

前言：作为完全 0 基础原本只是因为专业群老师的群发消息来混分的小朋友，很不幸，在写 WP 的前一夜因为经验为 0 将所有的脚本、截图、题目文件送入了回收站还清空了，加上时间紧凑，很多题目的 WP 只能通过文字描述和提问时使用的图像（出题的学长们都很耐心，我因为操作不当的一直发问给学长们带来了不少的麻烦，感谢并致以歉意）来呈现，见谅。

文章目录（以下并非完全按完成时间顺序排列）

Misc

Sign In

E99p1ant\_wan

神秘的海报

Where Am I

Reverse

Test Your IDA

lot

Help The Uncle Who Can't Jump Twice

Help Marvin

Crypto

RSA

神秘的电话

Web

Classic Childhood Game

Become A Member

Guess Who I Am

Show Me Your Beauty

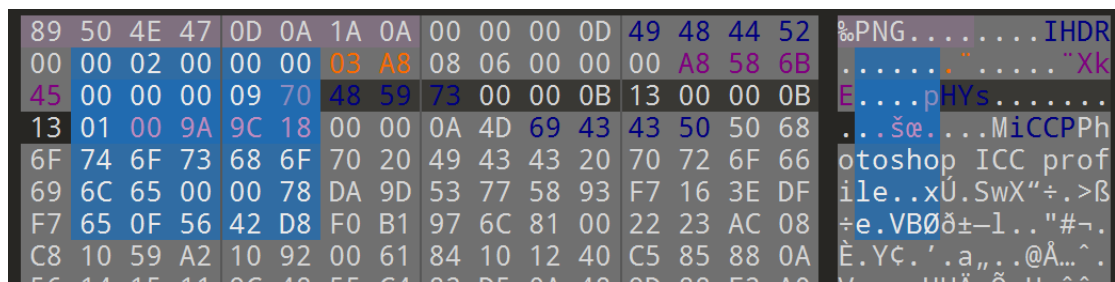
Misc

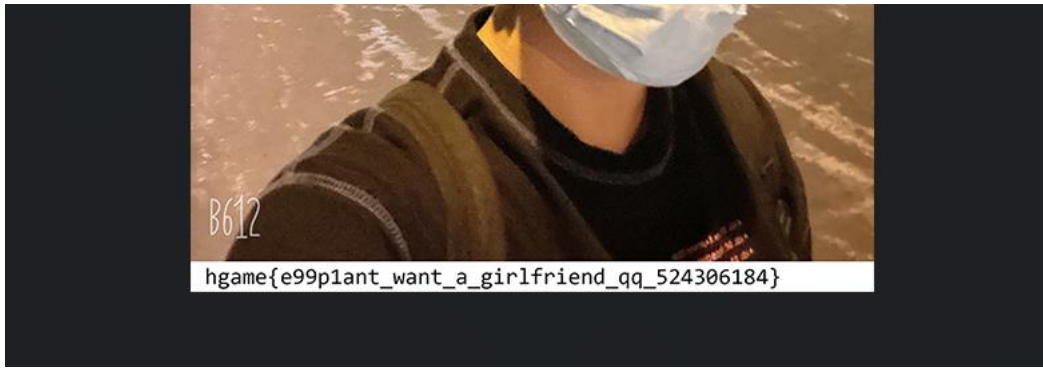
Sign In

通过线上的 base64 编译器得到了 flag。

E99p1ant\_wan

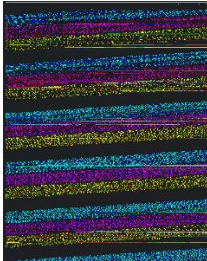
通过群内转发的图片隐写术文章产生了此图片有可能是通过改变宽高来进行文件的隐写，果不其然，通过 010Editor 修改了图片的宽和高之后得到 flag（修改了原本为 02 A8 的红色字符）。





神秘的海报

一开始也认为是通过改变宽高进行隐写，得到了神秘的东西。



在询问 Ek1ng 学长后加入了招新群，得到了有关 Misc 培训的资料，之后学会使用 stegsolve 来破解海报，非常巧的使用 PPT 上演示的完全一致的设置成功破解出一段话

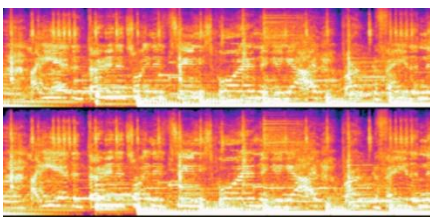
| Extract Preview  |                  |                   |
|------------------|------------------|-------------------|
| 6375726520656e6f | 7567682c20796f75 | Sure eno ugh, you |
| 07374696c6c2072  | 656d656d62657220 | still r emember   |
| 7768617420776520 | 74616c6b65642061 | what we talked a  |
| 626f757420617420 | 746861742074696d | bout at that tim  |
| 6521205468697320 | 6973207061727420 | e! This is part   |
| 6f66207468652073 | 65637265743a2060 | of the s ecret: ` |
| 6867616d657b555f | 4b6e30775f4c5342 | hgame{U_ Know_LSB |
| 657600a49207075  | 7420746865207265 | &W'.I pu t the re |
| 6374206f66207468 | 6520636f6e74656e | st of th e conten |
| 6420686572652c20 | 68747470733a2f2f | t here, https://  |

随后在网址上下载了 Bossanova.wav。

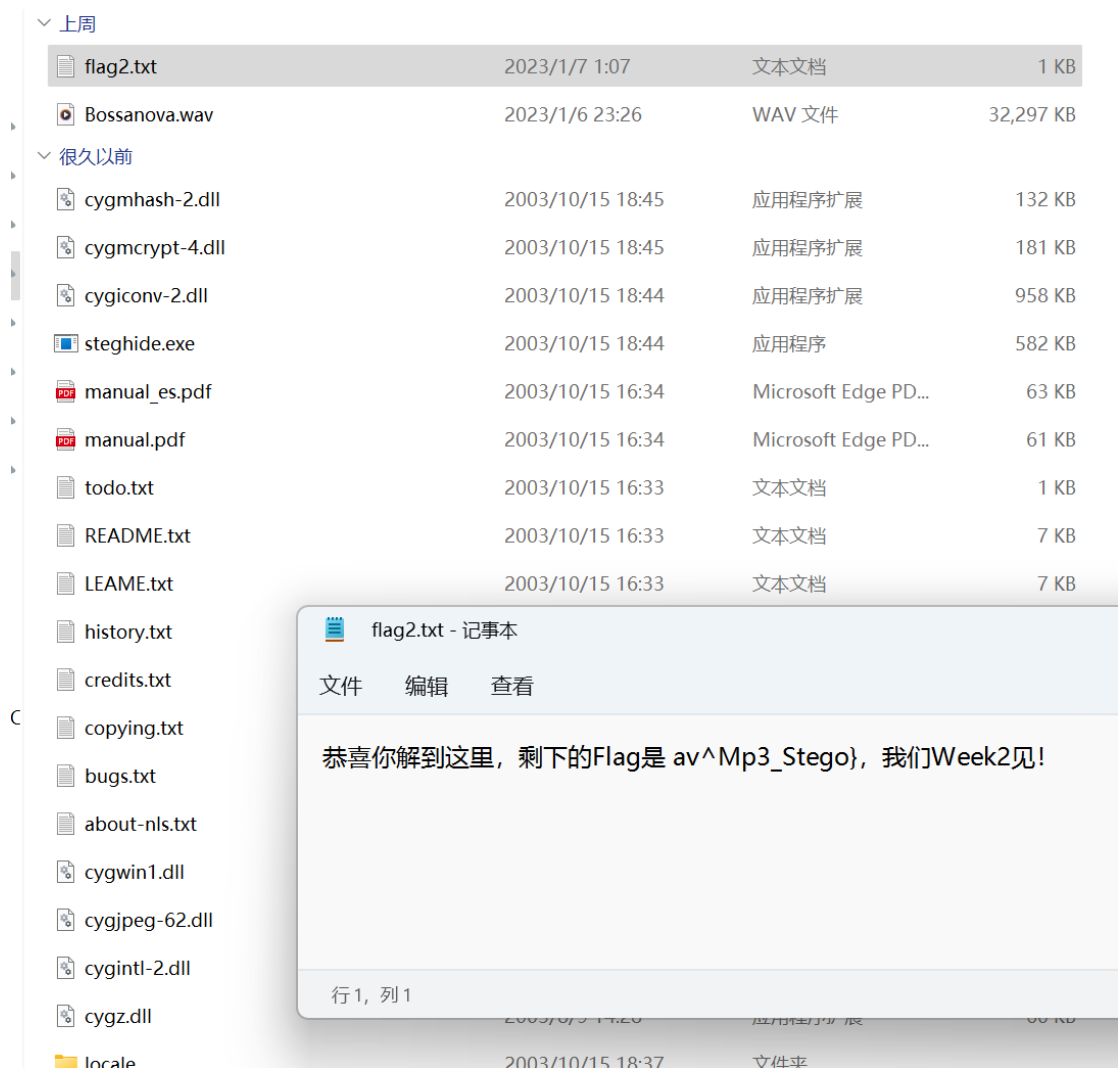
2023/1/6 23:26:05



身为萌新的我下意识参照 PPT 上的教学尝试使用 Audacity 和 Silentye 来处理音频，甚至还得到了一个看着好像有点问题的玩意。

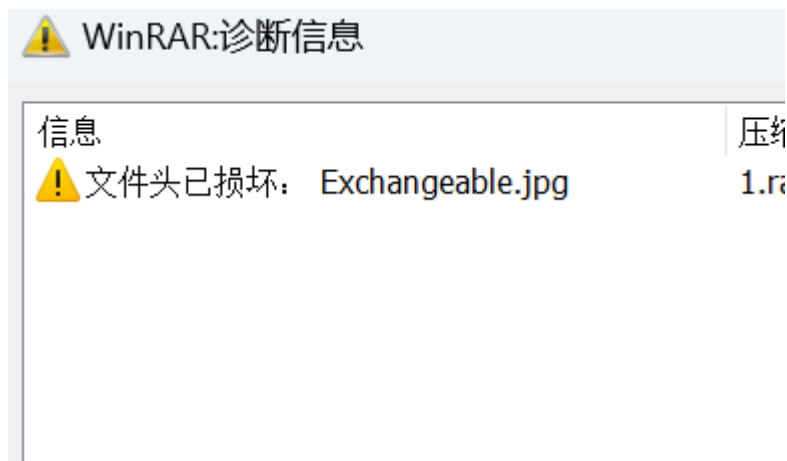


在核实这玩意完全没用之后在学长教导安装下使用 steghide 成功破解了后半 Flag



Where Am I

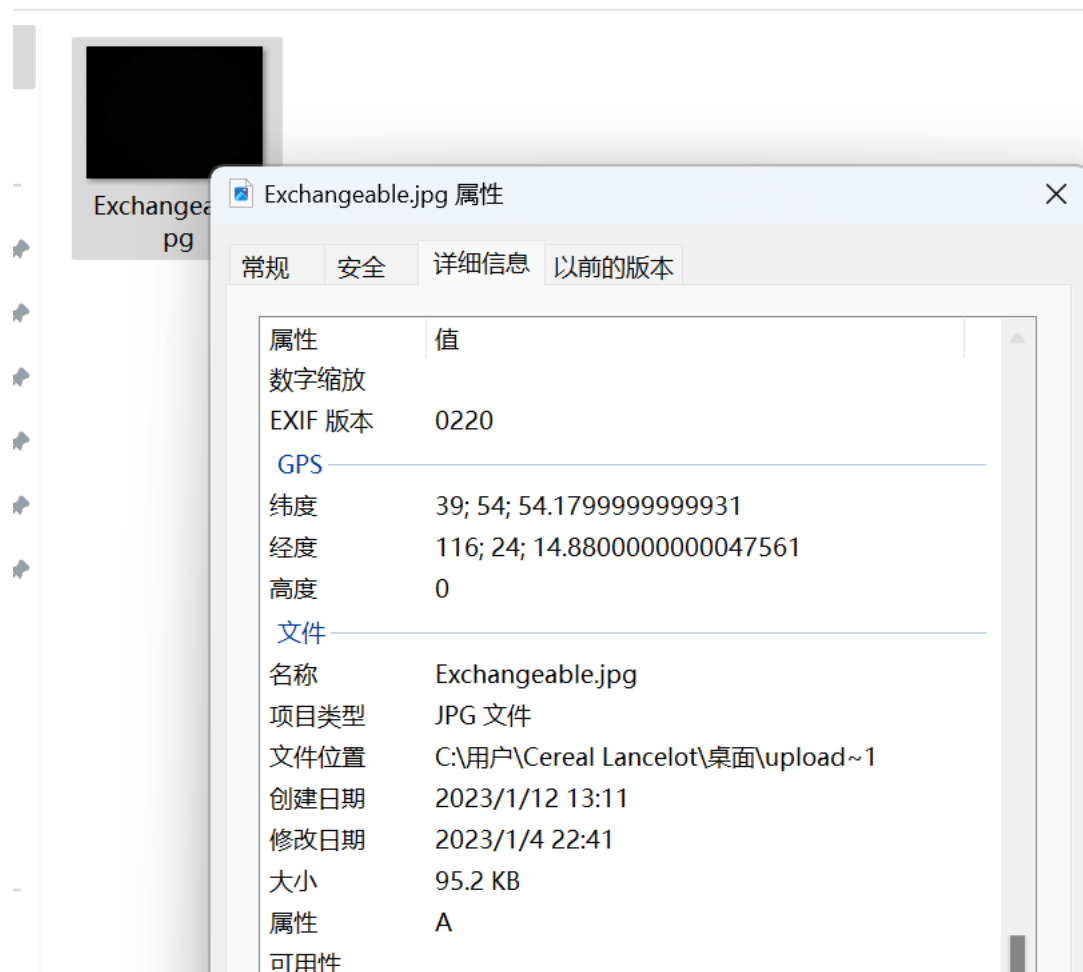




通过多方途径学习到了破解 RAR 伪加密的方法，

```
00 66 6F 72 6D 2D 64 61 74 61 3B 20 6E 61 6D 65 form-data; name
6D 22 75 70 6C 6F 61 64 22 3B 20 66 69 6C 65 6E ="upload"; file
61 6D 65 3D 22 66 61 6B 65 2E 72 61 72 22 0D 0A ame="fake.rar"..
63 6F 6E 74 65 6E 74 2D 54 79 70 65 3A 20 61 70 Content-Type: ap
70 6C 69 63 61 74 69 6F 6E 2F 6F 63 74 65 74 2D plication/octet-
73 74 72 65 61 6D 0D 0A 0D 0A 52 61 72 21 1A 07 stream....Rar!..
00 CF 90 73 00 00 0D 00 00 00 00 00 00 00 00 00 .I.s.....t.
74 20 90 35 00 BC CF 00 00 0F 7D 01 00 02 74 88 t .5.¼I...}...t^
FB 9C 38 B5 24 56 1D 33 10 00 20 00 00 00 45 78 ûæ8µ$V.3... ..Ex
63 68 61 6E 67 65 61 62 6C 65 2E 6A 70 67 00 F0 changeable.jpg.đ
67 4E 32 18 1E 15 50 C8 8E 21 C0 12 1D F3 32 48 gN2...PĚŽ!Ä..ó2H
00 D7 00 86 8A 57 44 44 46 25 15 15 1D F2 2B 2D .×.†ŠWDDF%...ò+-
D0 70 18 FA AD 51 2A 88 B5 AE FA FA C0 AD 51 16 .p.ê-0*^u@úêÄ-0.
```

解压出了全黑的 Exchaneable.jpg，期间损耗了近两天时间在 010Editor 的代码上寻找端倪、使用 foremost 来尝试拆分图片和通过盲水印脚本试图使图片显像（我估计学长在我最后以这个前提问出问题的時候哭笑不得了，但在原则上保持了沉默，毕竟算是最后一步），最后在第三天偶然点开属性得到经纬度。



Reverse

Test Your IDA

一开始下载的 IDA 无法打开附件（可能是我移动了什么部件导致的），而后通过重新下载导入附件得到 Flag。

lot

Help The Uncle Who Can't Jump Twice

得到 hint 后尝试学习 mqtt 了的本质，成功读懂了题干，并在第一晚试图通过 MQTTX 和手动枚举密码登录，因为愚蠢地从下方往上枚举在尝试了总数的约六分之一次后无果放弃，经过一整天的努力，在饭卡前辈的纠错、指导下和网络的力量下成功写出了能动的 python 脚本，通过 paho 链接了 MQTT 并进行爆破，最后成功登录并得到 Flag。第一次完全藉由代码脚本来攻克题目对我来说意义深重，不仅让我意识到了 python 的可贵，并且将对比赛的态度从“混分”转变为了“对 CTF 有一定兴趣而想去研究”(下图是最接近完成的版本，最终形态不幸被回收站杀害了，很可惜)。

```

filename='C:\\Users\\Cereal Lancelot\\Downloads\\Songs of Innocence and of Experience (2).txt'

with open(filename,'r') as file:
    for line in file.readline():

        import random
        from paho.mqtt import client as mqtt_client
        broker = '117.50.177.240:1883'
        port = 1883
        topic = "python/mqtt"
# generate client ID with pub prefix randomly
        client_id = f'python-mqtt-{random.randint(0, 100)}'
        username = 'Vergil'
        password = line

        def on_connect(client, userdata, flags, rc):
            if rc == 0:
                print("Connected to MQTT Broker!")
            else:
                print("Failed to connect, return code %d\n", rc)

        client = mqtt_client.Client(client_id)
        client.username_pw_set(username, password)
        client.on_connect = on_connect
        client.connect(broker, port)

```

(以下是 MQTTX 的成功样本)

|          |                |  |
|----------|----------------|--|
| * Name   | * Client ID ⓘ  | Username                                 |
| 2        | mqttx_c1901ab3 | Vergil                                   |
| Password | Keep Alive     | Clean Start                              |
| .....    | 60             | <input checked="" type="checkbox"/> true |

+ New Subscription

Nero/#

QoS 0

Plaintext

hgame{mqtt\_1s\_p0w3r}

2023-01-10 17:47:58:432

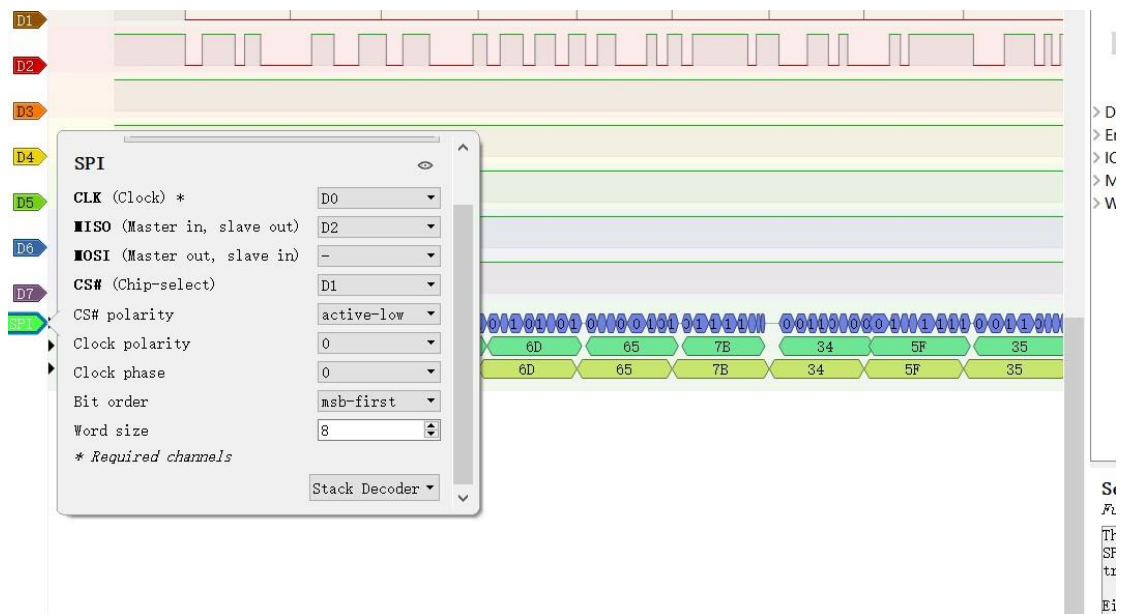
Topic: Nero/YAMATO QoS: 0

hgame{mqtt\_1s\_p0w3r}

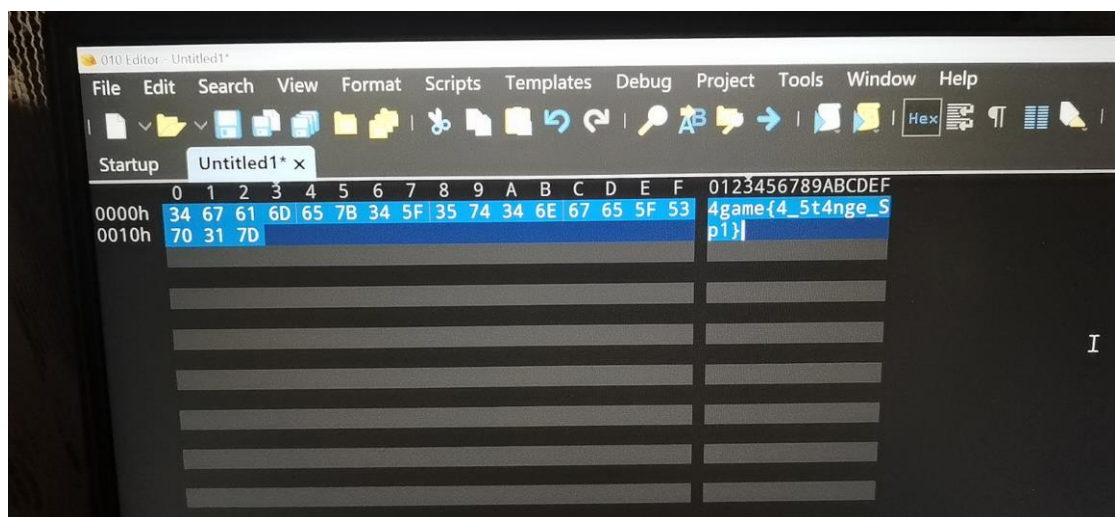
2023-01-10 17:48:08:437

Help Marvin (此处因键盘出问题打字困难, 只得简单描述)

由 hint:SPI 联想到使用 Pulseview 来解决问题, 下载后打开, 通过配置 SPI 得到型如下图



(此处因临近截止手忙脚乱误用 zadig 安装驱动挤占键盘导致键盘失灵，只得使用软键盘操作，故有误) 填入 010Editor 得到 Flag



Crypto

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}')

'''
c=110674792674017748243232351185896019660434718342001686906527789876264976328686134101972125493938434992787002915562500475480693297360
n=13512713834829975737419644706264085841692035009832009999311594971905135421354559664321673955545394619607811083472637547598179122306
'''

```

通过网站直接分解出了 p、q 的对应值。

| Search   | Sequences  | Report results  | Factor tables | Status |
|--|------------|---|---------------|--------|
| 135127138348299757374196447062640858416920350098320099993115949719051354213545596643216739555453946196078110834726375475981791223069451364024181952818056802 |            |   |               |        |
| Factorize!   |            |   |               |        |
| Result:  |            |   |               |        |
| tatus (?)  | digits     | number  |               |        |
| F  | 309 (show) | 1351271383...89 <309> = 1123913498...13 <155> · 1202291266...53 <155> |               |        |

之后使用浅薄的 python 知识和网络的力量成功写出了应用 gmsy2 函数的脚本并快速结束本题得到 Flag。（重新写的代码，不知道为什么 gmsy2 没掉了）

```

C:\Users> Cereal Lancelot > Desktop > import gmsy2.py > ...
1 import gmsy2
2 n = 135127138348299757374196447062640858416920350098320099993115949719051354213545596643216739555453946196078110834726375475981791223069451364024181952818056802
3 e = 65537
4 c = 110674792674017748243232351185896019660434718342001686906527789876264976328686134101972125493938434992787002915562500475480693297360867681000092725583284616
5 p = 11239134987804993586763559028187245057652550219515201768644770733869088185320740938450178816138394844329723311433549899499795775655921261664087997097294813
6 q = 120229126614209411592569751731802639375088427463430162252113082619617837010913002515450223656942836378041122163833359097910935638423464006252814266959128953
7 d = int(gmsy2.invert(e, (p-1) * (q-1)))
8 m = pow(c, d, n)
9 print(m)
10 print(hex(m))
11

```

## 神秘的电话

题目的第一部分是一段包含 + 和 Ascii 字符的代码，联想到尝试在结尾加上两个 == 用 base64 破解。

Base64 | URLEncode | MD5 | TimeStamp

请输入要进行 Base64 编码或解码的字符
 

5Yeg5Liq5pif5pyf5YmN77yM5oiR5Lus5pS25Yiw5LiA5Liq56We56eY55qE5ral5oGv44CC5L2G5piv6L+Z5Liq5ral5oGv6KKr6YeN6YeN5Yqg5a+G77yM5oiR5Lus5LiN55+H6YGT5a6D55qE55y5q2j5ZCr5LmJ5piv5LuA5LmI44CC5ZSv5LiA55+H6YGT55qE5L+h5oGv5piv5YVw5LqO5a+G6ZKl55qE77ya4oCc5Y+q5pyJ5YCS552A57+76L+H5Y2B5YVr5bGC55qE56+x56yG5omN6lO95oq16L6+5YyX5qyn56We6K+d55qE57ul54K54oCd44CC==

编码 (Encode)

解码 (Decode)

↕ 交换

(编码快捷键: **Ctrl** + **Enter**)

Base64 编码或解码的结果:
 

☐ 编/解码后自动全选

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。”

而后通过 Audacity 来观测电码得到密文：0223e\_priibly\_honwa\_jmgh\_fgkcaqoqtmfr  
由之前的线索得知，我们需要倒过来写密文并通过 18 位的栅栏来获取前一阶段的文本

rfmtqoaqckgf\_hgmj\_awnoh\_ylbirp\_e3220

此后考证 Vidar.club 了解到北欧神话与 vidar 的缘由联想到密钥或许是 vidar，最后将语句和密钥投入测试得到 Flag。

转换前：

rmocfhm\_wo\_ybipe2023\_ril\_hnajg\_katfqgg

密钥: vidar

加密>

解密>

转换后：

welcome\_to\_hgame2023\_and\_enjoy\_hacking

Web

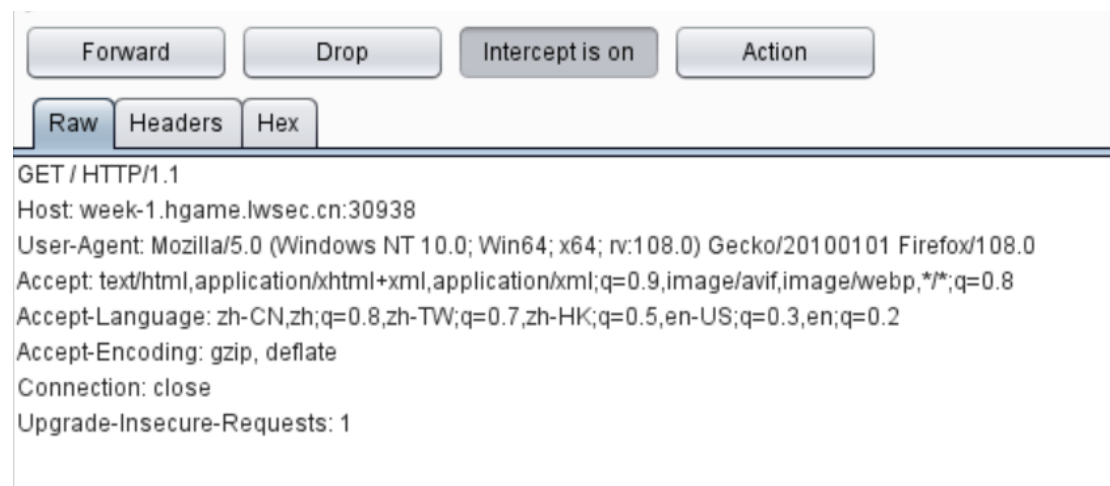
Classic Childhood Game

HGAME 开始的时候发现是唯一有机会 0 基础破拆的题目，花了相当多的时间正规攻克了魔塔，攻克第二天（给出 hint 前数日）在做别的题目的时候偶然知道了 F12，有点无语（0 基础的可怕）。

Become A Member

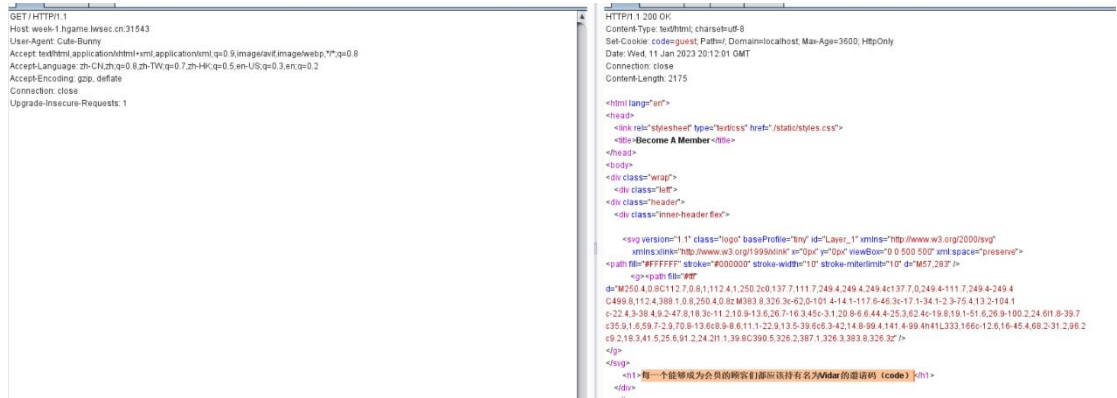
（这个题目是在 Show Me Your Beauty 之后完成的，因此 Burp 的熟练度提升很多）

由还没有开始做就白嫖来的 hint 得知，本题是与 HTTP 相关的题目，发现一个无法用鼠标直接交互的网页后，使用 burpsuite 尝试抓包。

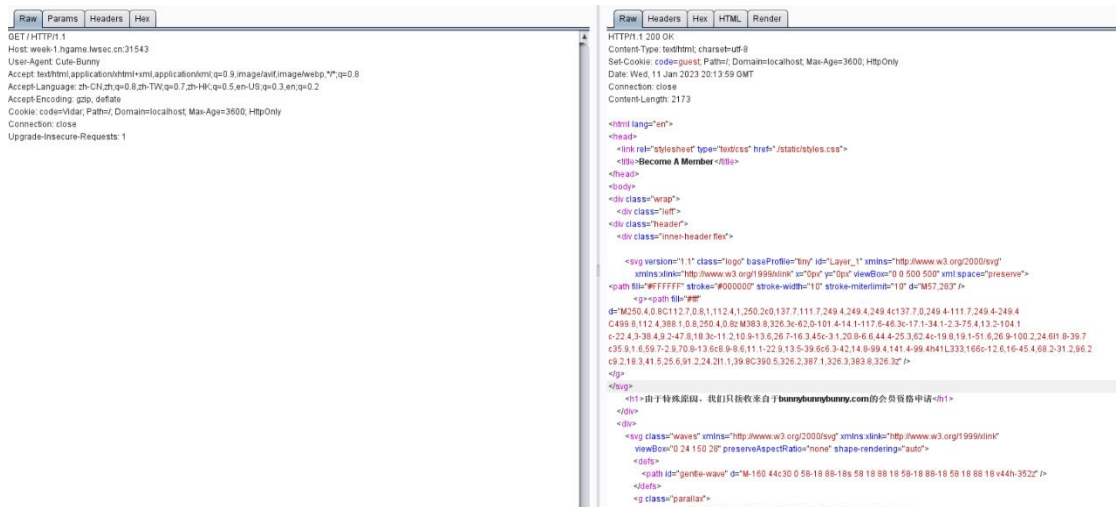


之后把文档送入 repeater 进行，依次通过“修改 User-Agent 来解决身份验证”；

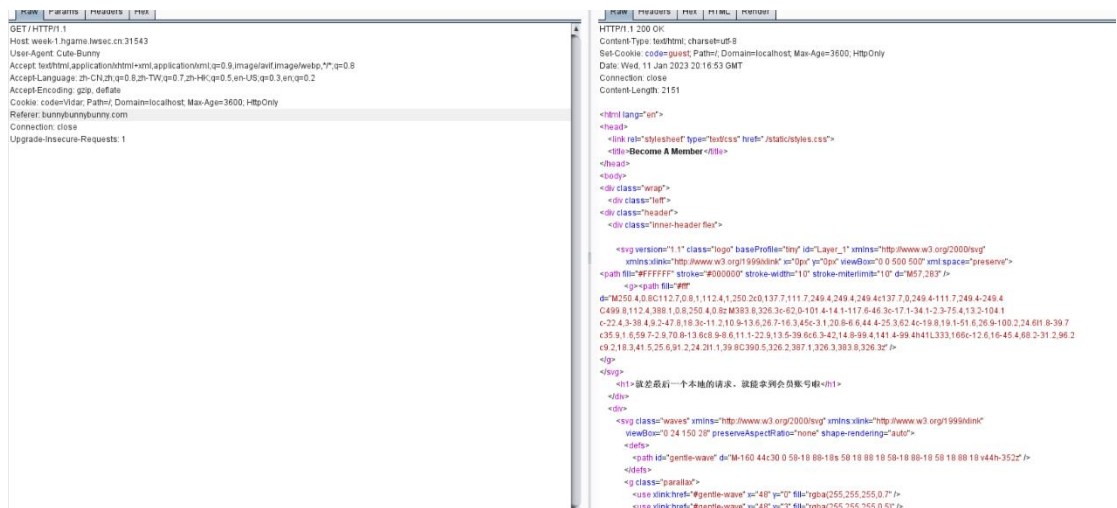
“增添 Cookie: code=Vidar; Path=/; Domain=localhost; Max-Age=3600; HttpOnly 来应对每一个能够成为会员的顾客们都应该持有名为 Vidar 的邀请码（code）”；



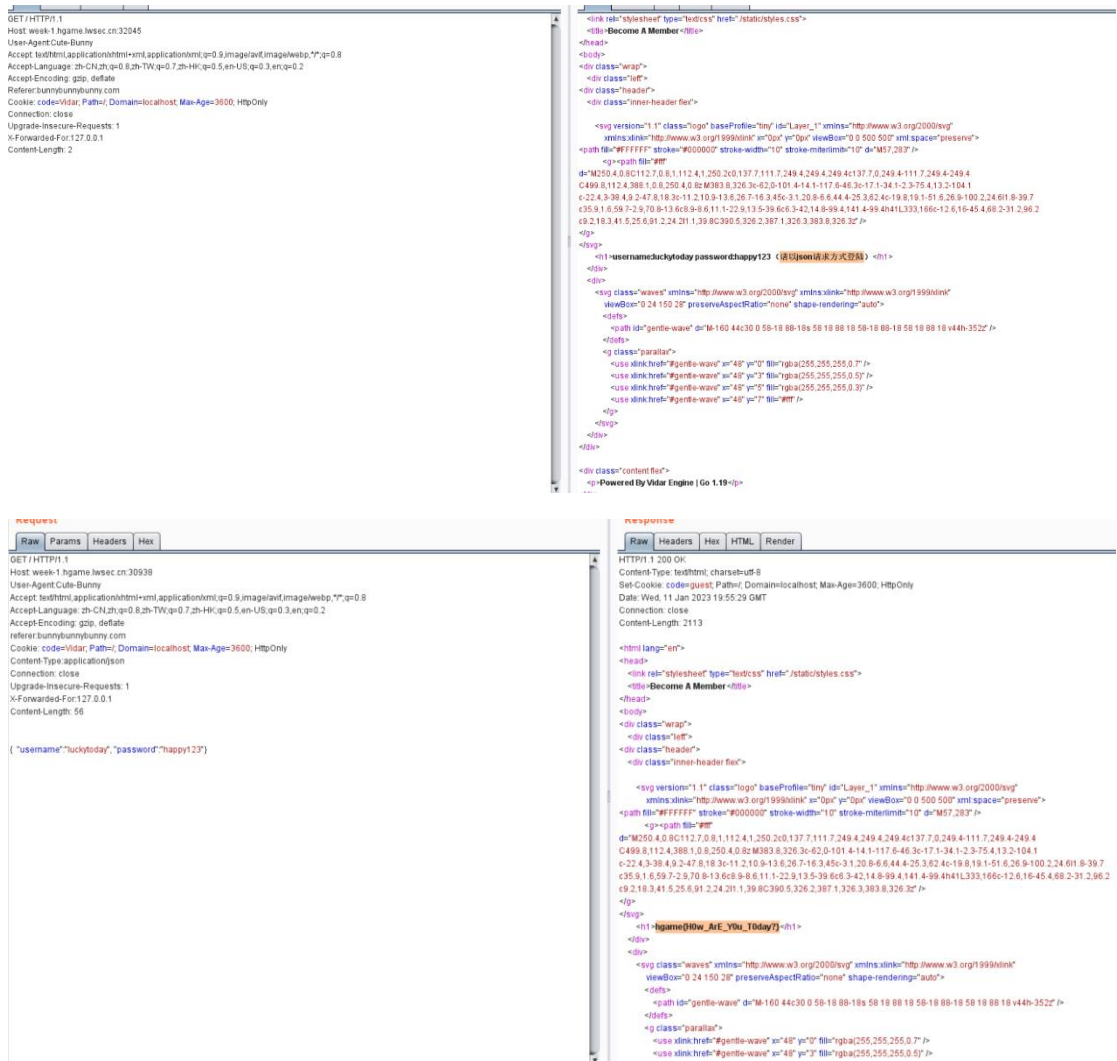
“增添 Referer: bunnybunnybunny.com 搞定了由于特殊原因，我们只接收来自于 bunnybunnybunny.com 的会员资格申请”；



很幸运，这次使用 X-Forwarded-For:127.0.0.1 就能达成本地的要求；



这一步后出现的要求看起来有点古怪，但是登录请求因言丁真，鉴定：为 POST，在学习了 HTTP 有关 POST 的知识和有幸得到了相关知识的指点后，成功突破，得到 Flag。

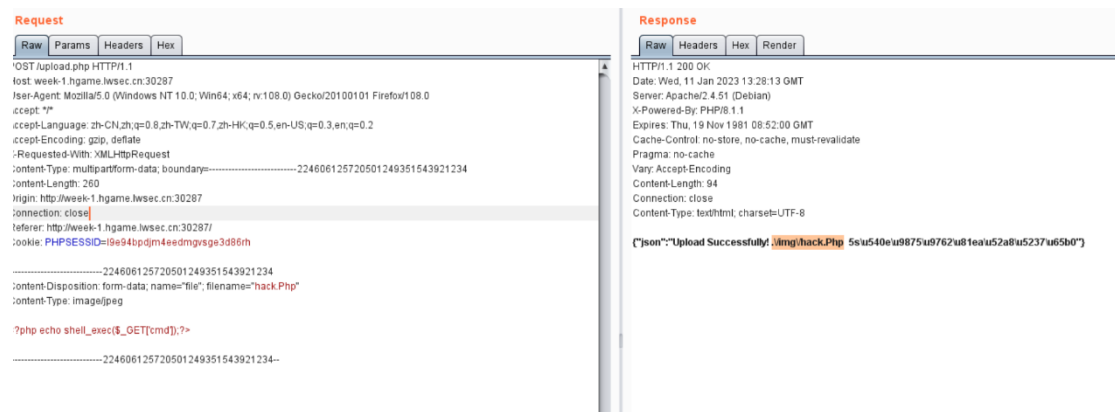


## Guess Who I Am

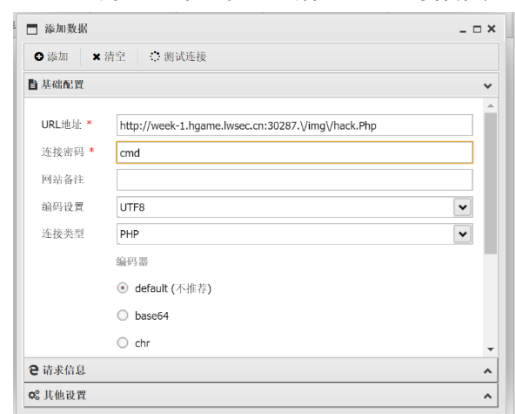
通过手扣得到 Flag 了解一下学长们都是些什么大佬，后来写脚本的时候发现手扣完的用时居然比我写脚本用时短很多，受到了很大的打击。

## Show Me Your Beauty

在向 R1esbyfe 学长寻求 hint 之前便有幸在网上学会了使用 php 一句话木马 + burpsuite + 蚁剑的 Combo 来试图得到 Flag。但是因为之前玩 Minecraft 装载了 Java17.0.5 导致 burpsuite 起初无法启动，试图不使用 burpsuite 突破失败，耽误了非常多的时间。之后使用了网上各种 php 上传方法却不得要领(同时因为知识的极度匮乏出了非常多操作错误)，在完成当日屡次麻烦 R1esbyfe 学长帮我纠错，并告知我不需要使用图片马。自认了解了源代码的意思，在用尽网上直接突破的方法无果后，偶然意识到后端有可能屏蔽了 php 的后缀，在之前使用的方法的基础上略加修改突破成功。



之后因为对路径的不了解再一次身陷图圖。



最后经过尝试对木马指令进行修改、在源代码中找到 php 文件的去向，成功链接获得了 Flag（虽然可能不大会再遇到这种题目，但是通过这次 HGAME，我有幸对 php 渗透类题目有了一定的理解）