HGAME Week1 Writeup

WEB

Cosmos的博客

（qaq请原谅我就是那个不懂事拿着扫描器乱扫大军的一员.....我再也不会这么干了555

根据明显的粗体提示 **版本管理工具** 和提及到的Github，谷歌一下后知道是.git的源码泄露，然后在url后加上.git/结果404 Not Found。

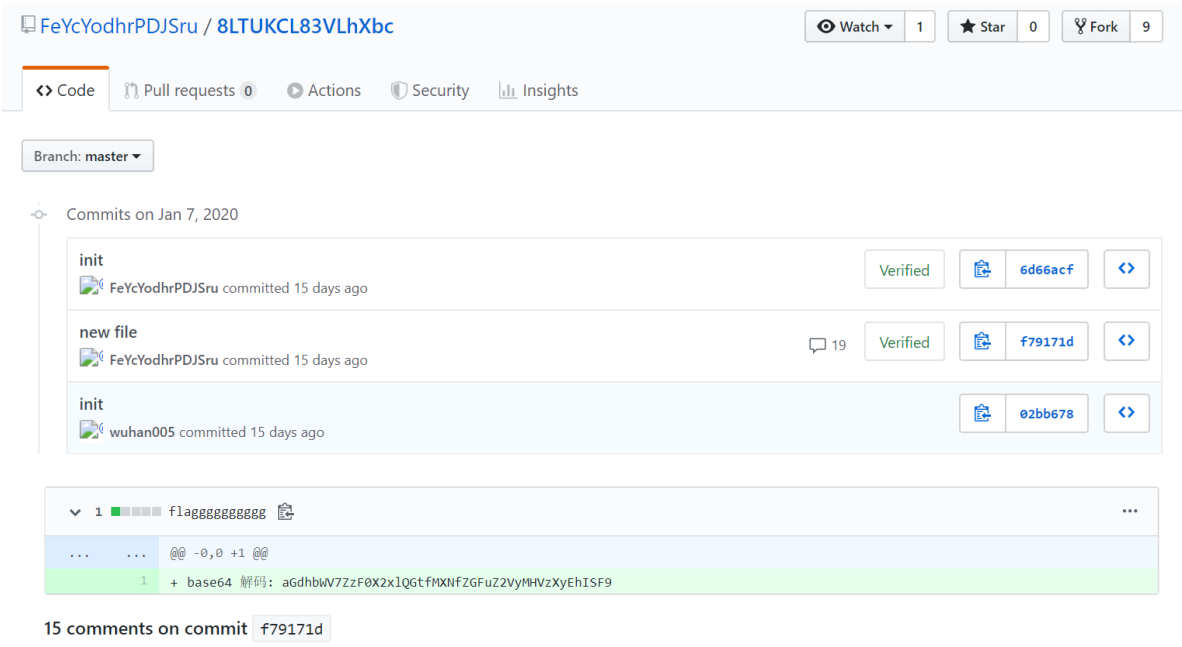
cosmos.hgame.n3ko.co/.git/

404 Not Found

一开始以为思路不对，但是好在没有放弃，手动遍历了/.git/下的目录，e.g. /.git/index和/.git/config等等，然后发现/.git/config成功返回，得到下一步的url，成功找到github页面。

```
[core]
        repositoryformatversion = 0
        filemode = true
        bare = false
        logallrefupdates = true
[remote "origin"]
        url = https://github.com/FeYcYodhrPDJSru/8LTUKCL83VLhXbc
        fetch = +refs/heads/*:refs/remotes/origin/*
```

直接进入历史版本commits，~~第一眼就看到前面大佬留下的10个指路评论~~，十分愉快地了打开new file文件。



得到一串base64编码，解码后得到flag。

Code World

打开题目，403？？？一开始以为题目挂了，但是打开源码发现提示有302重定向，以及出题人留下的hint...XD

```
<html>
    <head><title>403 Forbidden</title></head>
    <body bgcolor="white">
        <center><h1>403 Forbidden</h1></center>
        <hr><center>nginx/1.14.0 (Ubuntu)</center>
    </body>
    <script>
        console.log("This new site is building....But our stupid developer Cosmos did 302 jump to this page..F**k!")
    </script>
</html>
```

一开始想谷歌一下找到重定向的上一个地址的方法但是对于这题貌似无果，此时突然想起之前看过一篇wp里一句话让我印象深刻，“要想为什么访问这个地址第一个页面不是index.php而是其它的？”于是在bp里抓包试着将url改为/index.php，发现状态码变为405 Method Not Allowed。

```
<html>
<head><title>405 Not Allowed</title></head>
<body bgcolor="white">
    <center><h1>405 Not Allowed</h1></center>
    <hr><center>nginx/1.14.0 (Ubuntu)</center>
</body>
</html>
```

得知是请求方式不对后，将GET改为POST得到回显，到这里终于知道自己思路没错了（超开心。

```
<center><h1>人鸡验证</h1><br><br>目前它只支持通过url提交参数来计算两个数的相加，参数为a<br><br>现在,需要让结果为10</center>
```

根据提示先尝试直接将url改成/index.php?a=1+9发现不太正确

```
POST /index.php?a=1+9 HTTP/1.1
Host: codeworld.hgame.day-day.work
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/79.0.3945.130 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/s
igned-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close
```

```
HTTP/1.1 403 Not Allowed
Server: nginx/1.14.0 (Ubuntu)
Date: Wed, 22 Jan 2020 03:36:50 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Location: new.php
Content-Length: 190

<center><h1>人鸡验证</h1><br><br>目前它只支持通过url提交参数来计算两个数的相加，参数为a<br><br>现在,需要让结果为10<br><br><h2>再想想？ </h2></center>
```

再次审题：只支持通过url提交参数，如果这里的url不只是指参数的提交方式，还指url编码呢？于是将1+9进行url编码后再通过url提交，得到flag。

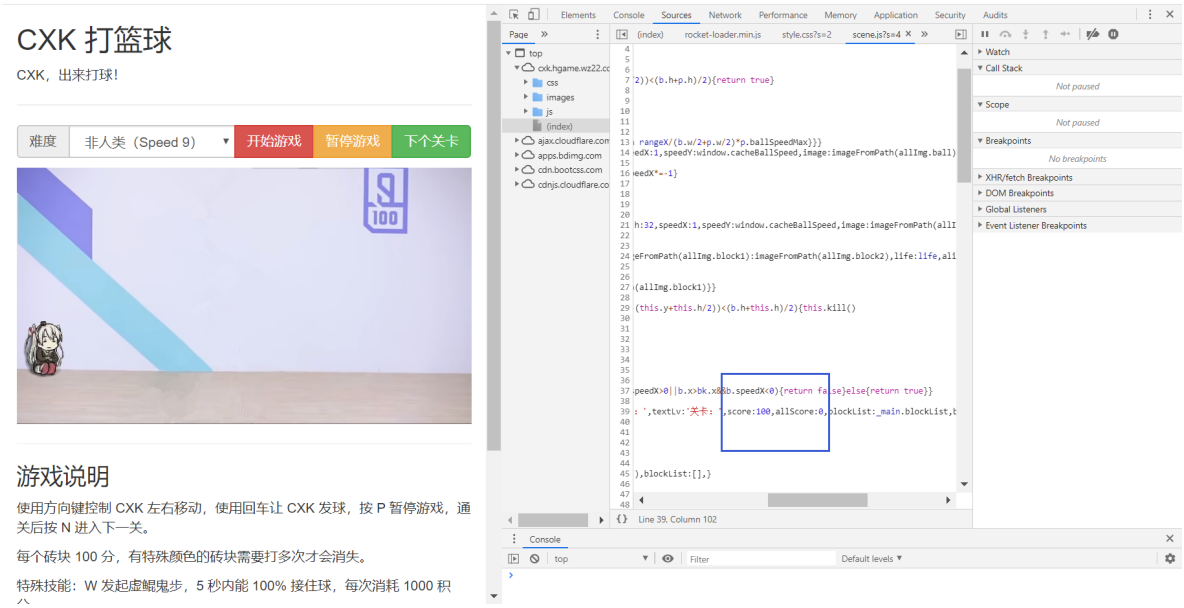
```
POST /index.php?a=1%2b9 HTTP/1.1
Host: codeworld.hgame.day-day.work
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleV
Chrome/79.0.3945.130 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/we
igned-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close
```

尼泰玫

打开游戏网站，玩上它个几局，哇塞好好玩，（一晚上过去后）我要干什么来着...噢噢对，拿flag（挠头

球掉了之后有弹窗提示分数要到30000分，八成就是要找到储存分数的变量然后修改。

顺着这个思路F12打开Source并使用肉眼法观察js文件，找到了两个跟分数有关的变量



于是将allscore修改成大于30000的数后保存，开始游戏，得到弹窗flag

接头霸王

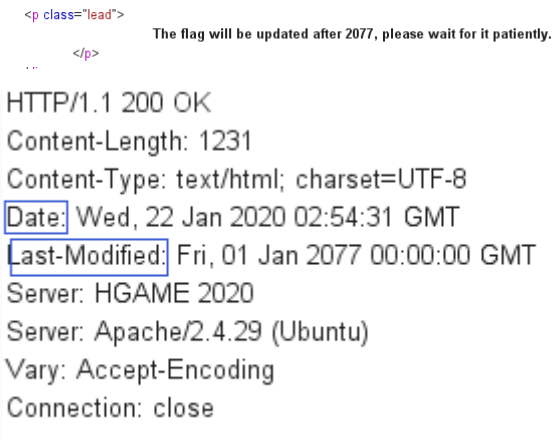
由提示可以想到要修改请求头来源地址，打开web好帮手burp suite截包改头，加入Referer: https://vidar.club/，得到回显

```
<p class="lead">
        You need to visit it locally.
    </p>
```

localhost访问，则添加X-Forwarded-For:127.0.0.1

```
<p class="lead">
        You need to use Cosmos Brower to visit.
    </p>
```

得到提示要用Cosmos Brower访问，既然不能让Cosmos学长弄一个浏览器出来让我们用，那只能在User-Agent中添加Cosmos Brower。下一步提示我们需要修改请求方式，则在bp中将GET改为POST。最后来到~~本周全场最狗的地方~~：提示我们要在2077年访问！



观察响应头有Last-Modified，第一反应是在请求头上加上If-Modified-Since，修改无果后又尝试了把和时间有关的请求头响应头都修改了一遍，（就是完美避开了正确答案）。卡了一天后，在学长处得到 修改头 的思路是对的提示后，~~再结合茄子学长的尿性~~，不抱希望地尝试了第一个被排除的修改方式：在请求头添加If-Unmodified-Since，结果得到了flag...（人都傻了

Misc

欢迎参加HGame!

Li0tIC4uLi0tIC4tLi4gLS4tLiAtLS0tLSAtLSAuIC4uLS0uLSAtIC0tLSAuLi0tLi0gLi4tLS0gLS0tLS0gLi4tLS0gLS0tLS0gLi4tLS4tIC4uLi4gLS0uIC4tIC0tIC4uLi0t

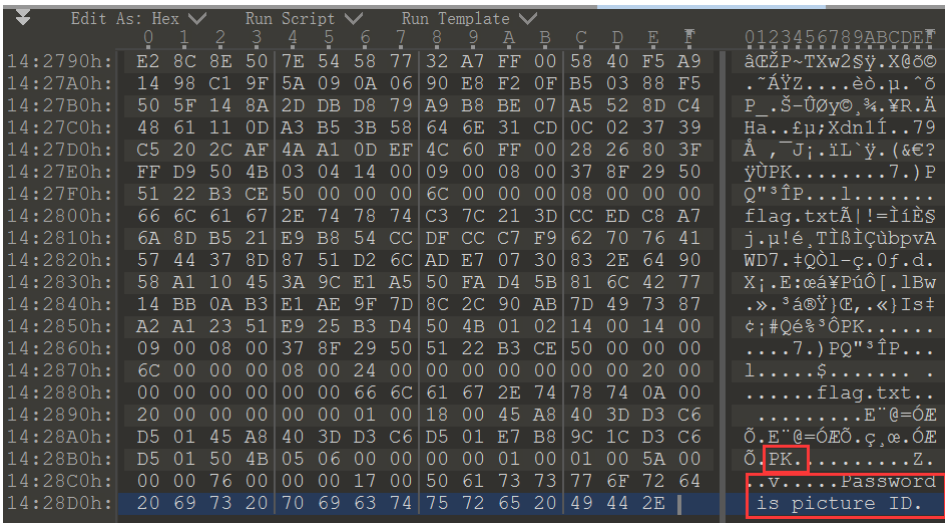
一开始看到这长长一大串重复的字符串完全没思路（看到Li0开头第一反应就是！），于是 **遇事不决Base64**，解出来得到摩斯密码

.-- ...-- .-.. -.-. ----- -- . ..--.- - --- ..--.- ..--- ----- ..--- ----- ..--.- .... --. .- -- ...--

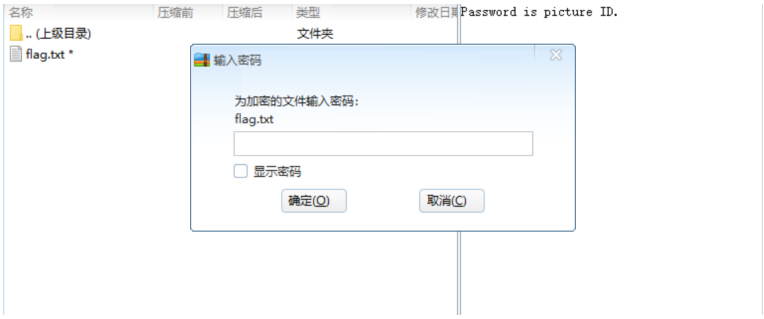
解码得到flag，签到成功！

壁纸

得到一张ObjectNotFound学长老婆的图片，直接丢进010 Editor里看有什么。

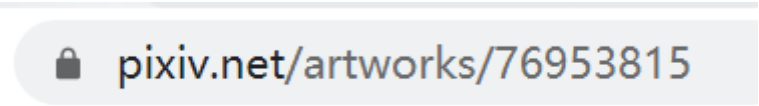


发现两个关键信息，PK...说明是个zip格式以及它的密码是图片id，于是将文件格式改成zip



需要密码解锁压缩文件，在16进制文件里没有找到可以入手的地方，于是思考如果一张图片只存在本地是不需要id的，但是如果放到某些网站……于是乎将图片丢到谷歌搜图中，加上文件名中pixiv的限制

找到图后，根据多年逛站经验知道id一般都会在url中，8位数字即为图片id和zip的密码



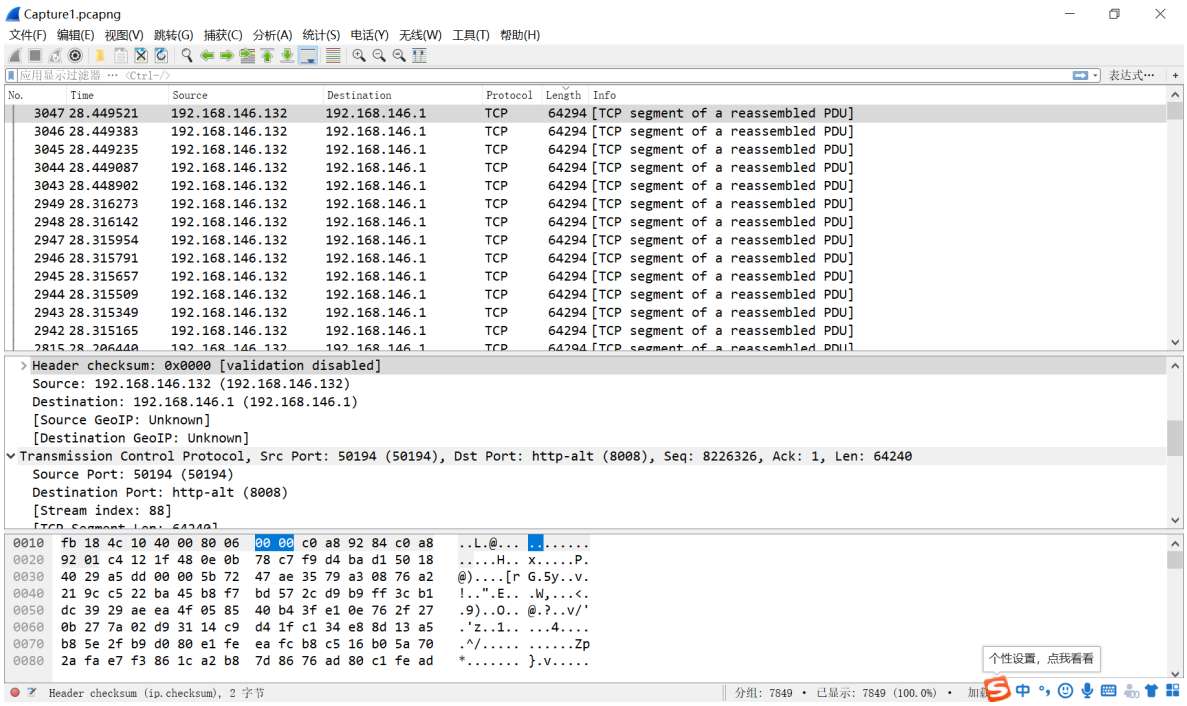
输入密码到zip中得到txt文件内容

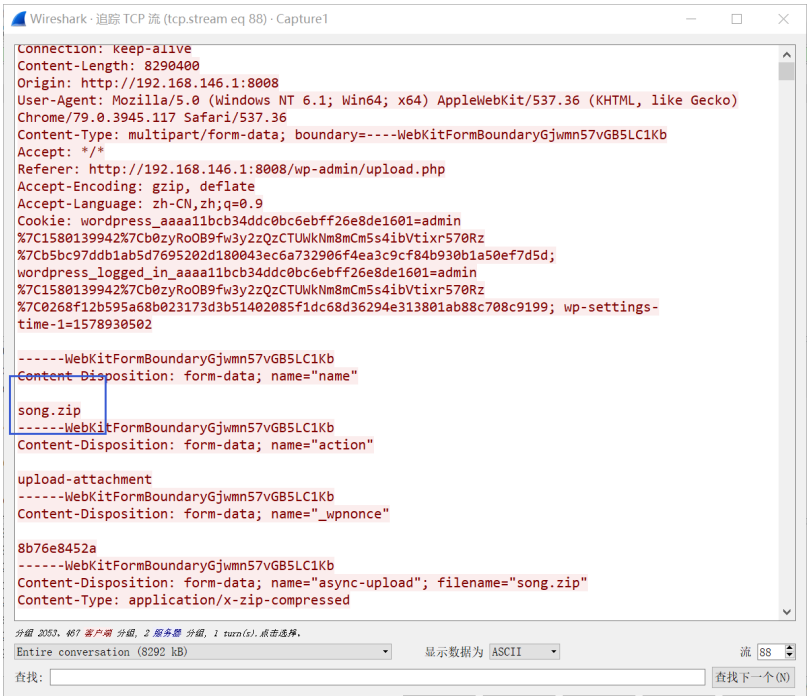
\u68\u67\u61\u6d\u65\u7b\u44\u6f\u5f\u79\u30\u75\u5f\u4b\u6e\u4f\u57\u5f\u75\u4e\u69\u43\u30\u64\u33\u3f\u7d

由Unicode编码解码得到flag

每日推荐

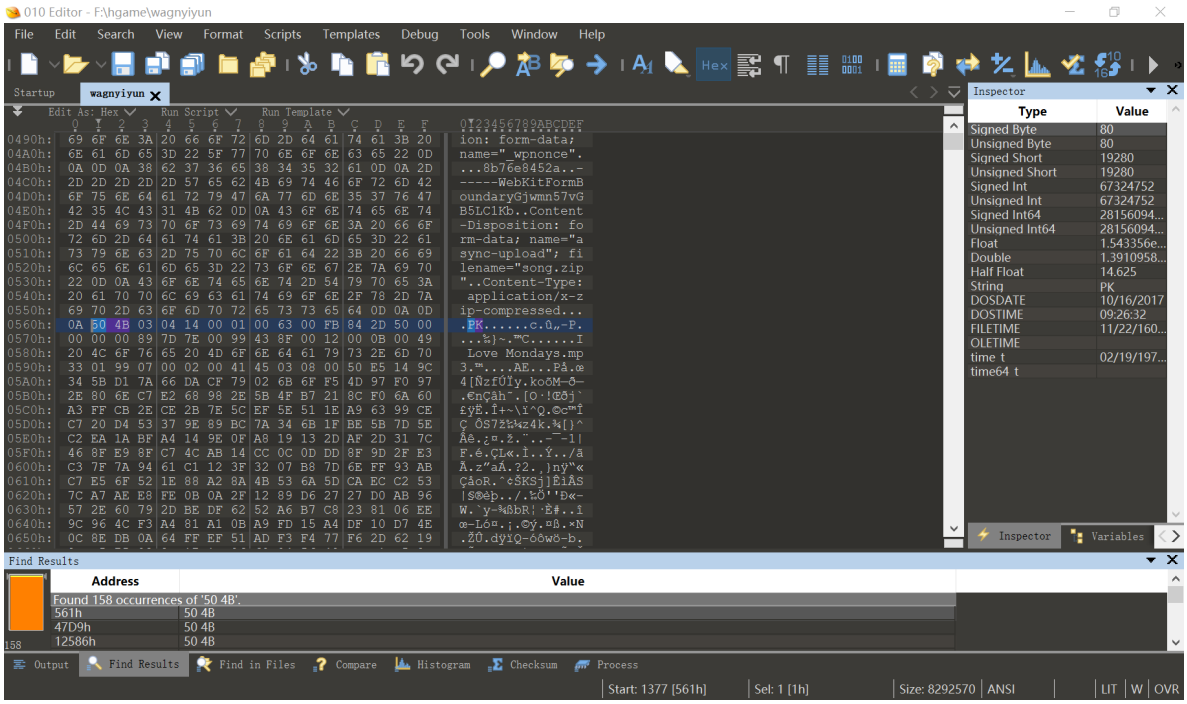
下载文件用wireshark打开pcapng，根据题目中说到ObjectNotFound学长在听歌，流量应该不小，所以将数据包按长度排序，找到最长的并跟踪TCP流





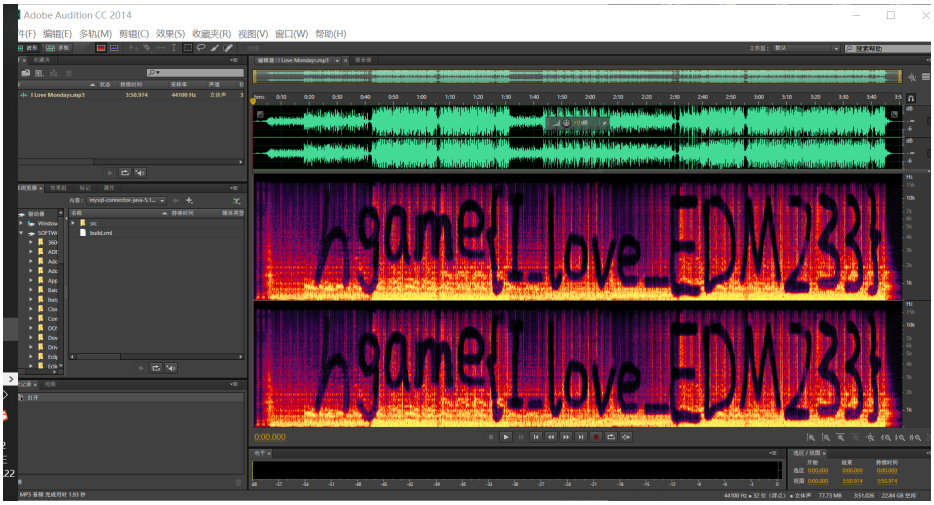
发现数据中含有song.zip，猜测flag就在这个压缩包里。

将数据包内容保存到本地并用010editor打开，google查询zip文件的头尾标识然后在010editor中去头去尾得到zip文件



打开发现压缩包需要密码，提示是6位数字，一开始在wireshark和十六进制文件中找寻无果之后，想着只有六位不如直接爆破

成功得到密码，解压出mp3文件。再丢到010editor和binwalk中没发现能下手的地方，没有思路，就只能开始听歌去做别的题qaq……然后听着听着发现中间会有几段的声道会突然改变，而我一开始在网易云听到的版本并不是这样的！第一反应丢进AU里看看，同时参考一些wp发现频谱里可能会有猫腻，结果就得到了flag。



签到题ProPlus

根据提示3栅栏密码，5凯撒密码，用对应工具解码后得到一串先是英文句子然后是压缩包密码。

Rfsd djfwx qfyjw fx mj kfhji ymj knwnsl xvzfi, Htqtsjq Fzwjqnfst Gzjsinf bfx yt wjrjrgjw ymfy inxyfsy fkyjwstts bmjs mnx kfymjw yttp mnr yt inxhtajw nhj.

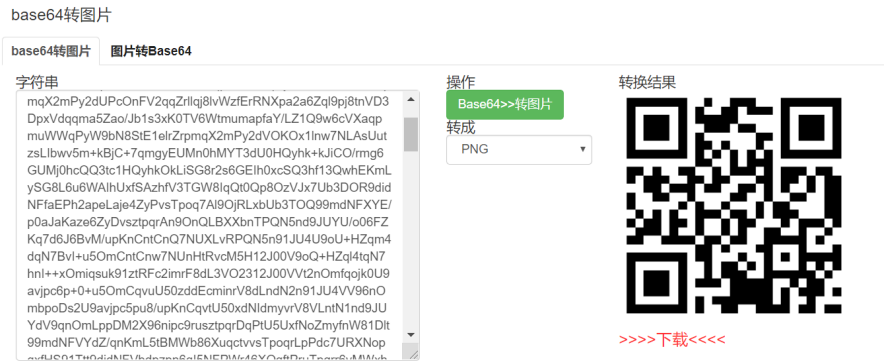
JFARZGFVMVRAJUIY

位移 5 加密 解密

Many years later as he faced the firing squad, Colonel Aureliano Buendia was to remember that distant afternoon when his father took him to discover ice.

EAVMUBAQHQMVEPDT

解压后看到一大串Ook，谷歌一下知道是Ook加密，丢到工具中解密得到base32编码的密文，解码后得到base64编码，利用工具转为图片得到二维码，扫码后出现flag



Crypto

InfantRSA

根据题目知道这是一道RSA加密题，到谷歌上现学现卖一下算法原理，然后找到相应的py解密算法拿来改一改

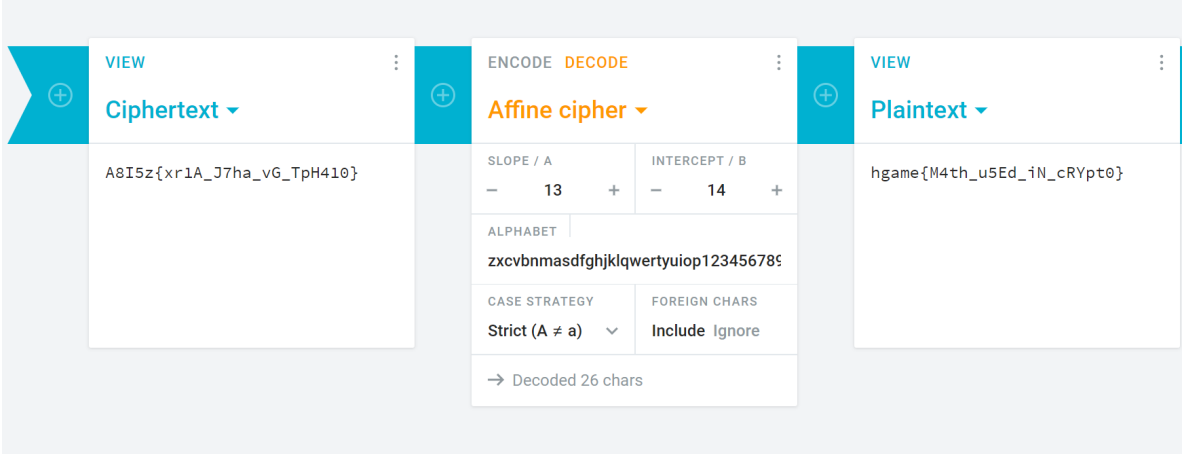
```
import gmpy2
import binascii

p = gmpy2.mpz(681782737450022065655472455411)
q = gmpy2.mpz(675274897132088253519831953441)
e = gmpy2.mpz(13)
phi_n = (p-1)*(q-1)
d =gmpy2.invert(e,phi_n)

c = gmpy2.mpz(27569846508236107014517368841149631154217290260855985901984l)
m = pow(c,d,p*q)
m_hex = hex(m)[2:]
print("ascii:\n%s"%(binascii.a2b_hex(m_hex).decode("utf8")))
```

```
gamison@LAPTOP-4B5UO0VR:~/src/hgame$ python infantrsa.py
ascii:
hgame{t3Xt600k_R5A!!!}
```

（有现成的就不自己动手写orz

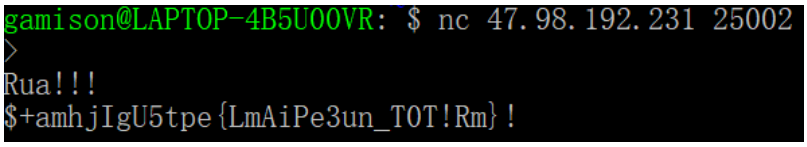
Affine

先谷歌一下affine是啥，了解一下原理，直接利用现成网站工具手动得到flag（先mark一下，以后再研究解密的算法...

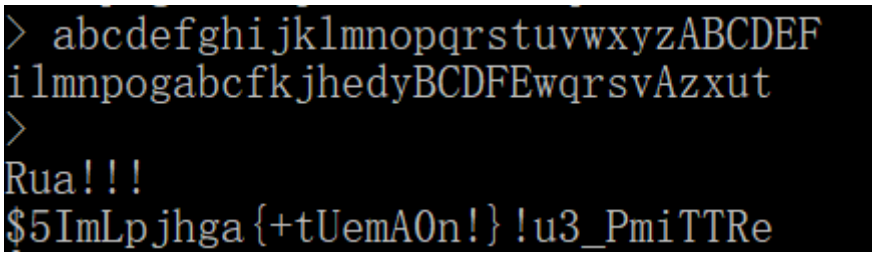


Reorder

先nc一下，得到一个非常像flag的字符串，再结合题目大概知道是要把这个字符串重排序



然后用不同字母尝试几次，发现是通过一定的位移得出密文，而每一次nc的位移算法都是不一样的，所以可以通过用和密文相同长度的字符串得到位移量



再根据位置一一对应回去即得到flag