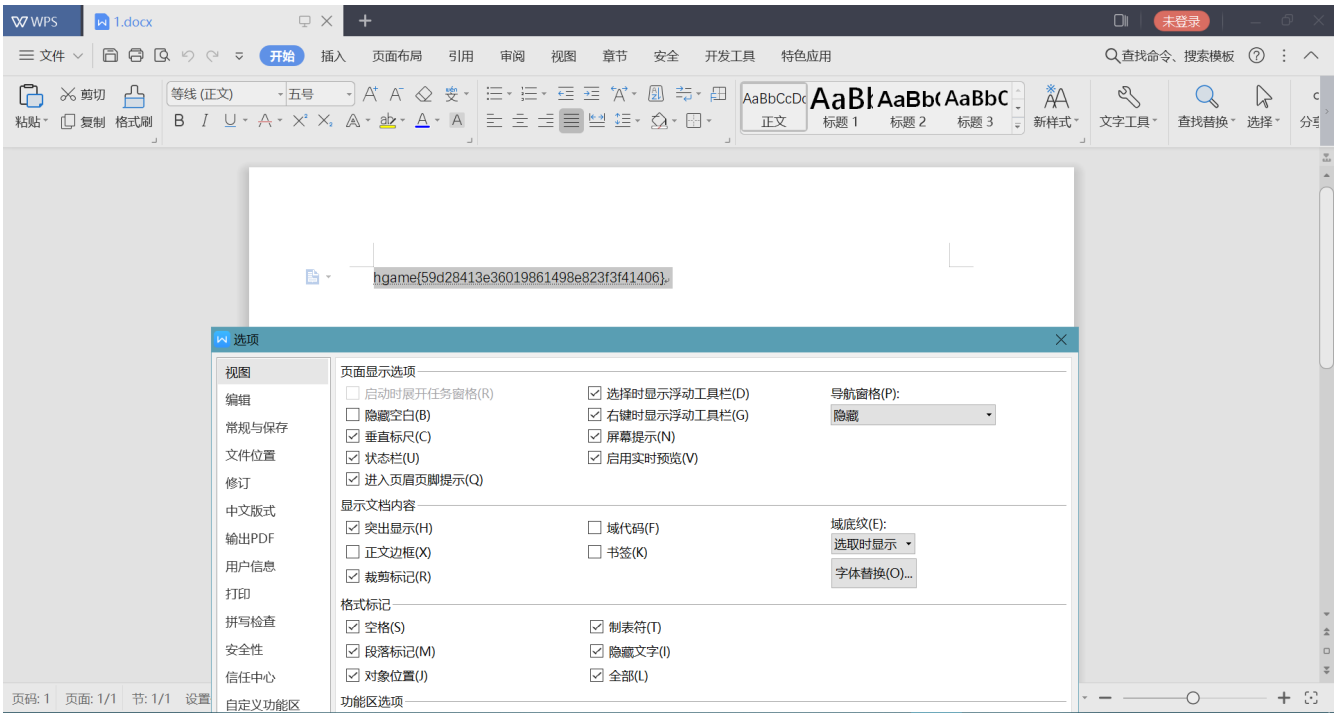
用binwalk扫一下发现一个zip包，在winhex中打开，搜索50 4B 03 04把zip文件分出来（其实这里做的有点麻烦，写writeup时候用`binwalk -e`命令或者`foremost`命令都可以直接分出来）





看到zip包中选中的目录加密区跟全局加密区数据不一致，如果两者都是奇数的话应该是真加密，这个文件是显然是伪加密把两个地方都改成0后可以解压出一个docx文件，其实写writeup的时候发现，直接用winrar修复可以直接绕过伪加密（好像`binwalk -e`命令也可以直接绕过伪加密）

1.docx像是个空文件，首选项里格式标记勾选全部显示



得到flag：hgame{59d28413e36019861498e823f3f41406}

crypto

MIX

--.../....-/....-/-.../--.../...--/...../..-./-..../-../-..../..-./--.../----./....-/....-/--.../.----/-..../-.../--.../-.../-..../..---/...../.----/-..../-..../...-/....-/...--/-----/-..../...../--.../-..

拿到这串密文看出是摩斯电码，在线解一下，得到

744B735F6D6F7944716B7B6251663430657D

感觉是base64，解一下发现乱码，就把base32跟base16都试了一下，base16解出下面的

tKs_moyDqk{bQf40e}

发现结构不对，想到去年的writeup，打算试一下栅栏密码，这期间找到一个不错的网站，常用密码都能解，http://www.zjslove.com/3.decode/

解码后选前面是5个字母的，大概对应hgame，

tsmyq{Q4eK_oDkbf0}

把上面的结果用凯撒密码位移一下

第13次解密后得到 hgame{e4sy_crypt0}

结果一直提交错误，后来oyeye提示注意中间结果，发现这个网站把大小写忽略了，fine，改一下得到flag：hgame{E4sY_cRypt0}

base全家

这题看起来够吓人，不过蛮有意思的

思路是存个txt里，用一下py的文件操作写个脚本跑一下，代码贴下面

```
import base64

readfile = open('base.txt','r')
writefile = open('output.txt','w')
txt = readfile.readlines()[0]
txt= base64.b32decode(txt)

t=str(txt,'UTF-8')
writefile.write(t)
writefile.write('\n')
writefile.close()
readfile.close()
```

当初做是重复好多次，手动把b32改成b64跟b16做的，=。=我也想学一下这题正确做法是什么

解出一行

base58：2BAja2VqXoHi9Lo5kfQZBPjq1EmZHGEudM5JyDPREPmS3CxrpB8BnC

找到一个解base58的脚本：

```
__b58chars = '123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwxyz'
__b58base = len(__b58chars)
```

```
def b58encode(v):
    """ encode v, which is a string of bytes, to base58.
    """

    long_value = int(v.encode("hex_codec"), 16)

    result = ''
    while long_value >= __b58base:
        div, mod = divmod(long_value, __b58base)
        result = __b58chars[mod] + result
        long_value = div
    result = __b58chars[long_value] + result

    # Bitcoin does a little leading-zero-compression:
    # leading 0-bytes in the input become leading-1s
    nPad = 0
    for c in v:
        if c == '\0':
            nPad += 1
        else:
            break

    return (__b58chars[0] * nPad) + result


def b58decode(v):
    """ decode v into a string of len bytes
    """

    long_value = 0L
    for (i, c) in enumerate(v[::-1]):
        long_value += __b58chars.find(c) * (__b58base ** i)

    result = ''
    while long_value >= 256:
        div, mod = divmod(long_value, 256)
        result = chr(mod) + result
        long_value = div
    result = chr(long_value) + result

    nPad = 0
    for c in v:
        if c == __b58chars[0]:
            nPad += 1
        else:
            break

    result = chr(0) * nPad + result
    return result

if __name__ == "__main__":
    print  b58decode("2BAja2VqXoHi9Lo5kfQZBPjq1EmZHGEudM5JyDPREPmS3CxrpB8BnC")
```

跑出来flag：hgame{40ca78cde14458da697066eb4cc7daf6}