WUST-CTF 2020 WriteUp

原创

WustHandy 于 2020-03-30 21:42:35 发布 2015 收藏 5

分类专栏： WriteUp 文章标签： 信息安全

本文链接：https://blog.csdn.net/weixin_45883223/article/details/105193948

WUST-CTF 2020 WriteUp

前言

本文无任何跳步，过程十分详细，面向零基础的萌新（我也是萌新
这次又被“面向萌新，题目友好”给骗了，我还是tcl。

Web

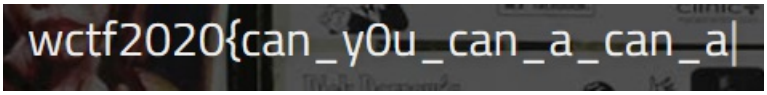
checkin

打开链接发现要提交作者的名字，但是提交键无法点击

于是按F12查看网页源代码找到提交键的元素，发现了disabled=“disabled”，把它删除后发现无法输入作者“52Hertz”，发现有maxlength="3"的限制，把3改成大数后就可以点击提交了。

弹出来一个框，输入url进入博客。

在主页发现了滚动的前半段flag，接着在博客里找后半段。



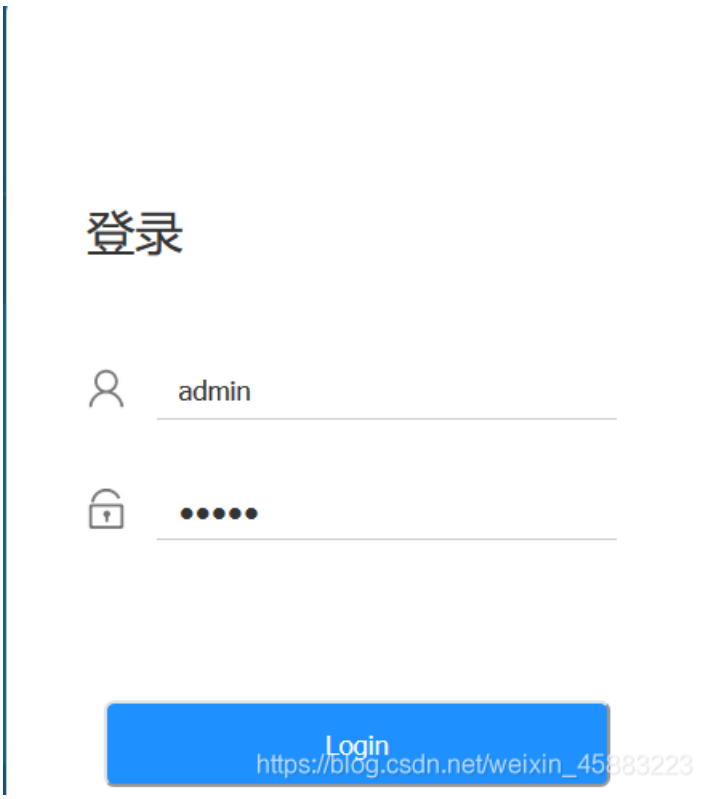
文章翻到最后一页的最后一篇博客发现了“远古的blog”

进入后翻到最下面找到了后半段，拼接即可。

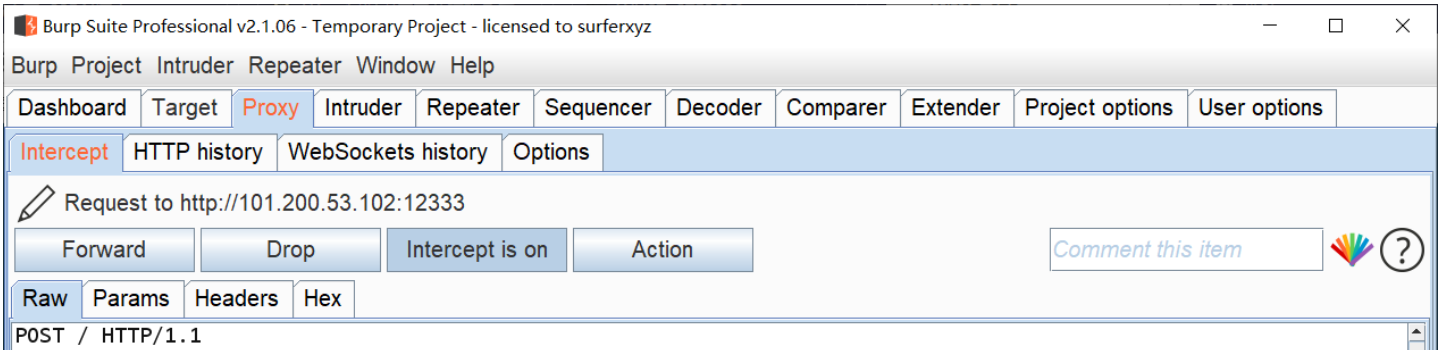
Here is your flag: 5_a_c@nner_can_Can_@_can}

admin

第一反应居然是sql注入。。。这里放的是我的奇yin方法，当时没想到要用万能密码（例如'or 1#）



随便输个账号密码，用Burp Suite（以下简称BP）抓包

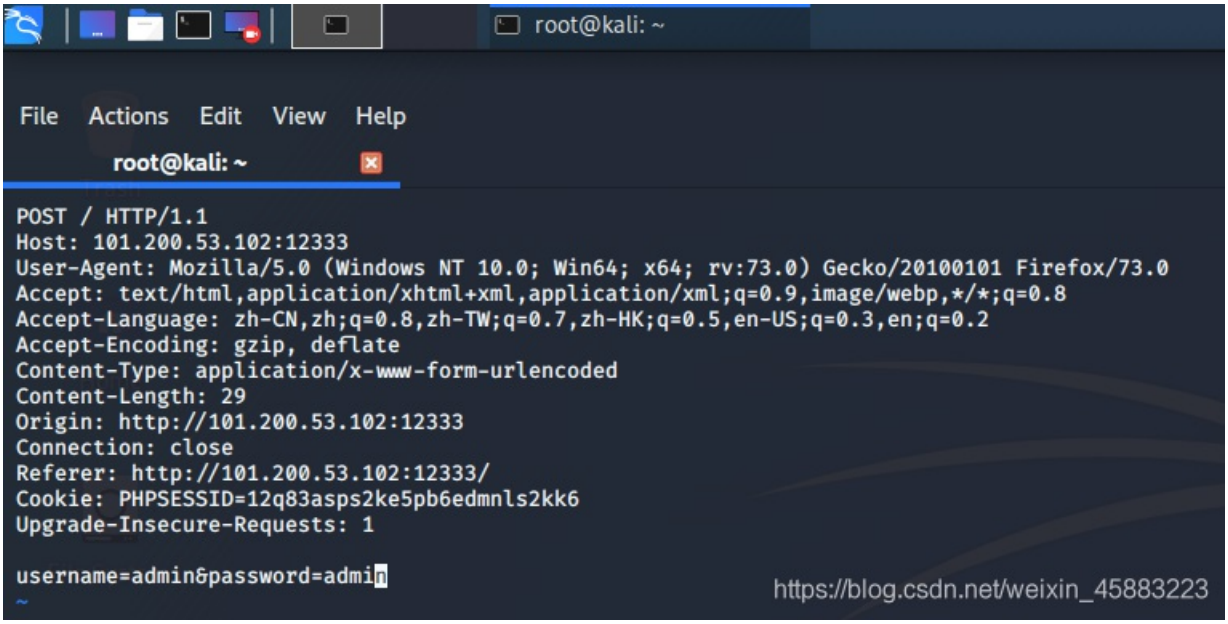


```
Host: 101.200.53.102:12333
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:73.0) Gecko/20100101 Firefox/73.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 29
Origin: http://101.200.53.102:12333
Connection: close
Referer: http://101.200.53.102:12333/
Cookie: PHPSESSID=stg1j3kcis8be9129hq79dna04
Upgrade-Insecure-Requests: 1

username=admin&password=admin
```

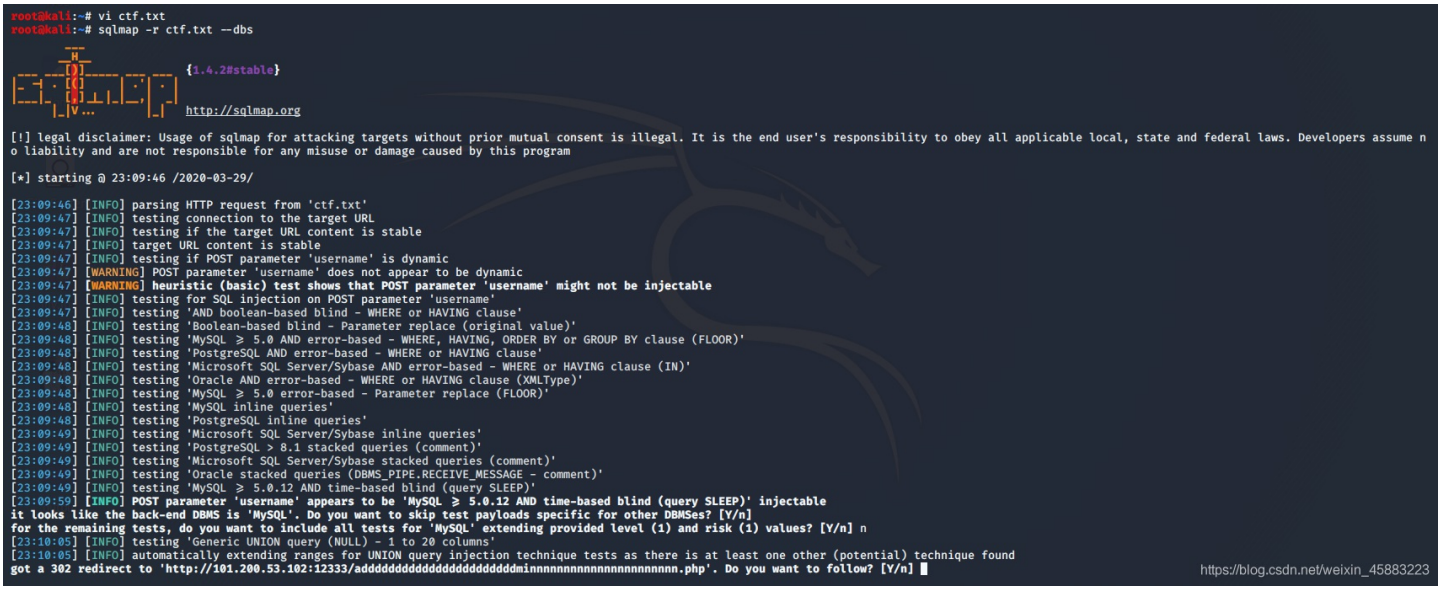
< + > Type a search term 0 matches

把包复制粘贴到kali linux的文本文档里并保存



sqlmap一下发现了

一个302的url



访问这个url，发现要本地ip，所以在BP抓的每一个包里都加上 X-Forwarded-For: 127.0.0.1

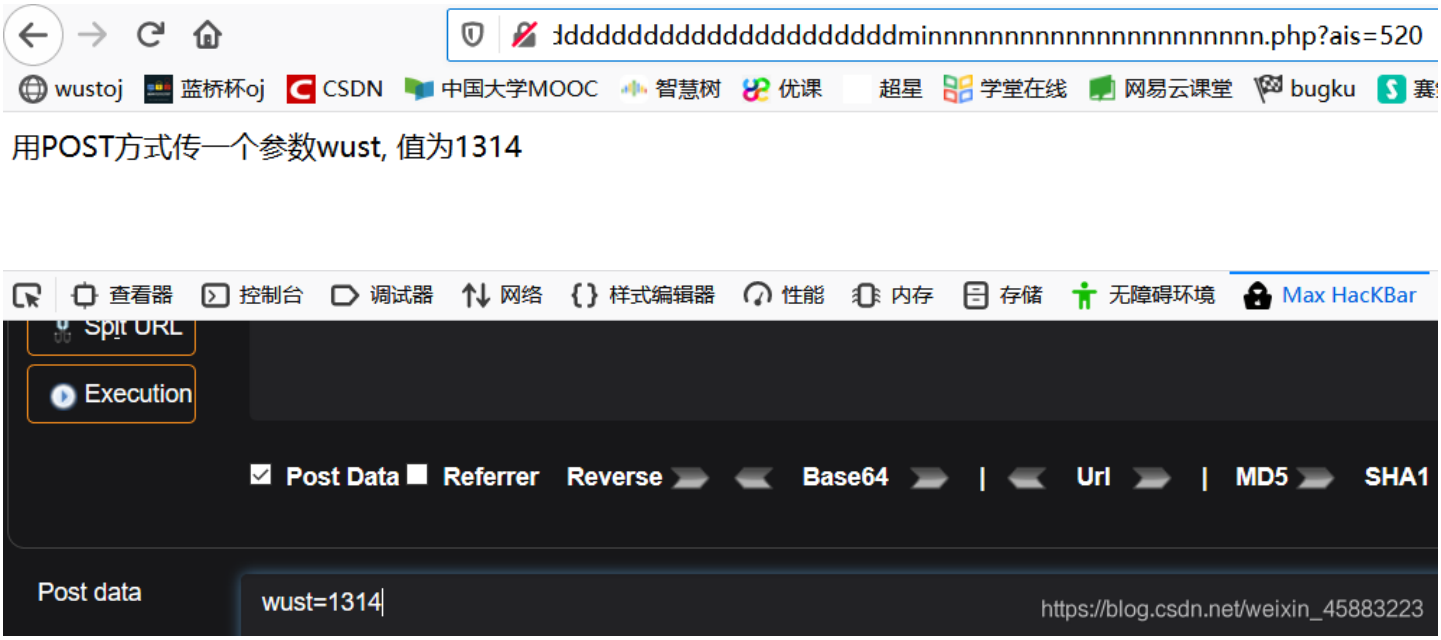
必须本地ip才能访问

```
GET /adddddddddddddddddddddddminnnnnnnnnnnnnnnnnnnnn.php HTTP/1.1
Host: 101.200.53.102:12333
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:73.0) Gecko/20100101 Firefox/73.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Cookie: PHPSESSID=stg1j3kcis8be9129hq79dna04
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
X-Forwarded-For: 127.0.0.1
```


GET请求，构造?ais=520

用GET方式传一个参数ais, 值为520

POST请求，用Max HackBar 勾选Post Data wust=1314 Execution



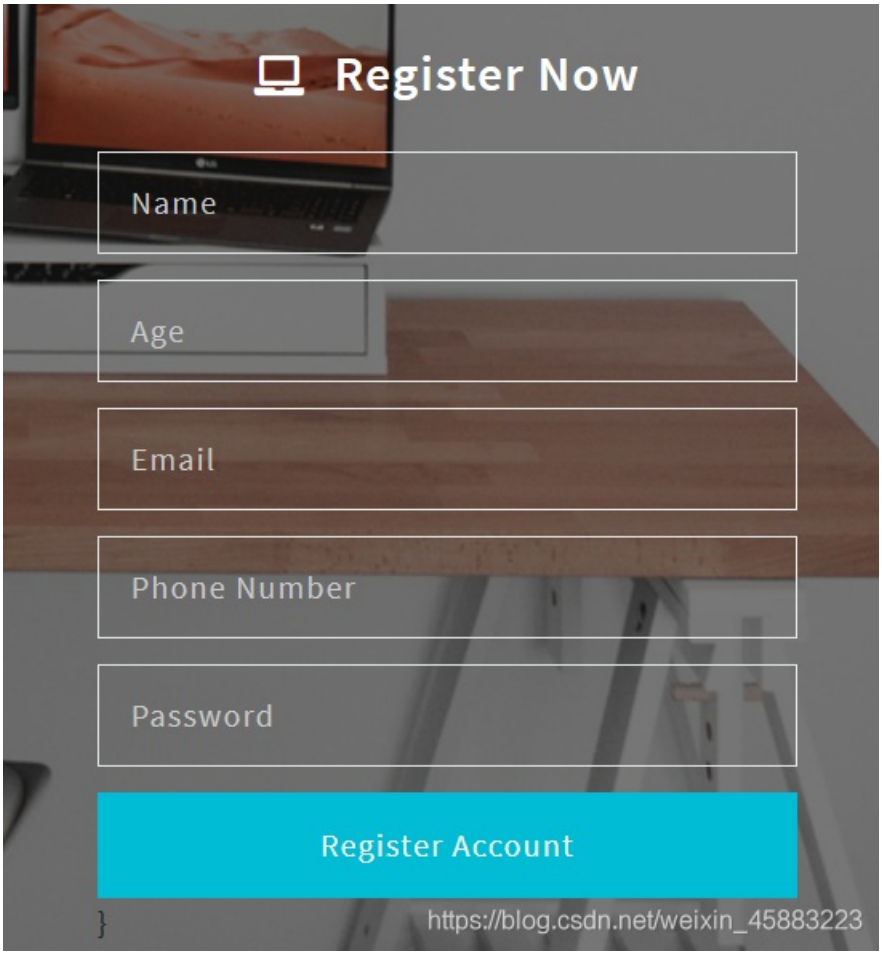
p和aste应该连起来，剩下的4dz,Rqr,cSf2排列组合即可

你离flag已经很近了，网址给你了：4dz aste.ubuntu.com/p/ https://p Rqr cSf2

把得到的这段base64解码即可

CV Maker

先注册一个账号并登录



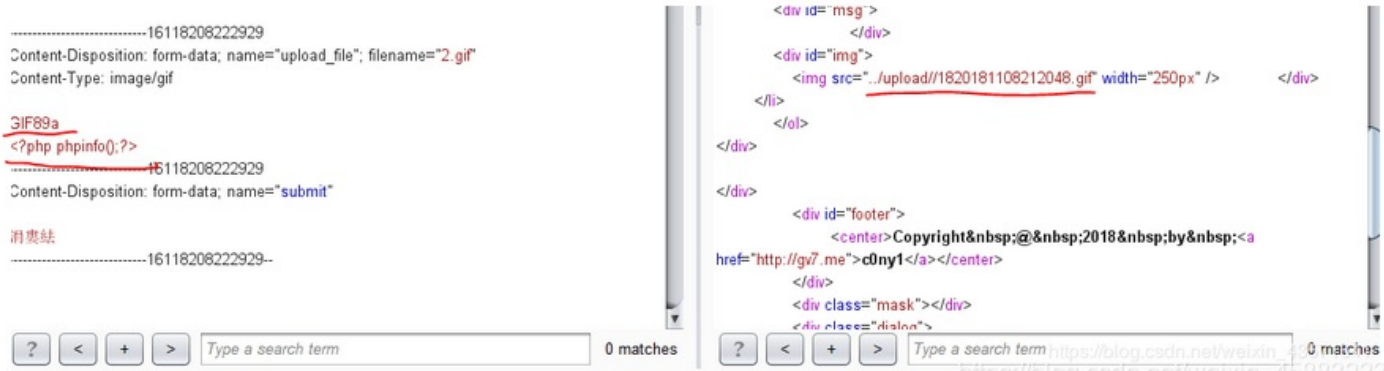
只有这个更换头像的地方可操作，看出是文件上传，试试一句话木马行不行

更改content-type为image/jpg等操作都无法上传成功，查一下exif_imagetype

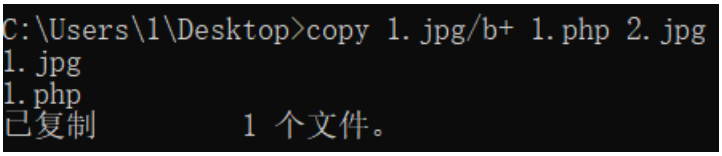
找到CSDN里有这样一篇博客

分析，该代码通过exif_imagetype判断文件类型
通过图片马进行上传绕过
制作图片马

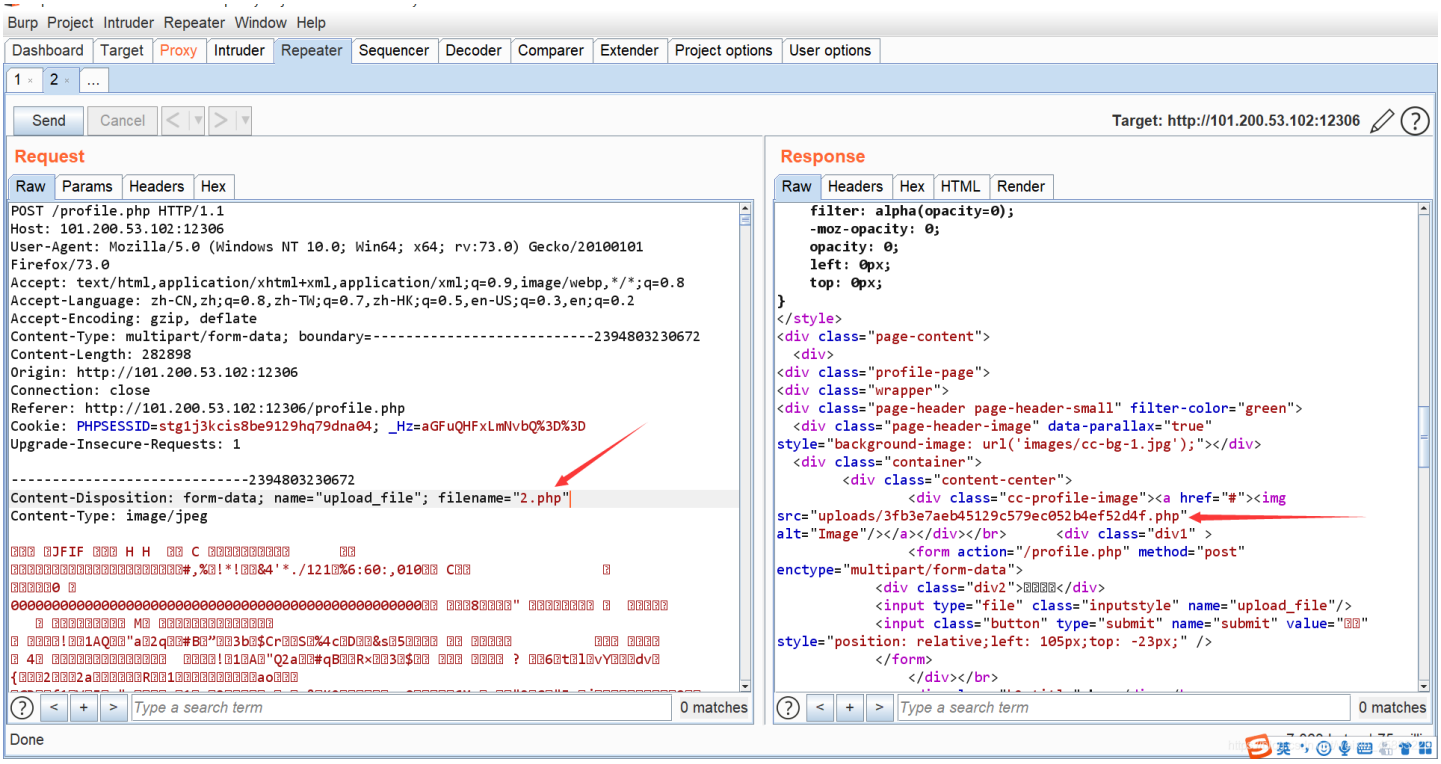
接着用bp上传文件



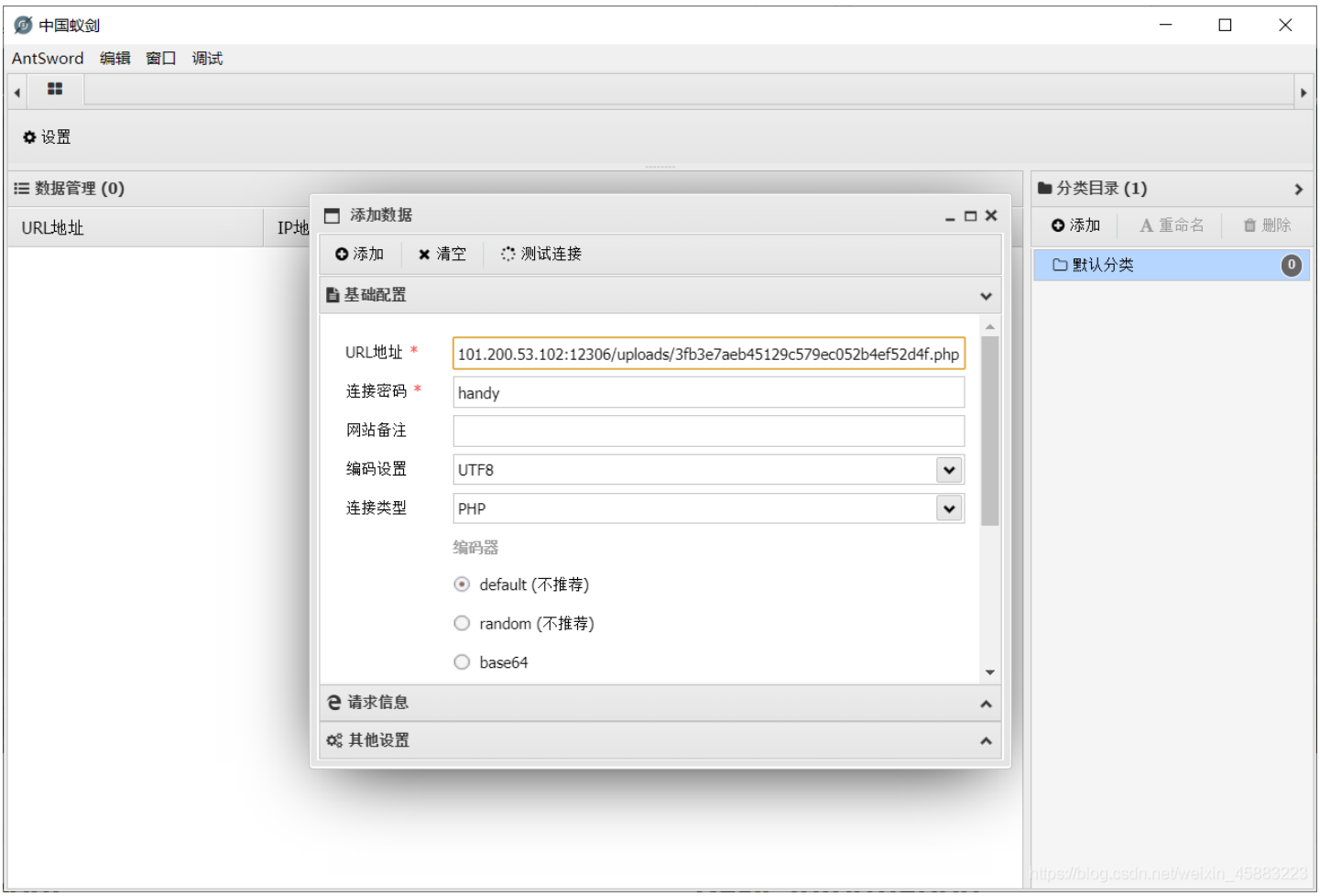
制作图片马



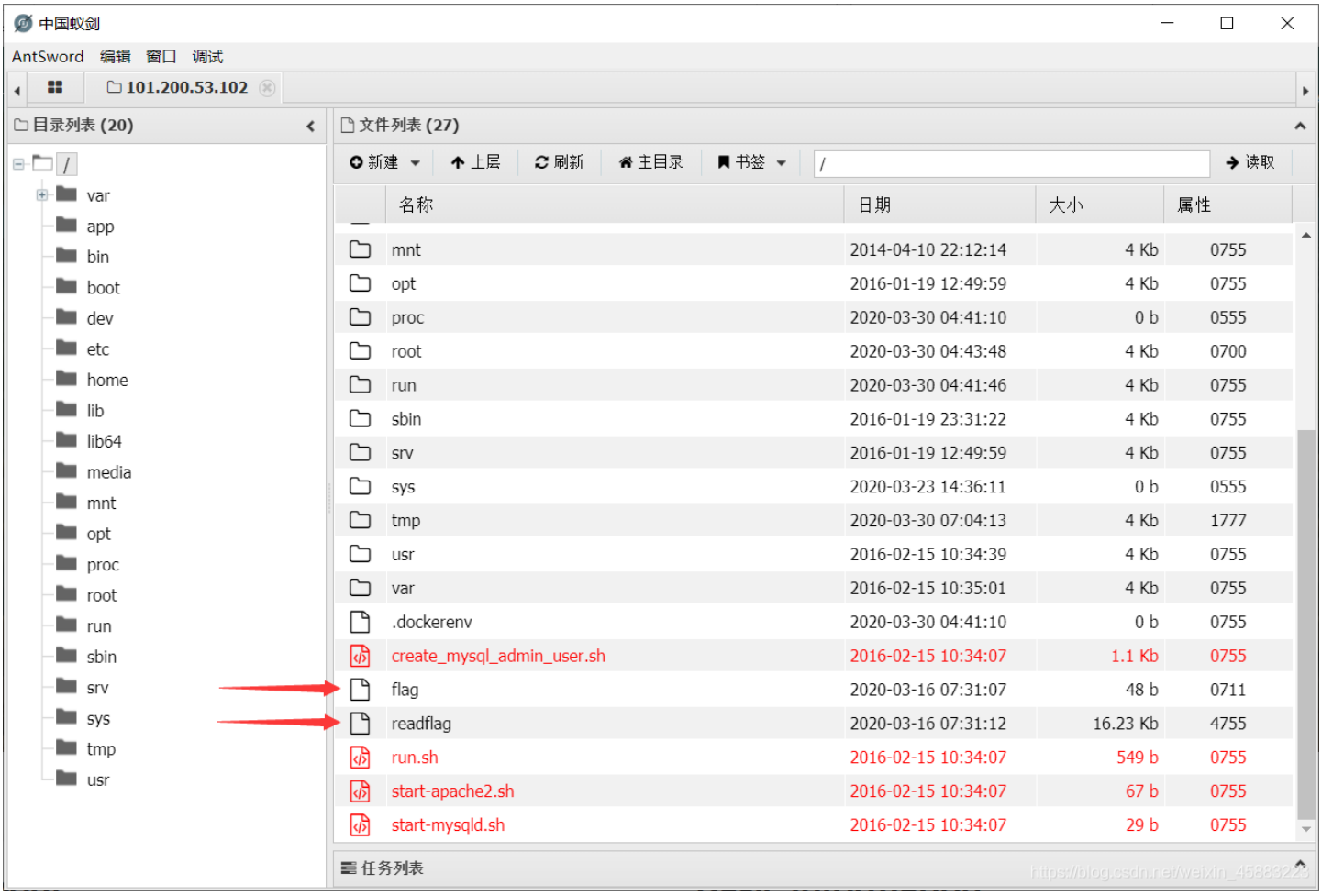
上传2.jpg，抓包，send to repeater，把filename的后缀改为.php，send，在response里找到了上传的回显路径



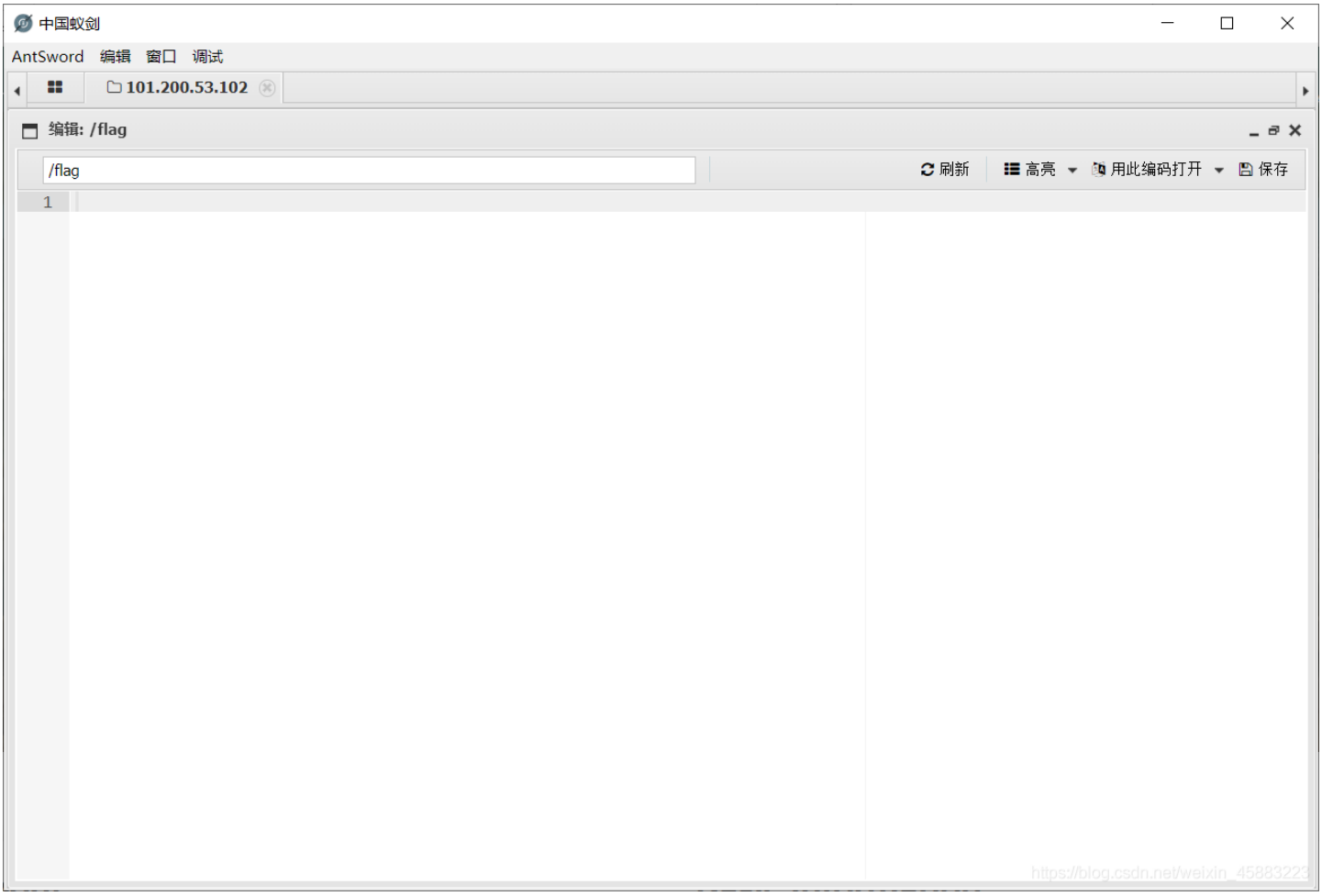
打开中国蚁剑进行连接，url为那个路径，密码为一句话木马的POST里的那个



刚开始在html文件夹里翻了半天flag在哪（受ctfhub影响），最后在根目录里找到了flag和readflag

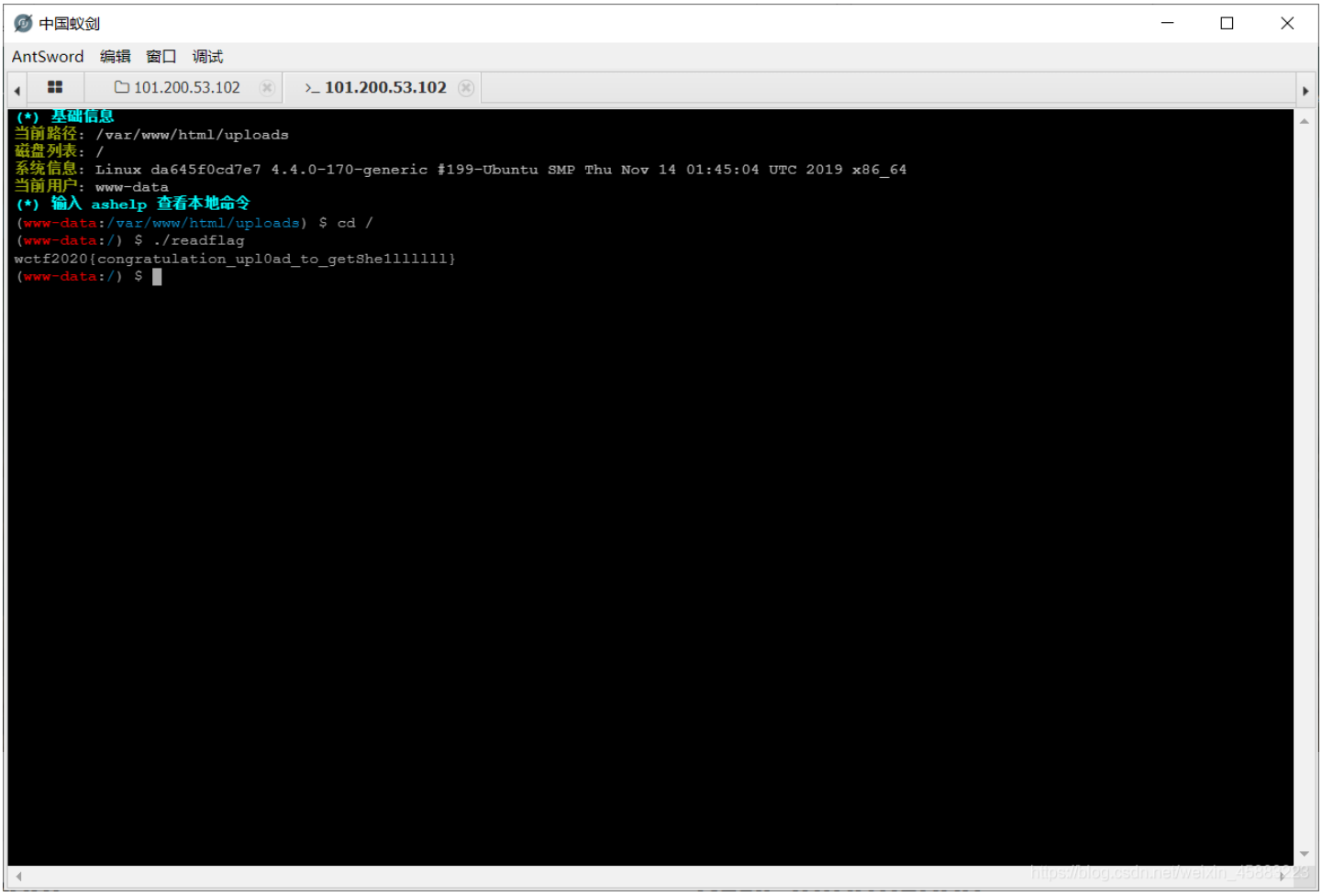


打开flag发现是空的



再打开readflag，是一堆乱七八糟的

于是打开终端使用命令 ./readflag 执行文件得到flag

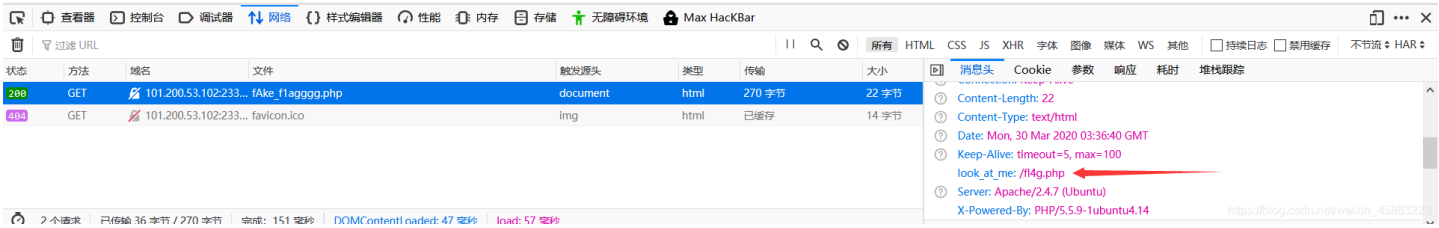


朴实无华

找不到什么注入点，sqlmap也无果，打开robots.txt看看发现了 /xxx.php

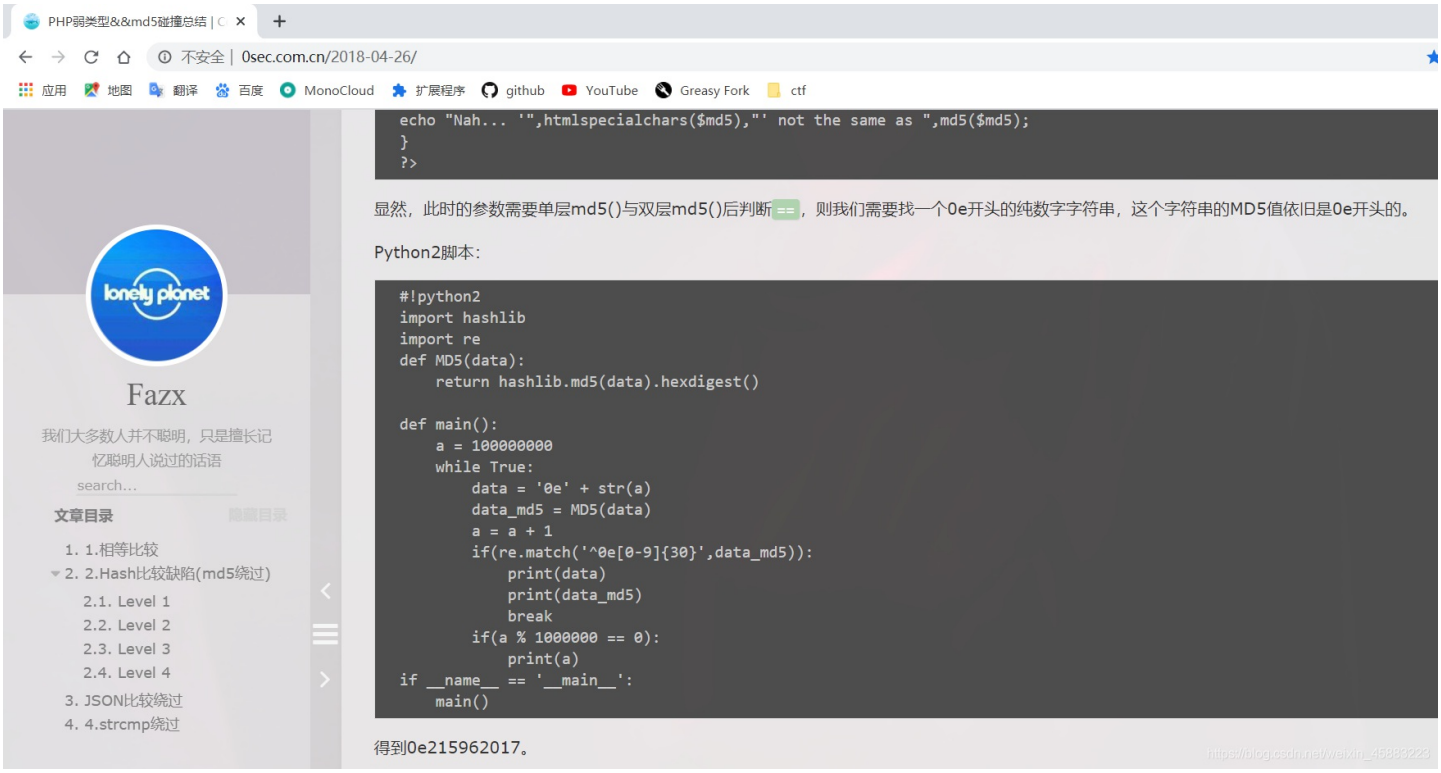
进去看看有一个假flag

按F12点网络看消息头，发现了look_at_me，进入/fl4g.php



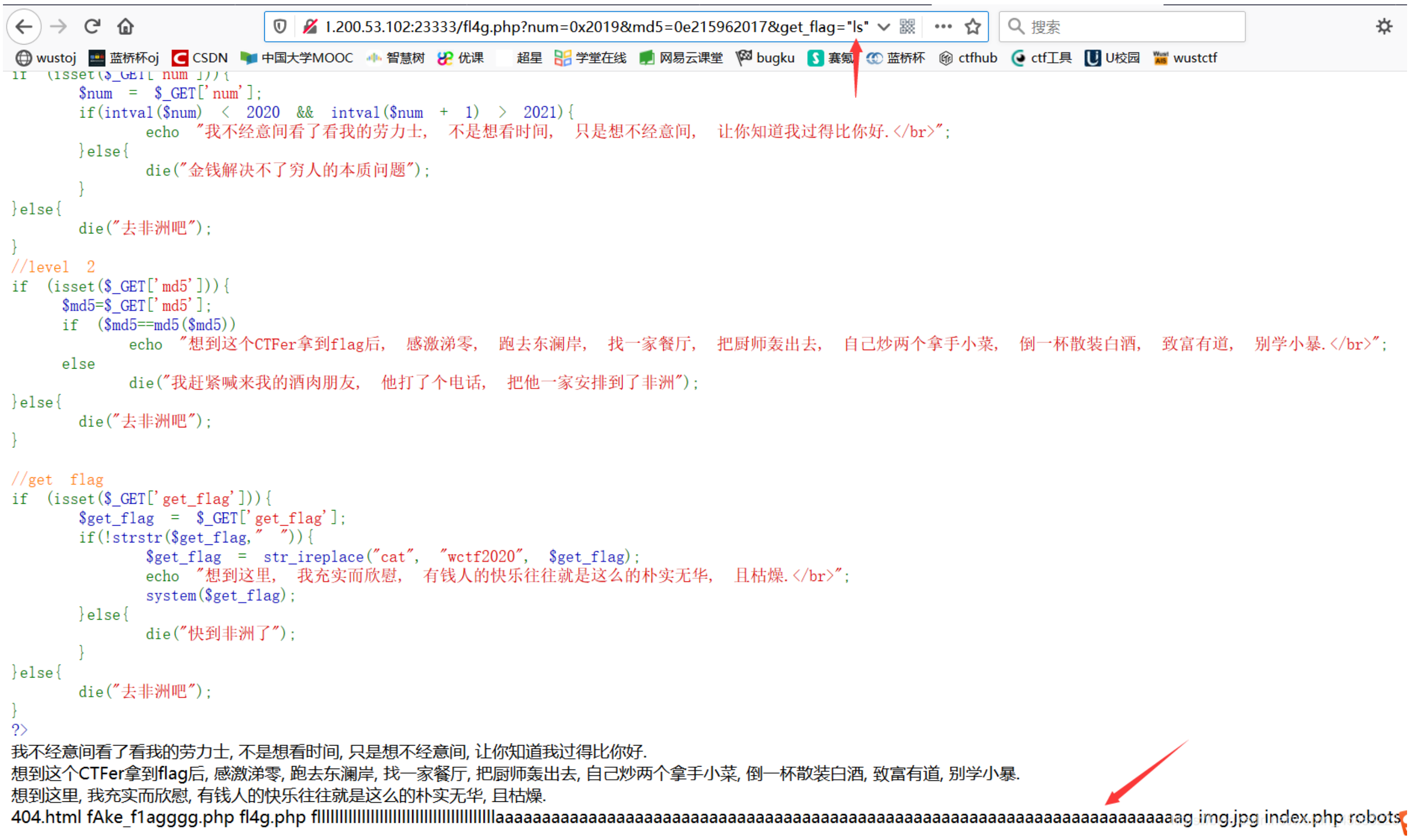
全是php代码，审计一波，有三个部分，第一个num要<2020，+1后还要>2021，查一下，php，弱类型，十六进制，构造num=0x2019成功了

第2步要求md5加密前后相等，php把科学计数法0e后面全是数字的全当作0，Google到了这篇博客，piao过来一个md5=0e215962017



最后一步：!strstr…可知get_flag里不能有空格，即空格被过滤了，要绕过，有<>,${IFS},%09等姿势；再看str_ireplace这句可知get_flag里面的cat都会被换成wctf2020，又要绕过；再看有system是执行get_flag里的命令语句，所以先用ls看一下目录(不用加"")

cat用别的命令来代替，比如nl,tac等



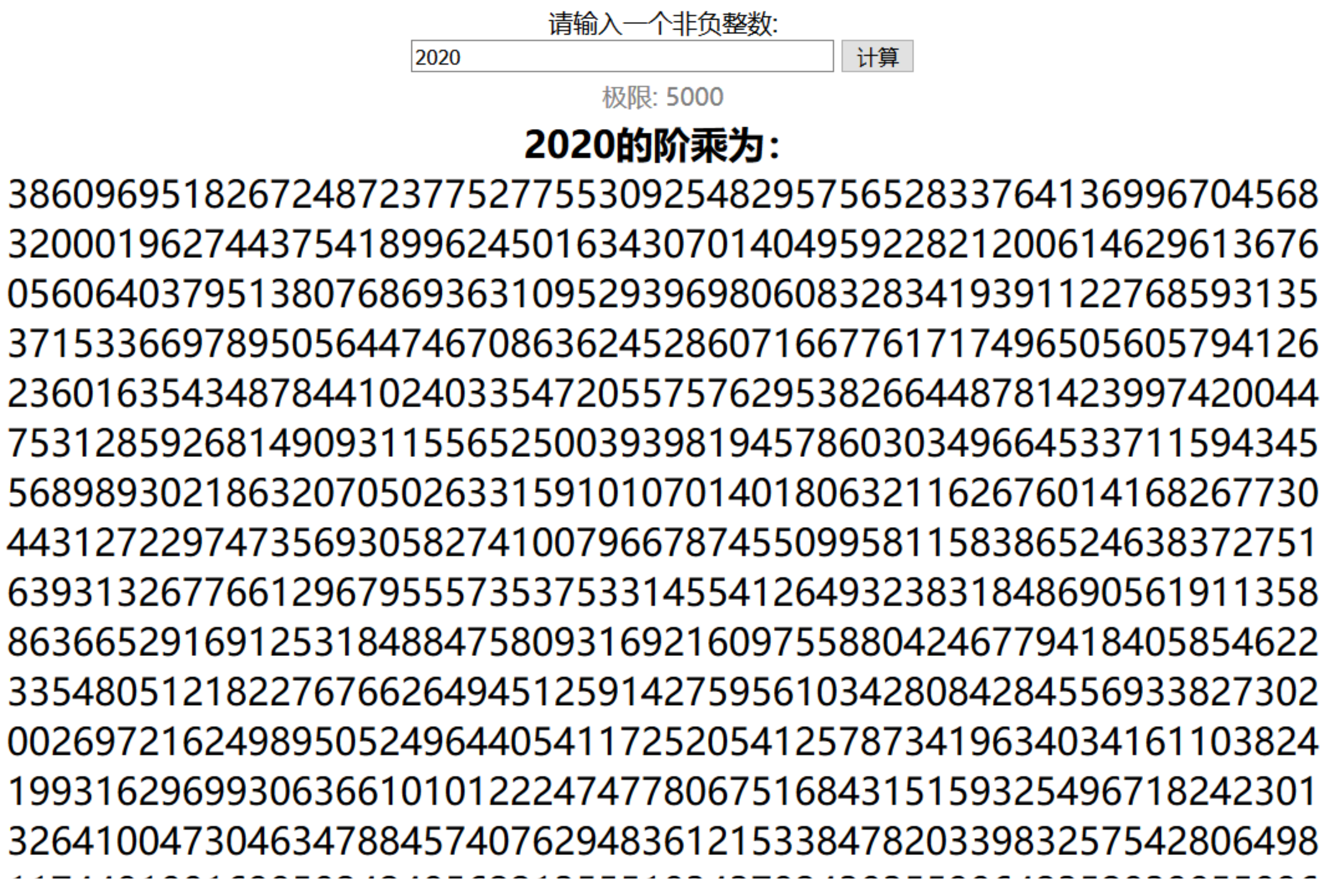
Crypto

大数运算

我也只配拿这种水题的第一解了。。。


Challenge
55 Solves
×
大数运算
100
Author: 52HeRtz
flag等于 wctf2020{Part1-Part2-Part3-Part4} 每一Part都为数的十六进制形式（不需要0x），并用 '-' 连接
Part1 = 2020×2019×2018× ... ×3×2×1 的前8位
Part2 = 520^1314 + 2333^666 的前8位
Part3 = 宇宙终极问题的答案 x, y, z绝对值和的前8位
Part4 = 见图片附件，计算结果乘上1314
Part4.jpg
Flag
Submit
https://blog.csdn.net/weixin_45883223

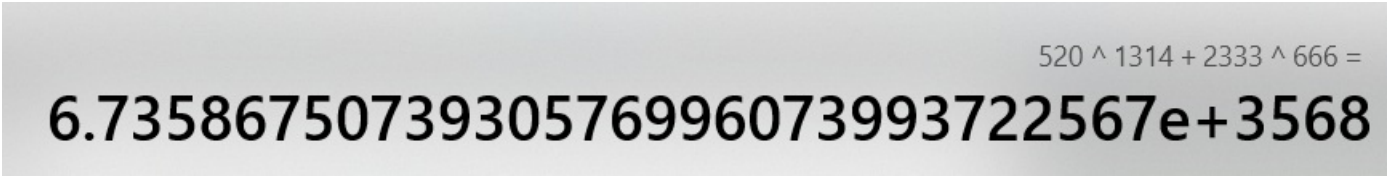
第一步在线算阶乘


请输入一个非负整数:
2020
计算
极限: 5000
2020的阶乘为:
38609695182672487237752775530925482957565283376413699670456832000196274437541899624501634307014049592282120061462961367605606403795138076869363109529396980608328341939112276859313537153366978950564474670863624528607166776171749650560579412623601635434878441024033547205557576295382664487814239974200447531285926814909311556525003939819457860303496645337115943455689893021863207050263315910107014018063211626760141682677304431272297473569305827410079667874550995811583865246383727516393132677661296795557353753314554126493238318486905619113588636652916912531848847580931692160975588042467794184058546223354805121822767662649451259142759561034280842845569338273020026972162498950524964405411725205412578734196340341611038241993162969930636610101222474778067516843151593254967182423013264100473046347884574076294836121533847820339832575428064

```
1174481001698502424856221355518343782430355906423528390550961830475012627097276670238093720719301807238114160366367509212421110772532252914909245456323279250571497160997952297339893
```

前八位转16进制

第二步用电脑自带的计算器



搜索宇宙终极，这篇文章里有两种xyz，第一种试了下不对，第二种对了

宇宙终极数字“42”被破解！X^3+Y^3+Z^3=42是怎么求解的?

语文

数学

英语

物理

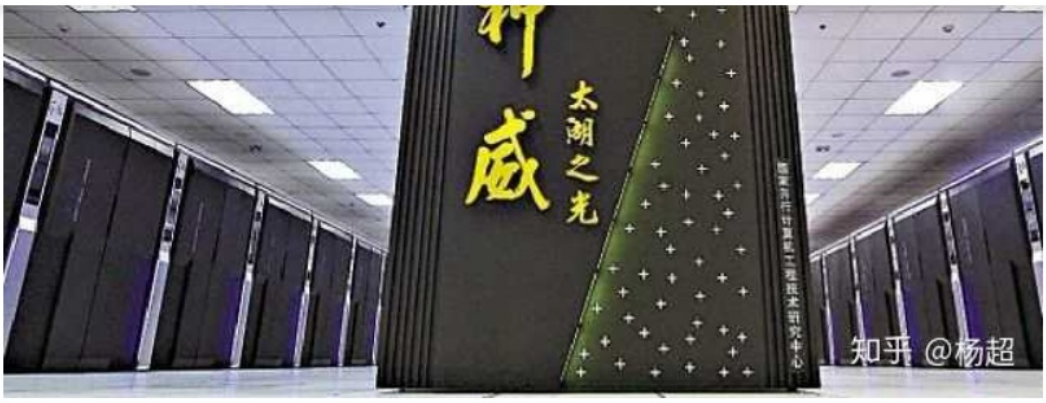
化学

历史

地理

生物

综合



我国自主研发的超级计算机—神威·太湖之光，曾蝉联世界超级计算机三连冠

2019年2月，布里斯托大学数学教授安德鲁·布克(Andrew Booker)创建了一个算法，来寻找x^3 + y^3 + z^3 = k的解，该算法运行时涉及到10^16次数值，在算法运行几周后获得了33的答案：(8,866,128,975,287,528)^3 + (–8,778,405,442,862,239)^3 + (–2,736,111,468,807,040)^3 = 33。

2019年9月，来自麻省理工学院研究人员Andrew Sutherland和英国布里斯托尔大学的Andrew Booker合作进行了一项超长时间计算，他们使用了超100万小时的慈善引擎计算后，终于破解了42，（-80538738812075974）^3+80435758145817515^3+12602123297335631^3=42。

80538738812075974 + 80435758145817515 + 12602123297335631 =

173,576,620,255,229,120

2进制 8进制 10进制 16进制

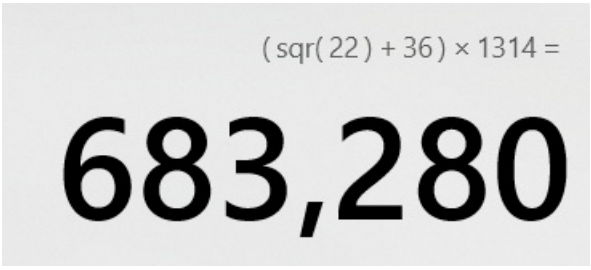
17357662

转 换

8进制结果:	102155536
2进制结果:	1000010001101101101011110
16进制结果:	108db5e

算积分

$$\int_0^{22} 2x\,dx + 36$$



8进制结果:	2466420
2进制结果:	10100110110100010000
16进制结果:	a6d10

情书

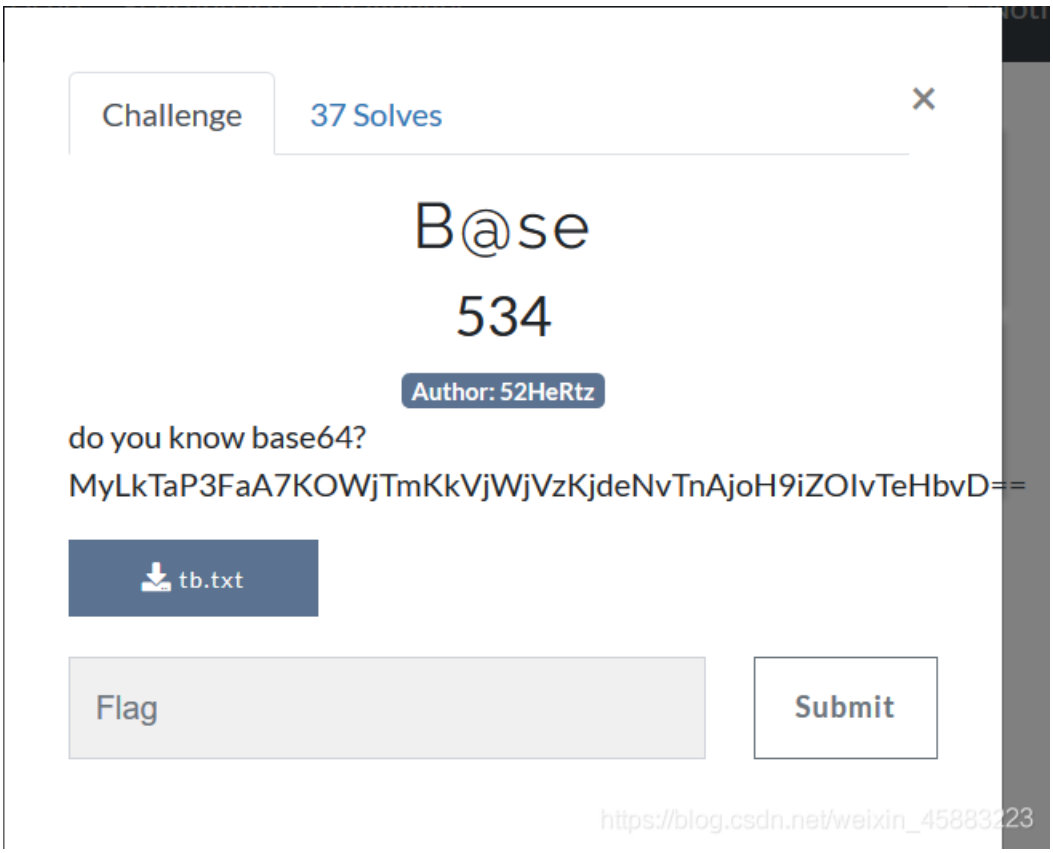
RSA算法，四位一组共八组，公钥2537和13，私钥2537和937，解出来iloveyou（也可根据“情书”盲猜诈胡）

Premise: Enumerate the alphabet by 0、1、2、..... 、25
Using the RSA system
Encryption:0156 0821 1616 0041 0140 2130 1616 0793
Public Key:2537 and 13
Private Key:2537 and 937

flag: wctf2020{Decryption}

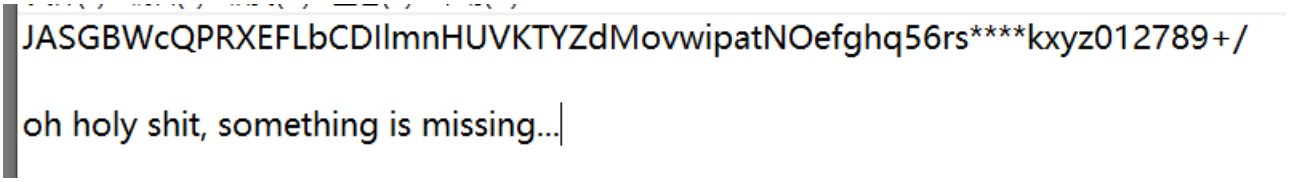
B@se

base64的变种



可知变种表里缺了四位，分别是

j，u，3，4，排列组合一波



写个python脚本，最后试出来34uj的顺序是对的

```
import base64
s1="JASGBWcQPRXEFLbCDIlmnHUVKTYZdMovwipatNOefghq56rs34ujkxyz012789+/"
s2="ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"
base=b'MyLkTaP3FaA7KOWjTmKkVjWjVzKjdeNvTnAjoH9iZOIvTeHbvD=='
flag=''
for i in base:
    if chr(i) != '=':
        index = s1.find(chr(i))
        flag += s2[index]
    else:
        flag +='='
print(flag)
print(base64.b64decode(flag))
```

运行结果

```
d2N0ZjIwMjB7YmFzZTY0XzFzX3YzcnlfZUBzeV9hbmRfZnVOfQ==
b'wctf2020{base64_1s_v3ry_e@sy_and_fuN}'
```

babyrsa

又是RSA

```
c = 28767758880940662779934612526152562406674613203406706867456395986985664083182
n = 73069886771625642807435783661014062604264768481735145873508846925735521695159
e = 65537
```

GitHub上下载RsaCtfTool，根据n和e算出来PEM格式的公钥

```
(RsaCtfTool) root@kali:~/Downloads/RsaCtfTool# ./RsaCtfTool.py --createpub -n 73069886771625642807435783661014062604264768481735145873508846925735521695159 -e 65537
-----BEGIN PUBLIC KEY-----
MDwwDQYJKoZIhvcNAQEBBQADKwAwKAIhAKGMFx5u+BM8Lz8Qgt2I5y4Vb0g6FTmq
5n/egdUD+DW3AgMBAAE=
-----END PUBLIC KEY-----
```

再根据公钥算出来PEM格式的私钥

```
(RsaCtfTool) root@kali:~/Downloads/RsaCtfTool# ./RsaCtfTool.py --publickey ./key.pub --private
-----BEGIN RSA PRIVATE KEY-----
MIGrAgEAAiEAoYwXHm74EzwvPxCC3YjnLhVvSDoVOarmf96B1QP4NbcCAwEAAQIg
RDc9w/Ij+ytc4Ap+2EFpLMvEBq3xSjtOkFJPr5QOHtECEQCOXkWHSf6fEnj5xxea
+xaVAhEBInyb41KVTsIHXcz7pniEGwIRAIK3QeV25gcOAe9sglr1AYUCEQDV7003
tXh02n0mXfLv9U4FAhA55E00OsLFka/zTv5I+HQg
-----END RSA PRIVATE KEY-----
```

用在线ctf工具根据私钥解析出d

结果

公钥(n)

```
730698867716256428074357836610140626042647684817351458735088469257355216951 59
```

公钥(e)

```
65537
```

私钥(d)

9486056472273

写个

python脚本

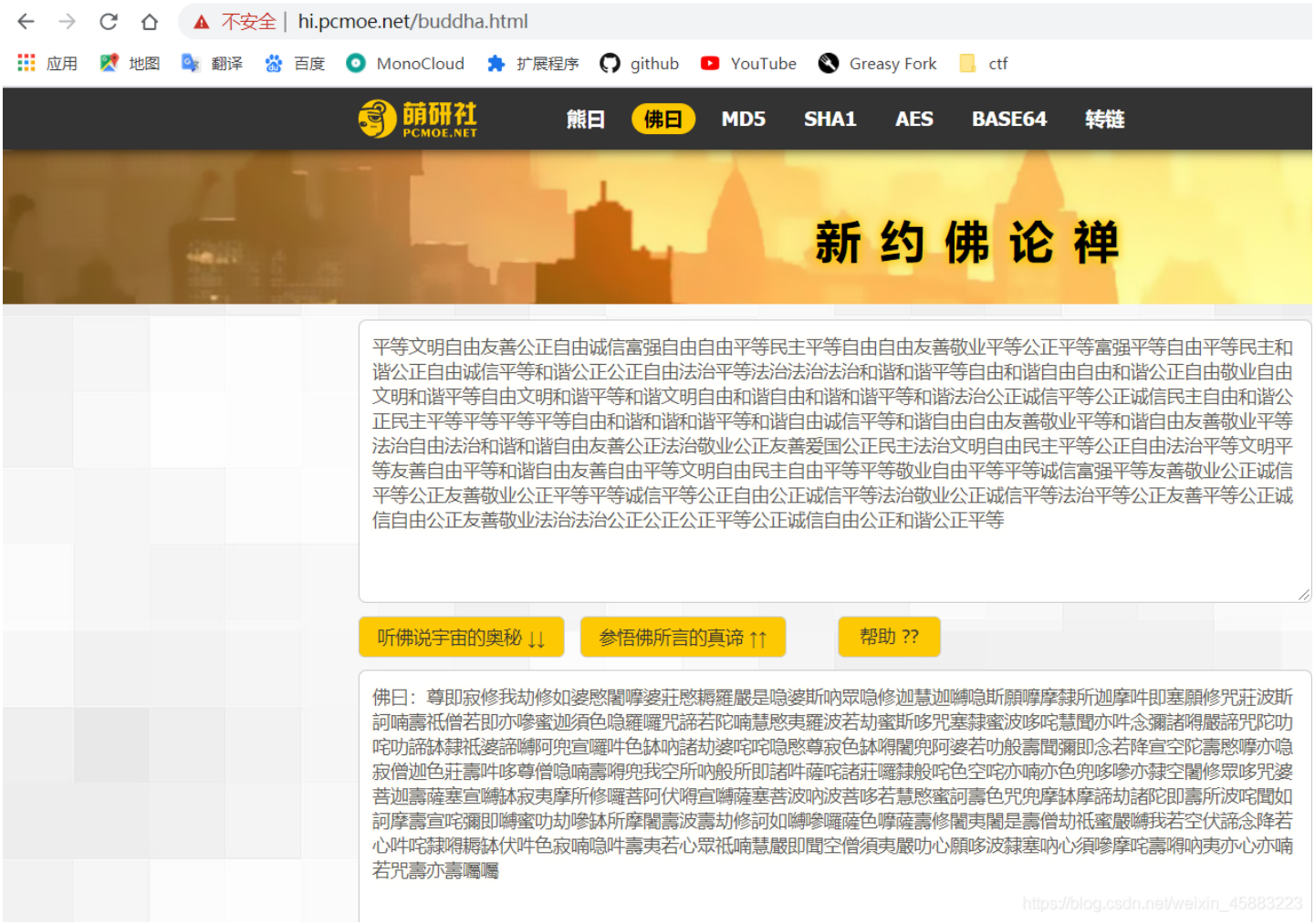
```
from binascii import a2b_hex
c=28767758880940662779934612526152562406674613203406706867456395986985664083182
d=30854876581442056228588093398155288897790570329196285069001545119486056472273
n=73069886771625642807435783661014062604264768481735145873508846925735521695159
flag=a2b_hex(hex(pow(c,d,n))[2:])
print(flag)
```

运行结果

```
b'wctf2020{just_@_piece_0f_cak3}'
```

佛说：只能四天

根据提示“圣经分为《旧约全书》和《新约全书》”可知是新与佛论禅，Google一下在线工具，复制粘贴并在开头加上 佛曰： 参悟佛所言的真谛，解码出了核心价值观编码



在线ctf工具解码，根据结尾的doyouknowfense可知是栅栏密码

编 码　　解 码

平等文明自由友善公正自由诚信富强自由自由平等民主平等自由自由友善敬业平等公正平等富强平等自由平等民主和谐公正自由诚信平等和谐公正公正自由法治平等法治法治法治和谐和谐平等自由和谐自由自由和谐公正自由敬业自由文明和谐平等自由文明和谐平等和谐文明自由和谐自由和谐和谐平等和谐法治公正诚信平等公正诚信民主自由和谐公正民主平等平等平等平等自由和谐和谐和谐平等和谐自由诚信平等和谐自由自由友善敬业平等和谐自由友善敬业平等法治自由法治和谐和谐自由友善公正法治敬业公正友善爱国公正民主法治文明自由民主平等公正自由法治平等文明平等友善自由平等和谐自由友善自由平等文明自由民主自由平等平等敬业自由平等平等诚信富强平等友善敬业公正诚信平等公正友善敬业公正平等平等诚信平等公正自由公正诚信平等法治敬业公正诚信平等法治平等公正友善平等公正诚信自由公正友善敬业法治法治公正公正公正平等公正诚信自由公正和谐公正平等

去掉结尾的doyouknowfense后在线解码，每组字数试到4出来了doyouknowCaesar可知是凯撒密码

RLJDQTOVPTQ6O6duws5CD6IB5B52CC57okCaUUC3SO4OSOWG3LynarAVGRZSJRAEYEZ_ooe_

每组字数 4　加密　解密

R5UALCUVJDCGD63RQISZTBOSO54JVBORP5SAT2OEQCWY6CGEO53Z67L_doyouknowCaesar_

栅栏密码是一种简单的移动字符位置的加密方法，规则简单，容易破解。栅栏密码的加密方式：把文本按照一定的字数分成多个组，取每组第一个字连起来得到密文1，再取每组第二个字连起来得到密文2……最后把密文1、密文2……连成整段密文。例如：

在线凯撒解码（凯撒因为位移是3所以被叫作凯撒）

R5UALCUVJDCGD63RQISZTBOSO54JVBORP5SAT2OEQCWY6CGEO53Z67L

位移 3　加密　解密

O5RXIZRSGAZDA63ONFPWQYLPL54GSYLOM5PXQ2LBNZTV6ZDBL53W67I

凯撒密码最早由古罗马军事统帅盖乌斯·尤利乌斯·凯撒在军队中用来传递加密信息，故称凯撒密码。这是一种位移加密方式，只对26个字母进行位移替换加密，规则简单，容易破解。下面是位移1次的对比：

把得到的编码base解码，base64失败，base32成功了

Base32编码解码

O5RXIZRSGAZDA63ONFPWQYLPL54GSYLOM5PXQ2LBNZTV6ZDBL53W67I

编码 解码 清空

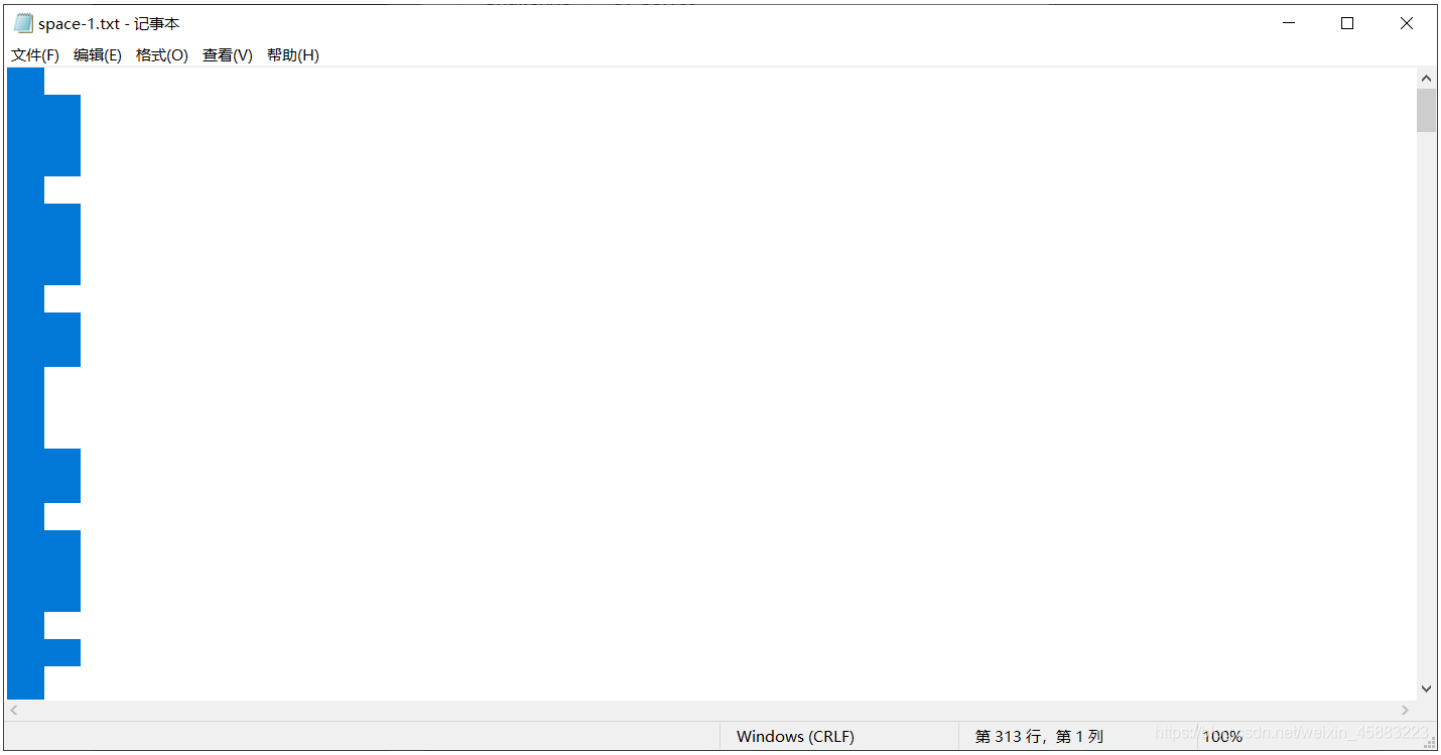
wctf2020{ni_hao_xiang_xiang_da_wo}

复制

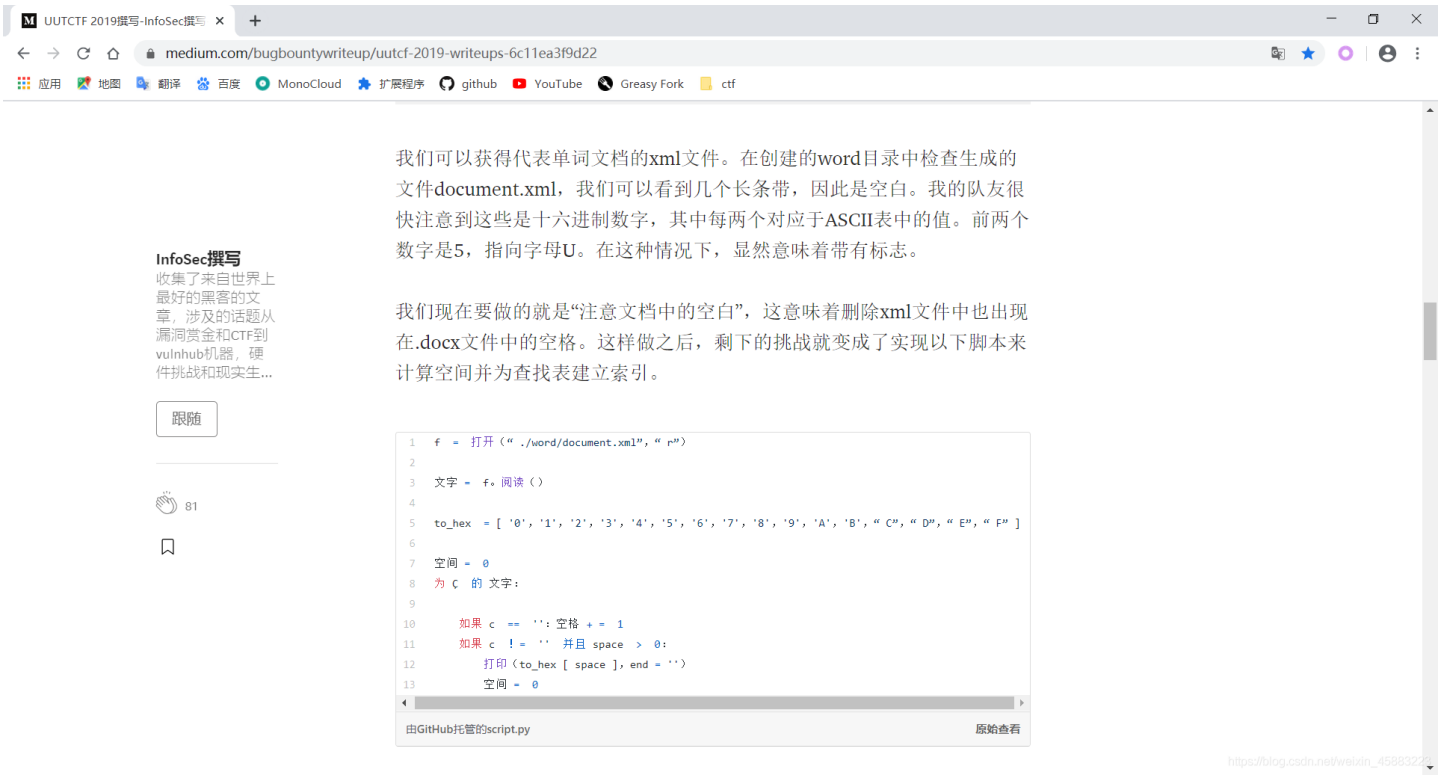
Misc

Space Club

打开文件是个空的txt，Ctrl+A发现有好多有规律的空格



Google一下找到了一篇UUTCTF 2019的writeup（英文的，翻译一下）



把代码copy一下，路径改成自己的，并要在前面加上一个r（报错了，百度才知道的），否则不行

```
f = open(r"C:\Users\1\Desktop\space-1.txt", "r")

text = f.read()

to_hex = [ '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F' ]

space = 0
for c in text:

    if c == ' ': space += 1
    if c != ' ' and space > 0:
        print(to_hex[space], end='')
        space = 0
```

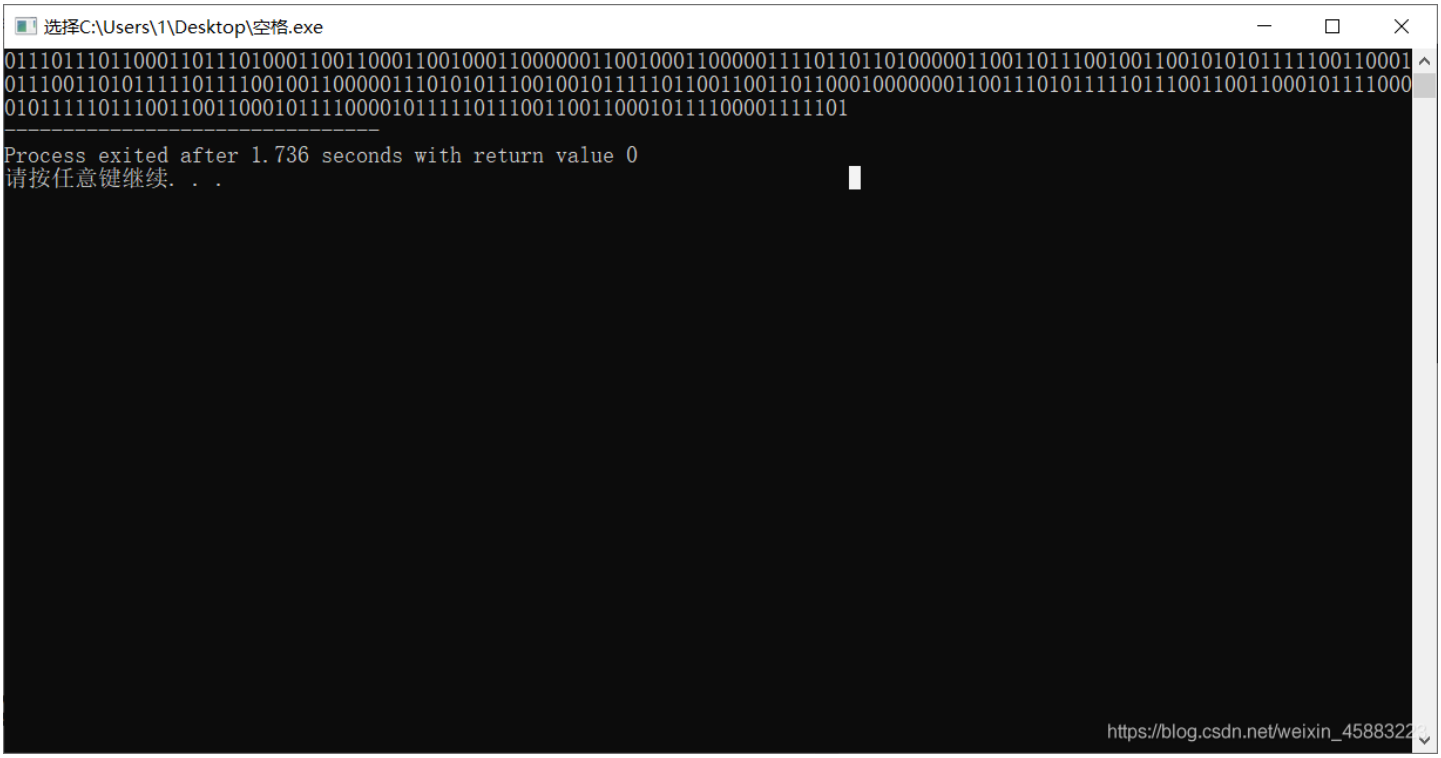
运行结果全是6和C

```
6CCC6CCC6CC666CC6CCC6C666CC66CC666CC66C666CC666666CC66C666CC66666CCCC6CC6CC6C66666CC66CC6CCC66C66CC66C6C6C6CCCCC
66CC666C6CCC66CC6C6CCCCC6CCCC66C66CC66666CCC6C6C6CCC66C66C6CCCCC6CC66CC66CC6CC666C6666666CC66CCC6C6CCCCC6CCC66CC
66CC666C6CCCC6666C6CCCCC6CCC66CC66CC666C6CCCC6666C6CCCCC6CCC66CC66CC666C6CCCC6666CCCCC6C
```

盲猜一波是二进制，自己写个C++把6换成0，C换成1

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
 string s="6CCC6CCC6CC666CC6CCC6C666CC66CC666CC66C666CC666666CC66C666CC66666CCCC6CC6CC6C66666CC66CC6CCC66C66CC66
C6C6C6CCCCC66CC666C6CCC66CC6C6CCCCC6CCCC66C66CC66666CCC6C6C6CCC66C66C6CCCCC6CC66CC66CC6CC666C6666666CC66CCC6C6CC
CCC6CCC66CC66CC666C6CCCC6666C6CCCCC6CCC66CC66CC666C6CCCC6666C6CCCCC6CCC66CC66CC666C6CCCC6666CCCCC6C";
 for(int i=0;i<s.size();i++)
 {
  if(s[i]=='6')
   s[i]='0';
  else
   s[i]='1';
 }
 cout<<s;
 return 0;
}
```

运行结果



在线二进制转字符串

输入二进制文本:

01110111011000110111010001100110001100100011000000110010001100000111101101101000001100110111001001100101010111110011000101110011010111110111100100110000011101010111001001011111011001100110110001000000011001110101111101110011000110001011110000101111000111001100110001011110000

转换后的文本:

wctf2020{h3re_1s_y0ur_fl@g_s1x_s1x_s1x}

Welcome

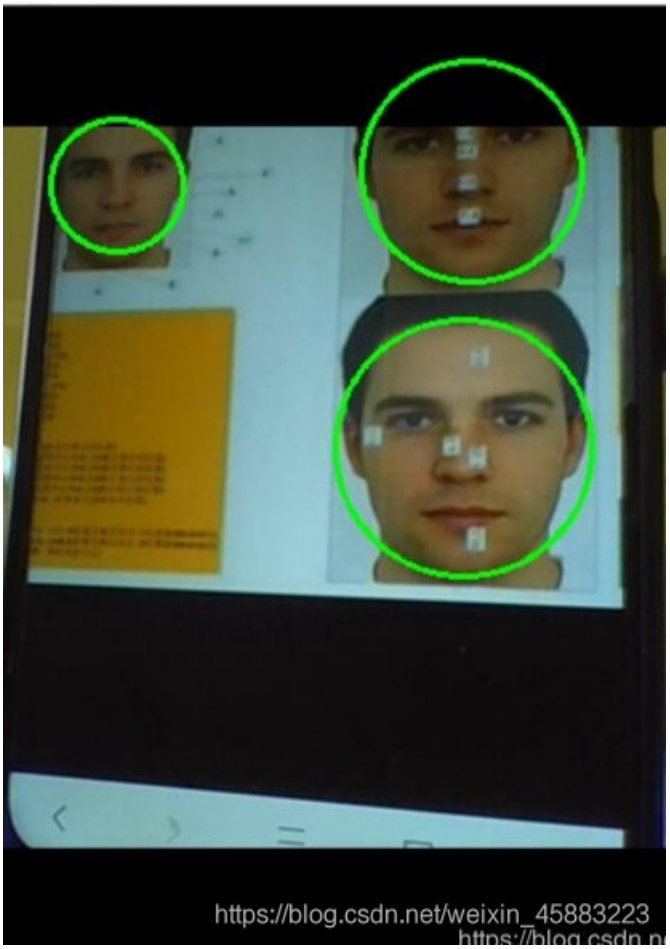
文件夹里的三个文件要一起，不能只把exe拿出来，丢到ida里也看不出来啥东西

haarcascade_frontalface_default.xml
welcome.exe
题目说明.txt

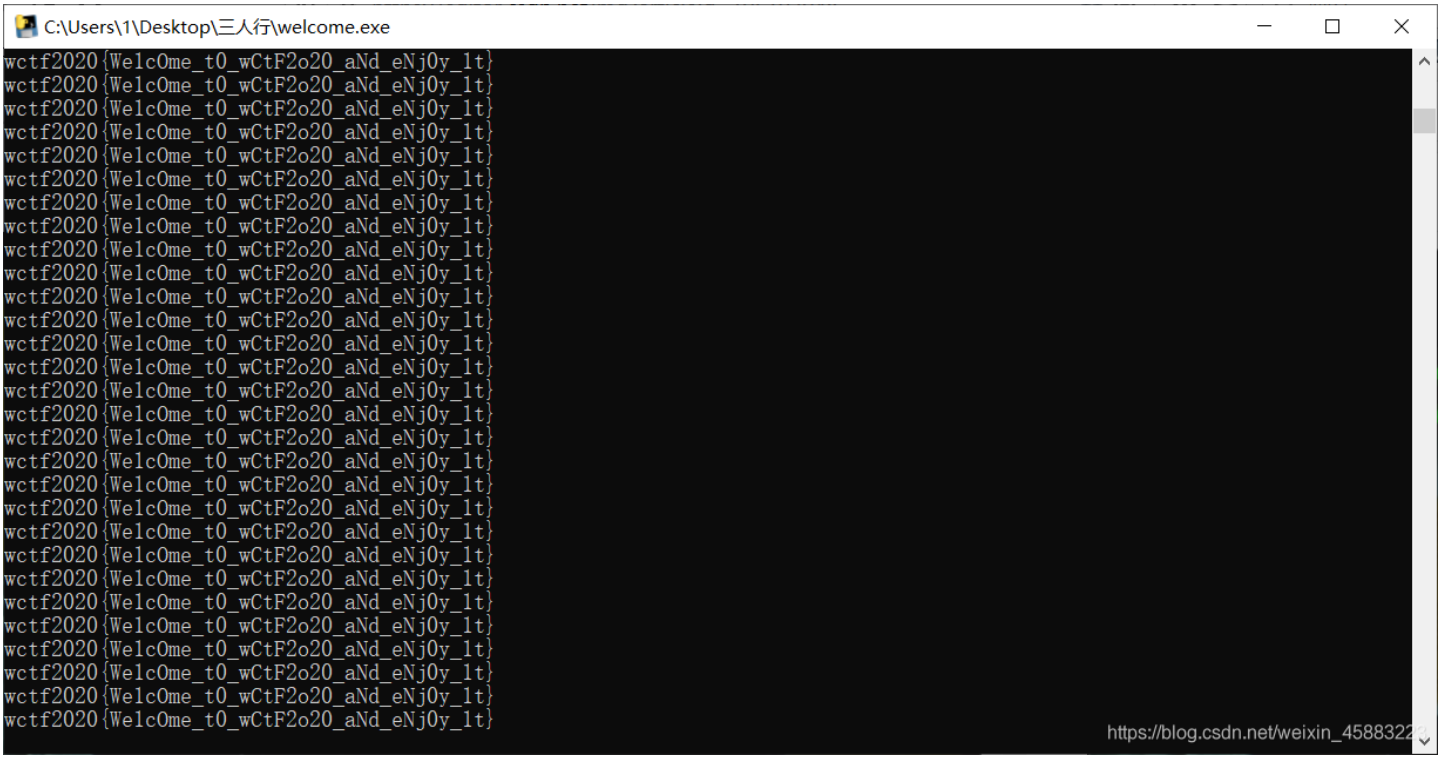
题目说明

可能需要Windows7以更高版本的Windows环境
需要用到你的摄像头
需要用到你帅气美丽的脸蛋

Google一下是第十二届全国大学生信息安全竞赛线上初赛的签到题，摄像头里要识别到三个人脸



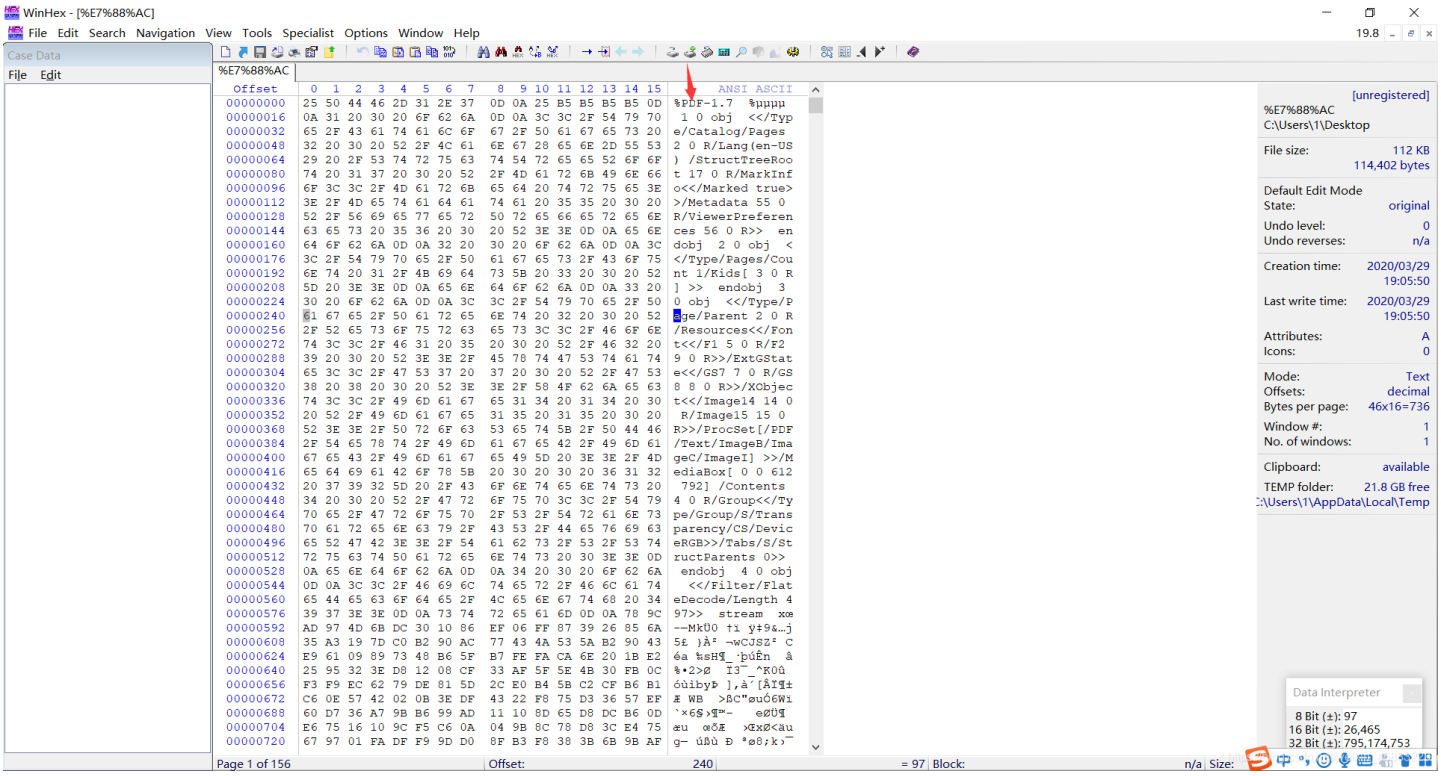
就出来flag了



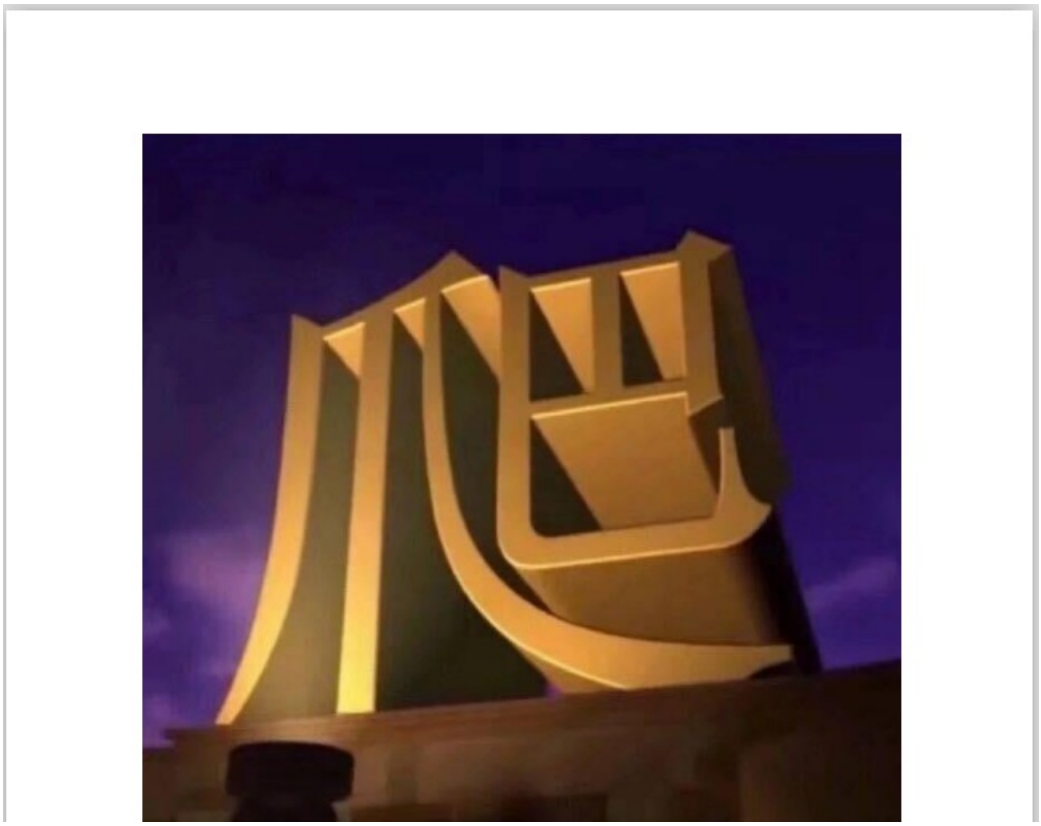
爬

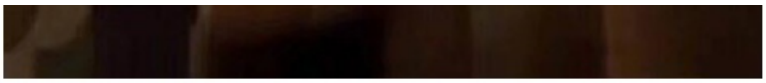
下载下来是无后缀文件

丢到winhex里看一下在开头发现了PDF



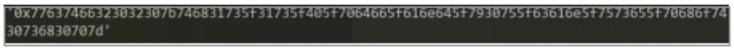
把后缀加上.pdf就可以打开了





Flag 被图片覆盖住了

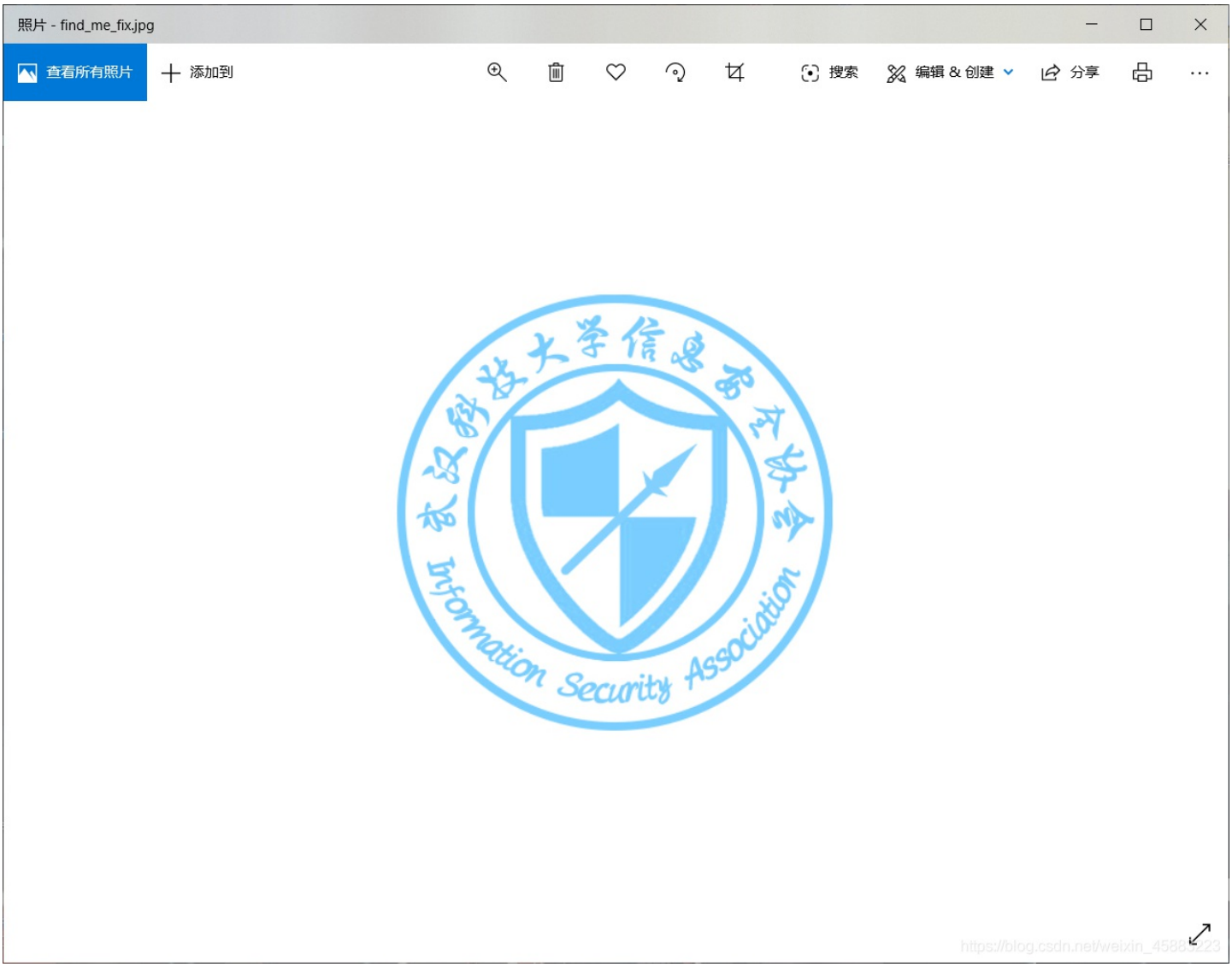
flag被图片覆盖住了，把图片拖开就看到了16进制字符串（不要转成word再拖），在线16进制转字符串即可



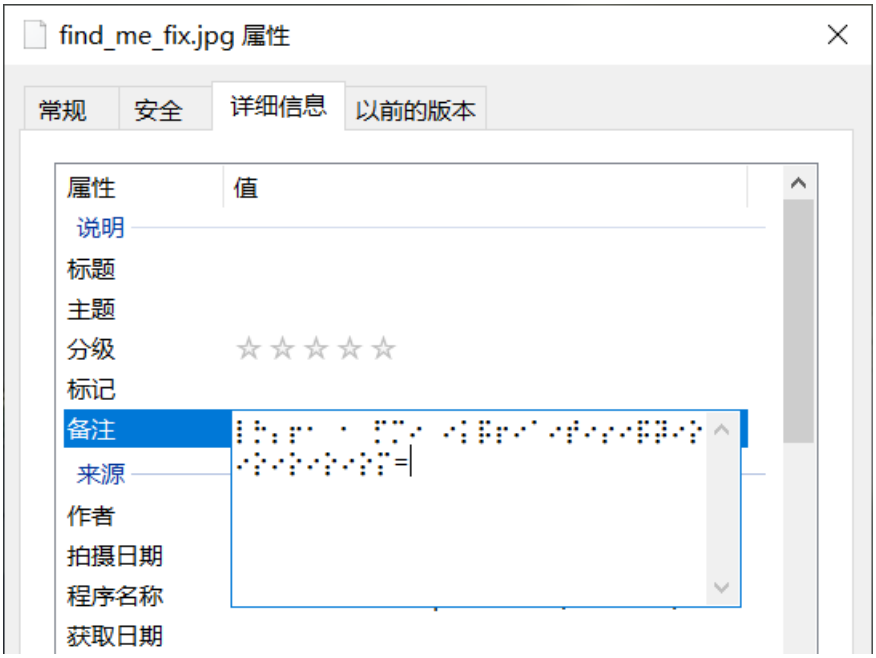
Flag 被图片覆盖住了

Find me

以为是图片隐写，winhex，binwalk都没啥线索



属性的详细信息里发现了八点式盲文，Google在线工具（千千秀字）解码即可，不要丢掉结尾的=，我刚开始以为没用，解不出来，白费了我几个小时搜别的工具

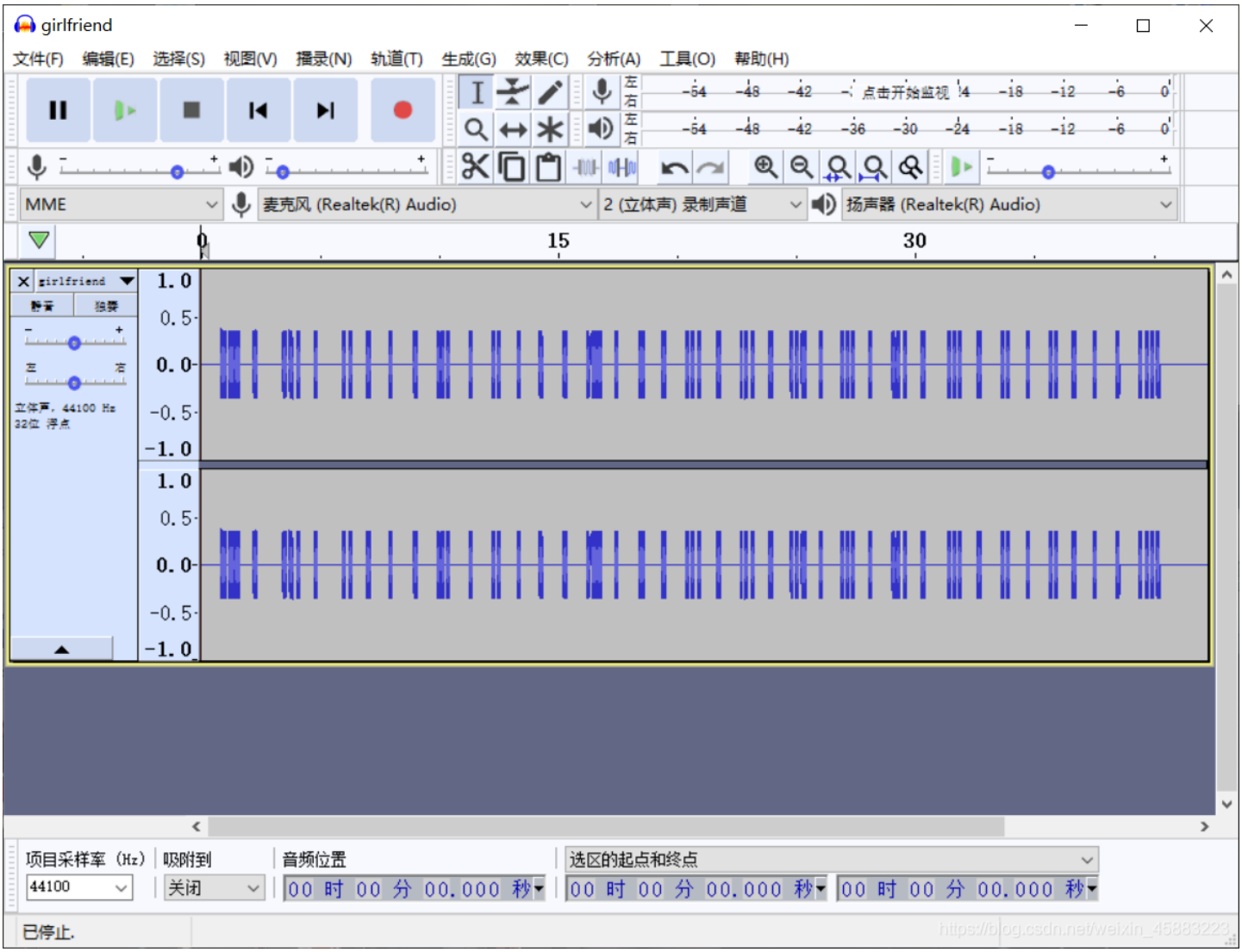


文本加密为盲文

girlfriend

音频文件隐写

丢到Audacity里，刚开始以为是摩斯密码（长短），解出来不对，听一下像是电话按键音



Google在线Detect DTMF Tones上传wav解出来下面的这段数字，像是手机拼音九键键盘，解出来发现只有前半段，根据名字girfriend盲猜是youaremygirlfriend，再跟音频对比一下发现位数不对，最后有一个连续按四下的音，再看九键键盘推理出来最后一位加上个s

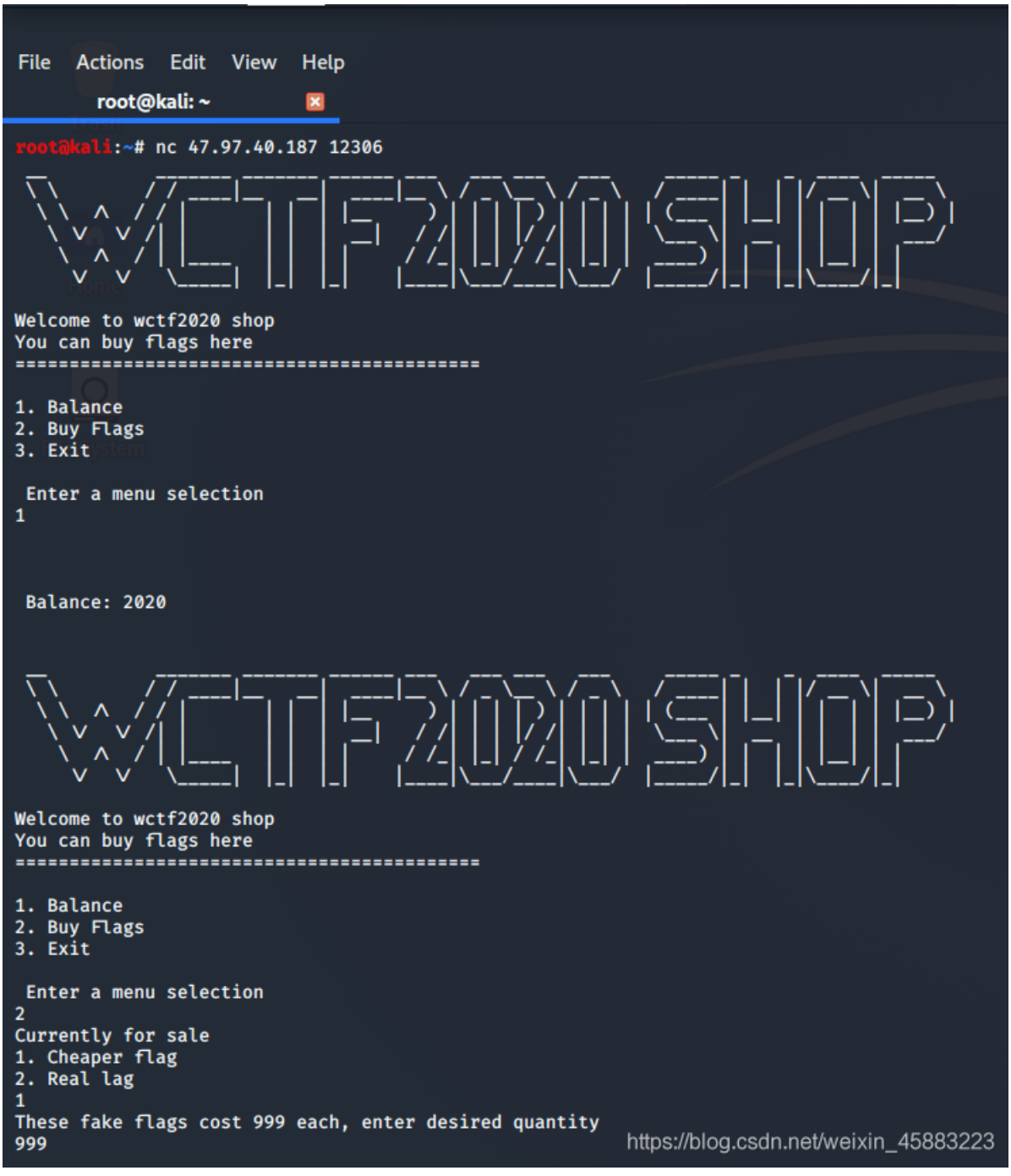
```
999*666*88*2*777*33*6*999*4*444*777*555*333*777
youaremygirlfr
```

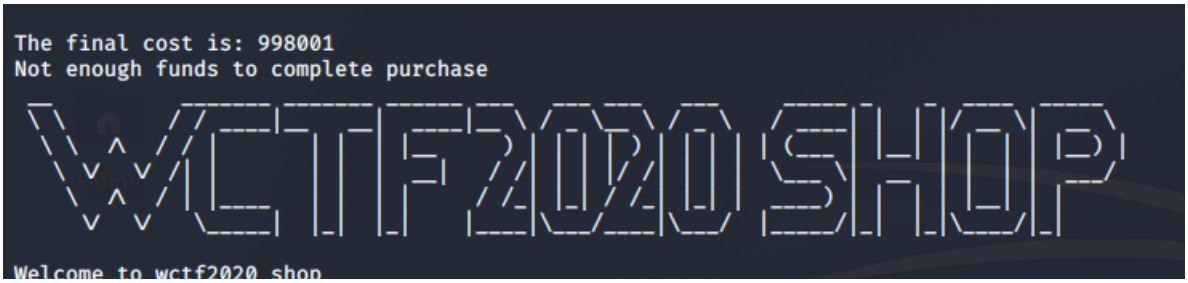
Shop

Shop

896

在kali linux里输入一下命令，再输入1,2多试试



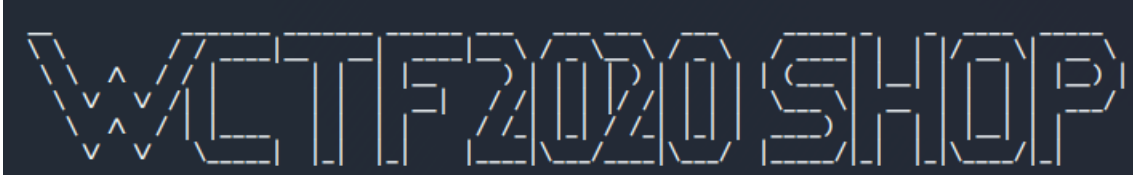


```
You can buy flags here
========================================

1. Balance
2. Buy Flags
3. Exit

 Enter a menu selection
2
Currently for sale
1. Cheaper flag
2. Real lag
2
Real flags cost 100000 dollars, and we only have 1 in stock
Enter 1 to buy one1

Not enough funds for transaction
```



```
Welcome to wctf2020 shop
You can buy flags here
========================================

1. Balance
2. Buy Flags
3. Exit
```


Google一下找到了一篇英文的PicoCTF 2019 Writeup，里面有一道flag_shop，可知如果输入一个大数，就会溢出变成一个大负数

flag_shop

Problem

There's a flag shop selling stuff, can you buy a flag? Source. Connect with `nc 2019shell1.picoctf.com 3967`.

source

Solution

By reading the source code, we see that the `total_cost` is stored as a 4 byte signed integer:

```
if(number_flags > 0){
    int total_cost = 0;
    total_cost = 900*number_flags;
    printf("\nThe final cost is: %d\n", total_cost);
    if(total_cost <= account_balance){
        account_balance = account_balance - total_cost;
        printf("\nYour current balance after transaction: %d\n\n", acco
    }
    else{
        printf("Not enough funds to complete purchase\n");
    }
}
```

If we enter a large number for `number_flags`, `900*number_flags` would overflow and turn into a large negative number:

```
$ pytnon
>>> ((1<<31)//900)*1.5
3579138.0
```

```
$ nc 2019shell1.picoctf.com 3967
Welcome to the flag exchange
We sell flags

1. Check Account Balance
```



```
2. Buy Flags

3. Exit

 Enter a menu selection
2
Currently for sale
1. Defintely not the flag Flag
2. 1337 Flag
1
These knockoff Flags cost 900 each, enter desired quantity
3579138

The final cost is: -1073743096

Your current balance after transaction: 1073744196

Welcome to the flag exchange
We sell flags

1. Check Account Balance

2. Buy Flags

3. Exit

 Enter a menu selection
2
Currently for sale
1. Defintely not the flag Flag
2. 1337 Flag
2
1337 flags cost 100000 dollars, and we only have 1 in stock
Enter 1 to buy one1
YOUR FLAG IS: picoCTF{m0n3y_bag5_cd0ead78}
```

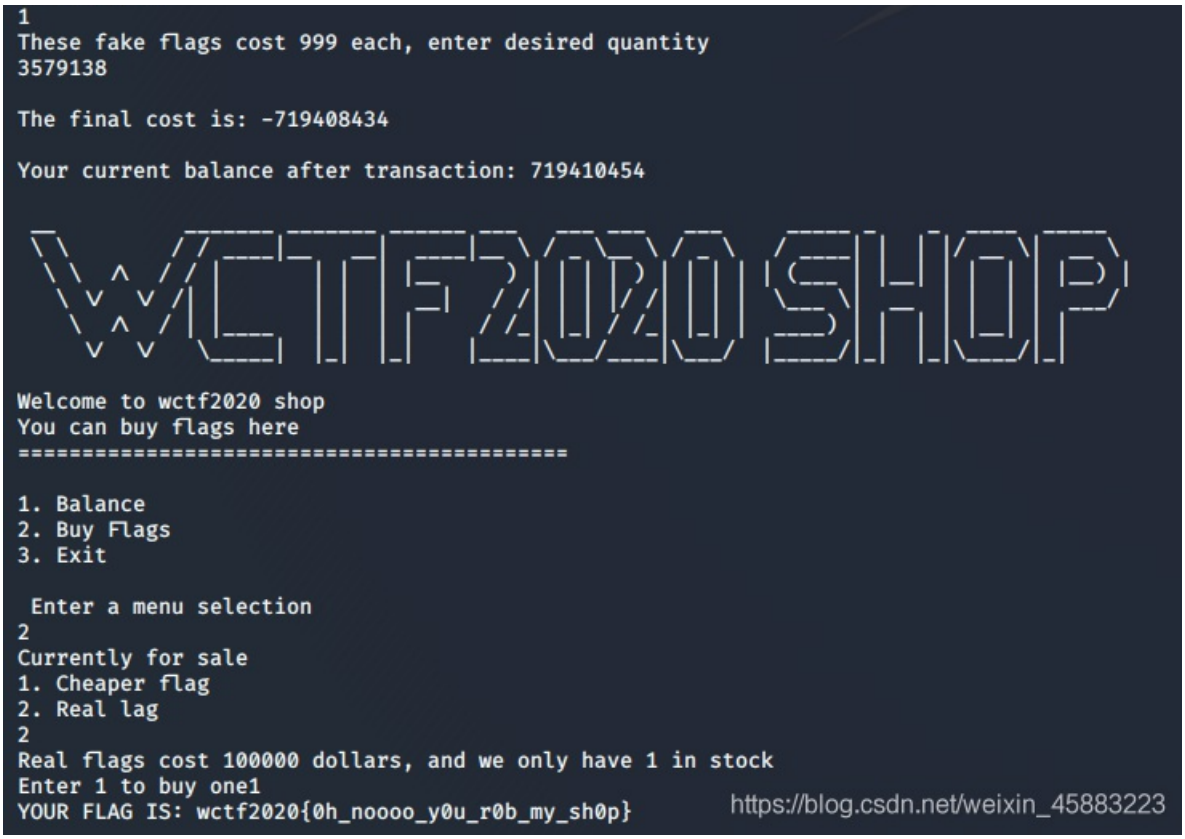

输入3579138，就有钱了，有钱就能买flag了，有钱真好

```
     V  V   \_____|  |_|  |_|    |____|\__/____|\__/  |_____/|_|  |_|\____/|_|

Welcome to wctf2020 shop
You can buy flags here
========================================

1. Balance
2. Buy Flags
3. Exit

 Enter a menu selection
2
Currently for sale
1. Cheaper flag
2. Real lag
```



Reverse

Cr0ssFun

拖进ida里，把里面每一个函数都点开，再copy到自己的C++里，把&&，==等格式改一下输出a1就行了

IDA View-A | Pseudocode-A | Hex View-1 | St

```c
int __cdecl main(int argc, const char **argv, const char **envp)
{
  char v4; // [rsp+0h] [rbp-30h]

  puts(" _    _ _  _ _____ _____   _____            ");
  puts("| |  | | | | /  ___|_   _| /  ___|          ");
  puts("| |  | | | | \\ `--.  | |   \\ `--.  ___  ___ ");
  puts("| |/\\| | | | |`--. \\ | |    `--. \\/ _ \\/ __|");
  puts("\\  /\\  / |_| /\\__/ / | |   /\\__/ /  __/ (__ ");
  puts(" \\/  \\/ \\___/\\____/  \\_/   \\____/ \\___|\\___|");
  while ( 1 )
  {
    puts("Input the flag");
    __isoc99_scanf((__int64)"%s", (__int64)&v4);
    if ( (unsigned int)check((__int64)&v4) == 1 )
      break;
    puts("Oops, your flag seems fake.");
    puts("==============================");
    rewind(_bss_start);
  }
  puts("Your flag is correct, go and submit it!");
  return 0;
}
```



```c
_BOOL8 __fastcall check(__int64 a1)
{
  return iven_is_handsome((_BYTE *)a1);
}
```

```c
_BOOL8 __fastcall iven_is_handsome(_BYTE *a1)
{
  return a1[10] == 112
      && a1[13] == 64
      && a1[3] == 102
      && a1[26] == 114
      && a1[20] == 101
      && (unsigned int)iven_is_c0ol(a1);
}
```

```cpp
#include<bits/stdc++.h>
using namespace std;
int main()
{
 char a1[100];
 a1[0] = 119 ;
 a1[6] = 50;
 a1[22] = 115;
 a1[31] = 110;
 a1[12] = 95;
 a1[7] = 48;
    a1[16] = 95;
    a1[11] = 112;
    a1[23] = 101;
    a1[30] = 117;
    a1[10] = 112;
    a1[13] = 64;
    a1[3] = 102;
    a1[26] = 114;
    a1[20] = 101;
    a1[1] = 99 ; a1[25] = 64;  a1[27] = 101;
    a1[4] = 50 ; a1[17] = 114 ; a1[29] = 102 ; a1[17] = 114 ; a1[24] = 95;
    a1[2] = 116;
    a1[9] = 99;
    a1[32] = 125;
    a1[19] = 118;
    a1[5] = 48;
    a1[14] = 110;
    a1[15] = 100;
    a1[8] = 123;
    a1[18] = 51;
    a1[28] = 95;
    a1[21] = 114;
    cout<<a1;
 return 0;
}
```

运行结果

```
C:\Users\1\Desktop\crossfun.exe
wctf2020{cpp_@nd_r3verse_@re_fun}□□
--------------------------------
Process exited after 1.751 seconds with return value 0
请按任意键继续. . .
```

level1

根据output可知这些数都是输出的结果

output.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
198
232
816
200
1536
300
6144
984
51200
570
92160
1200
565248
756
1474560
800
6291456
1782
65536000
```

Unix (LF) 第 1 行，第 1 列 100%

拖进ida里看一下，把for循环里面的逆过来就行了

IDA View-A | Pseudocode-A | Hex View-1

```
1  int __cdecl main(int argc, const char **argv, const char **envp)
2  {
3    FILE *stream; // ST08_8
4    signed int i; // [rsp+4h] [rbp-2Ch]
5    char ptr[24]; // [rsp+10h] [rbp-20h]
6    unsigned __int64 v7; // [rsp+28h] [rbp-8h]
7
8    v7 = __readfsqword(0x28u);
9    stream = fopen("flag", "r");
10   fread(ptr, 1uLL, 0x14uLL, stream);
11   fclose(stream);
12   for ( i = 1; i <= 19; ++i )
13   {
14     if ( i & 1 )
15       printf("%ld\n", (unsigned int)(ptr[i] << i));
16     else
17       printf("%ld\n", (unsigned int)(i * ptr[i]));
18   }
19   return 0;
20 }
```


把output作为a[20]里的值，但是注意i是从1开始的，所以要给a[0]赋个值，把ld改成c，unsigned int改成char，<<改成>>，*改成/

```cpp
#include<bits/stdc++.h>
using namespace std;
int main()
{
   long a[20]={0,198,232,816,200,1536,300,6144,984,51200,570,92160,1200,565248,756,1474560,800,6291456,1782,6553
6000};
 for (int i = 1; i <= 19; ++i )
  {
    if ( i & 1 )
      printf("%c", (char)(a[i] >> i));
    else
      printf("%c", (char)(a[i]/i));
  }
 return 0;
}
```

运行结果