【picoCTF 2022 一些题目的wp】

原创

aoao今晚吃什么 于 2022-03-27 12:31:41 发布 609 收藏

分类专栏： 比赛wp 文章标签： p2p 网络协议 网络

本文链接：https://blog.csdn.net/aoao331198/article/details/123746483

版权

比赛wp 专栏收录该内容

3 篇文章 0 订阅

订阅专栏

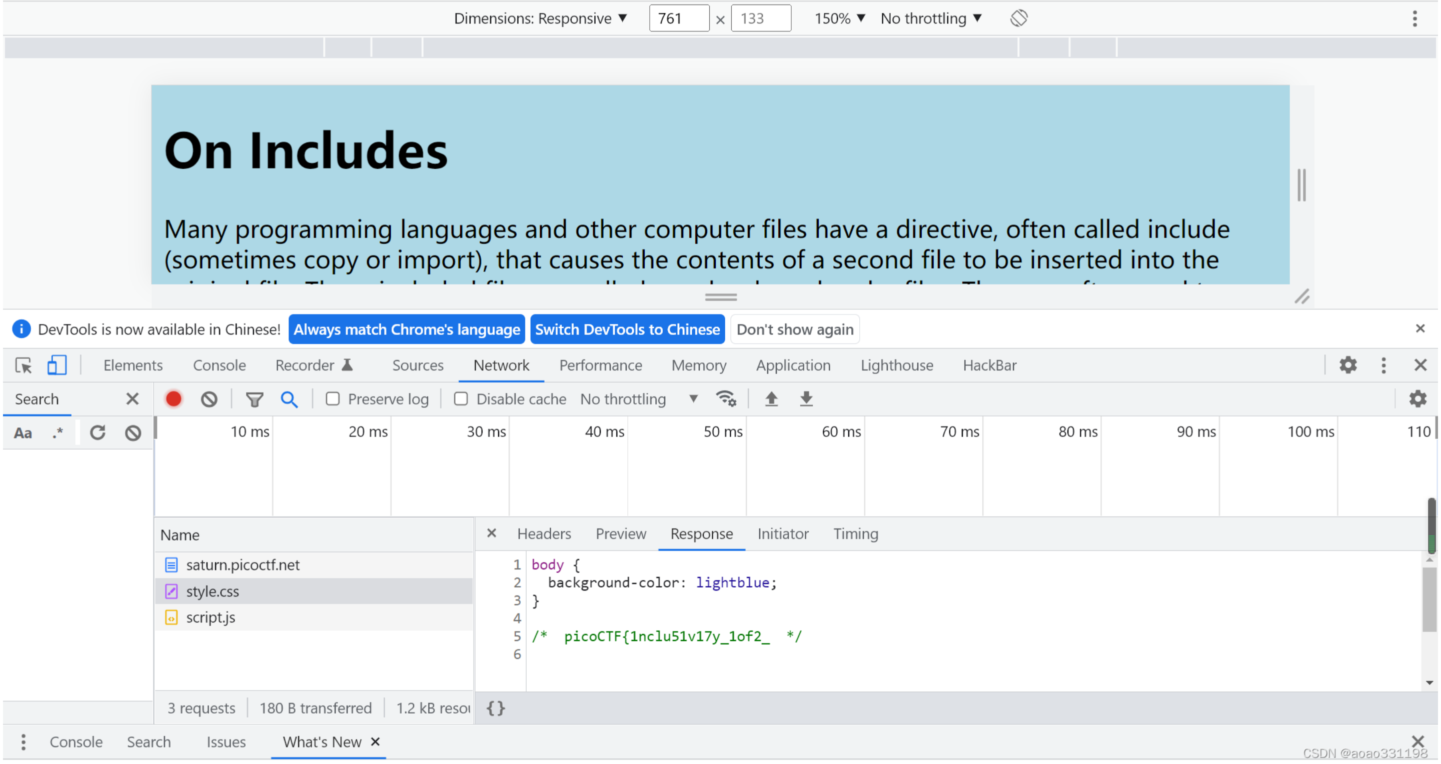
这里记录一下picoCTF 2022 中做出来的题目（WEB和Crypto以外的题目就不写了^_^，因为只会签到题qwq）

文章目录

WEB

1.Includes

打开开发者工具，在style.css中看见flag



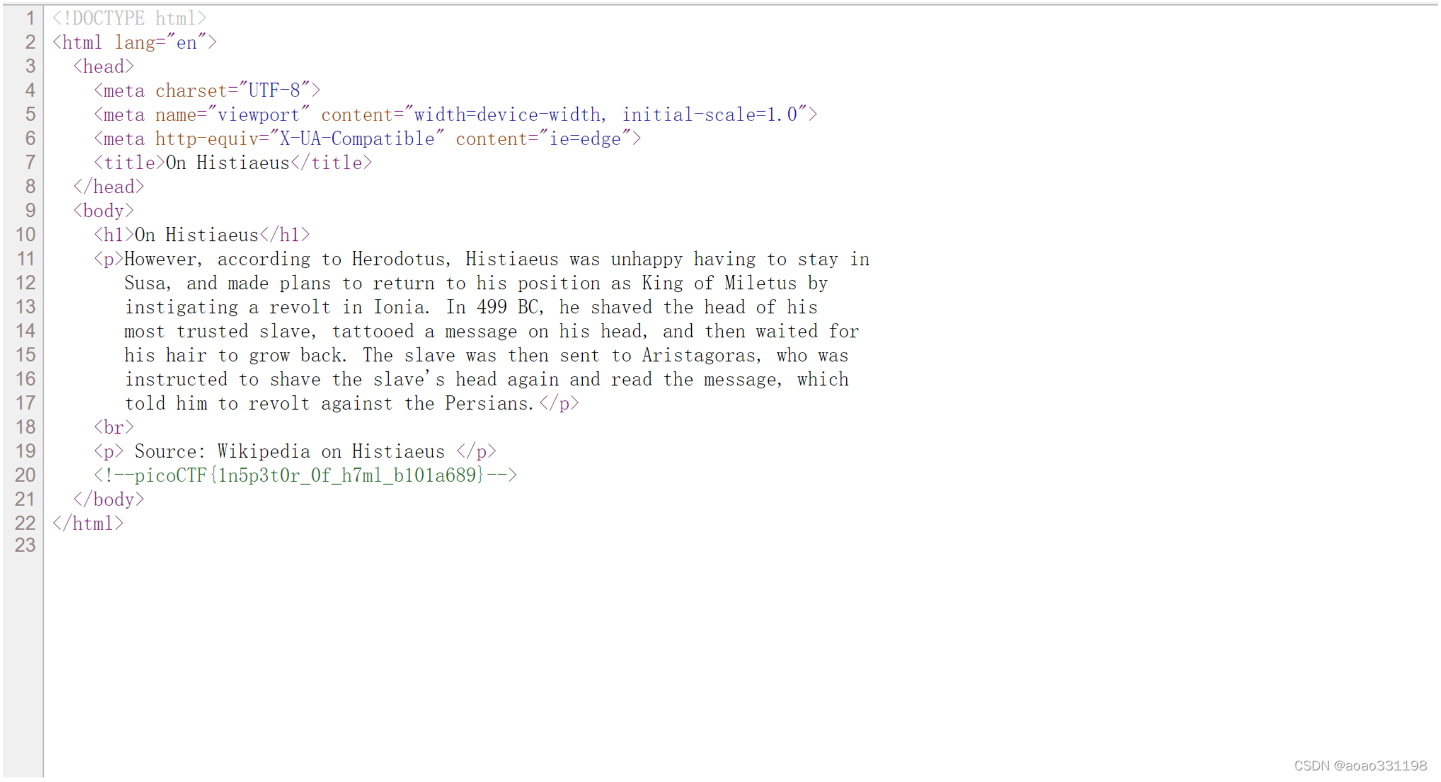
On Includes

Many programming languages and other computer files have a directive, often called include (sometimes copy or import), that causes the contents of a second file to be inserted into the

```
body {
  background-color: lightblue;
}

/*  picoCTF{1nclu51v17y_1of2_  */

```

2.Inspect HTML

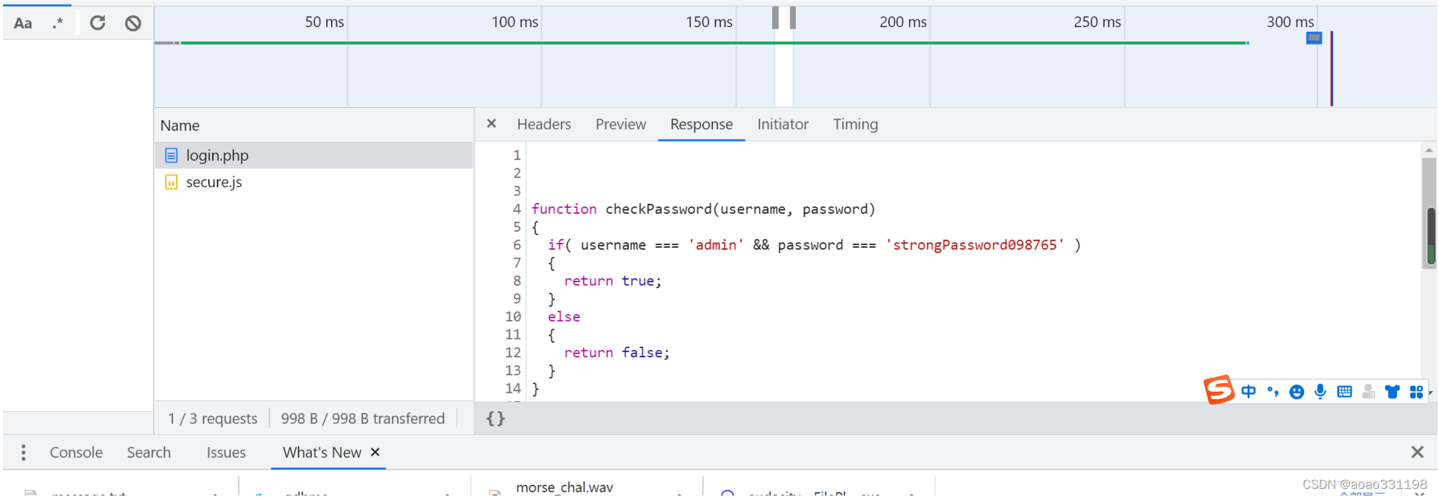
查看源代码



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>On Histiaeus</title>
  </head>
  <body>
    <h1>On Histiaeus</h1>
    <p>However, according to Herodotus, Histiaeus was unhappy having to stay in
       Susa, and made plans to return to his position as King of Miletus by
       instigating a revolt in Ionia. In 499 BC, he shaved the head of his
       most trusted slave, tattooed a message on his head, and then waited for
       his hair to grow back. The slave was then sent to Aristagoras, who was
       instructed to shave the slave's head again and read the message, which
       told him to revolt against the Persians.</p>
    <br>
    <p> Source: Wikipedia on Histiaeus </p>
    <!--picoCTF{1n5p3t0r_0f_h7ml_b101a689}-->
  </body>
</html>

```

3.Local Authority

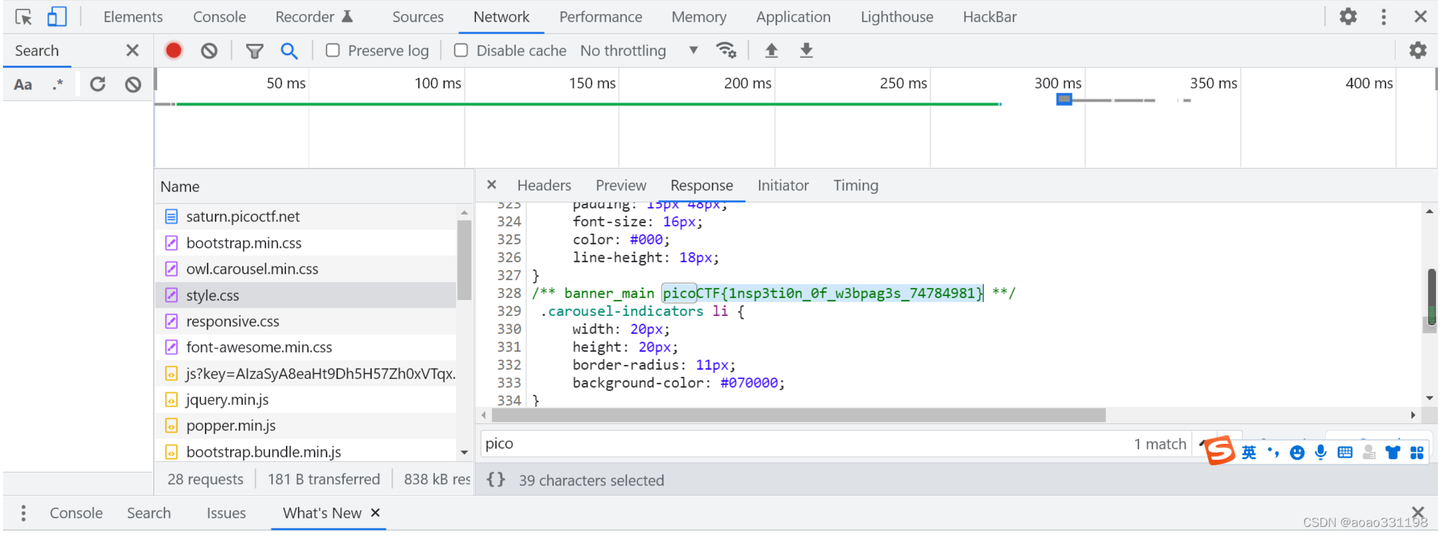
密码找到



```


function checkPassword(username, password)
{
  if( username === 'admin' && password === 'strongPassword098765' )
  {
    return true;
  }
  else
  {
    return false;
  }
}
```

4.Search source

Ctrl+F 在文件中一个一个搜索查找flag



```
    padding: 15px 48px;
    font-size: 16px;
    color: #000;
    line-height: 18px;
}
/** banner_main picoCTF{1nsp3ti0n_0f_w3bpag3s_74784981} **/
 .carousel-indicators li {
    width: 20px;
    height: 20px;
    border-radius: 11px;
    background-color: #070000;
}
```

5.Forbidden Paths

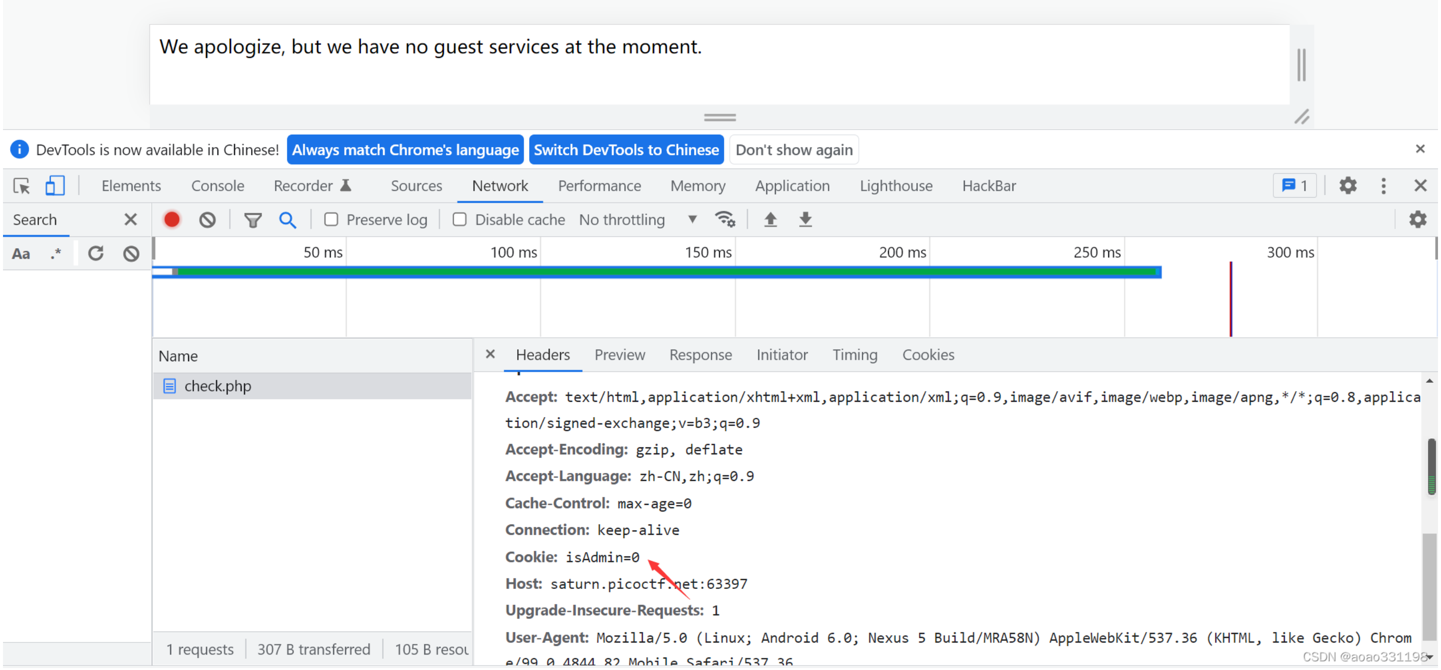
> We know that the website files live in `/usr/share/nginx/html/` and the flag is at `/flag.txt` but the website is filtering absolute file paths. Can you get past the filter to read the flag?

绝对路径被过滤，就用相对路径

```
../../../../flag.txt
```

6.Power Cookie

直接锁定cookie



We apologize, but we have no guest services at the moment.

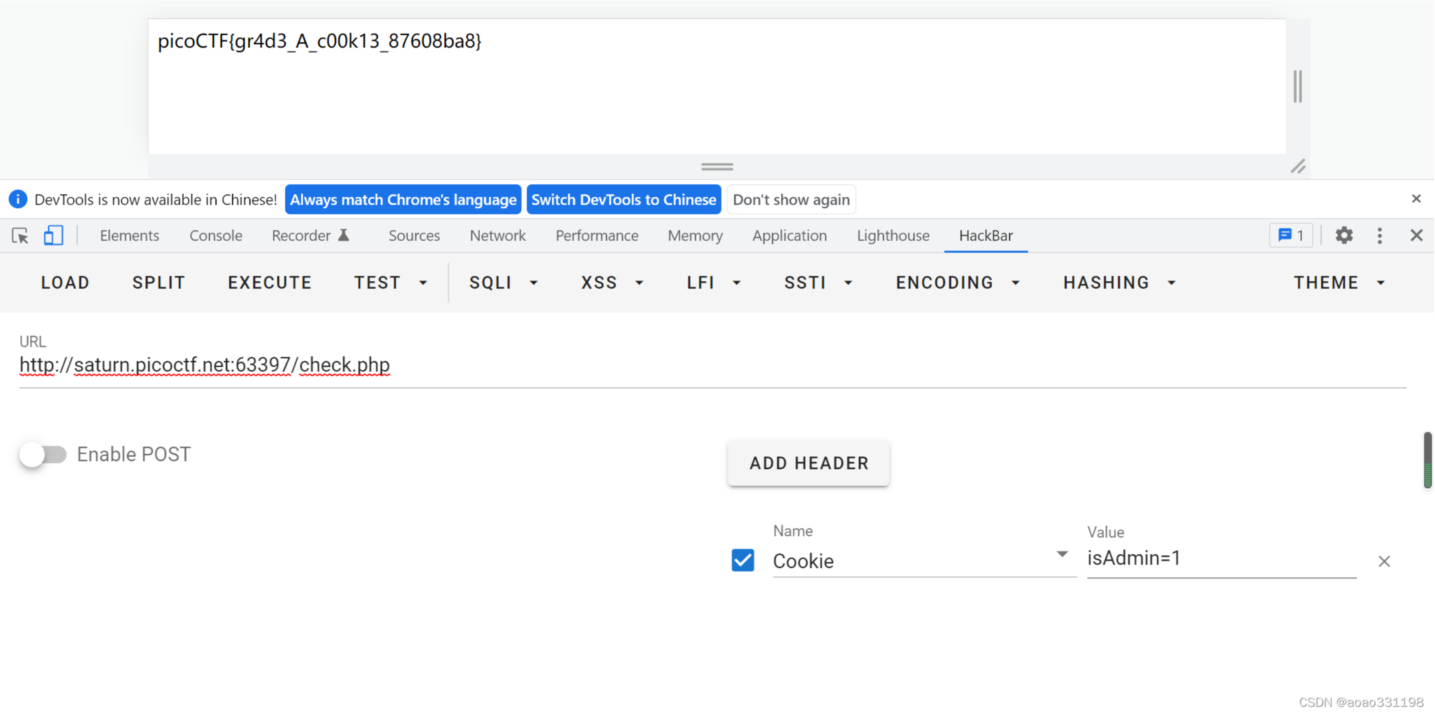
DevTools is now available in Chinese!

check.php

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applica
tion/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cache-Control: max-age=0
Connection: keep-alive
Cookie: isAdmin=0
Host: saturn.picoctf.net:63397
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrom
e/99.0.4844.82 Mobile Safari/537.36
```

1 requests | 307 B transferred | 105 B resou

```
isAdmin=1
```



picoCTF{gr4d3_A_c00k13_87608ba8}

DevTools is now available in Chinese!

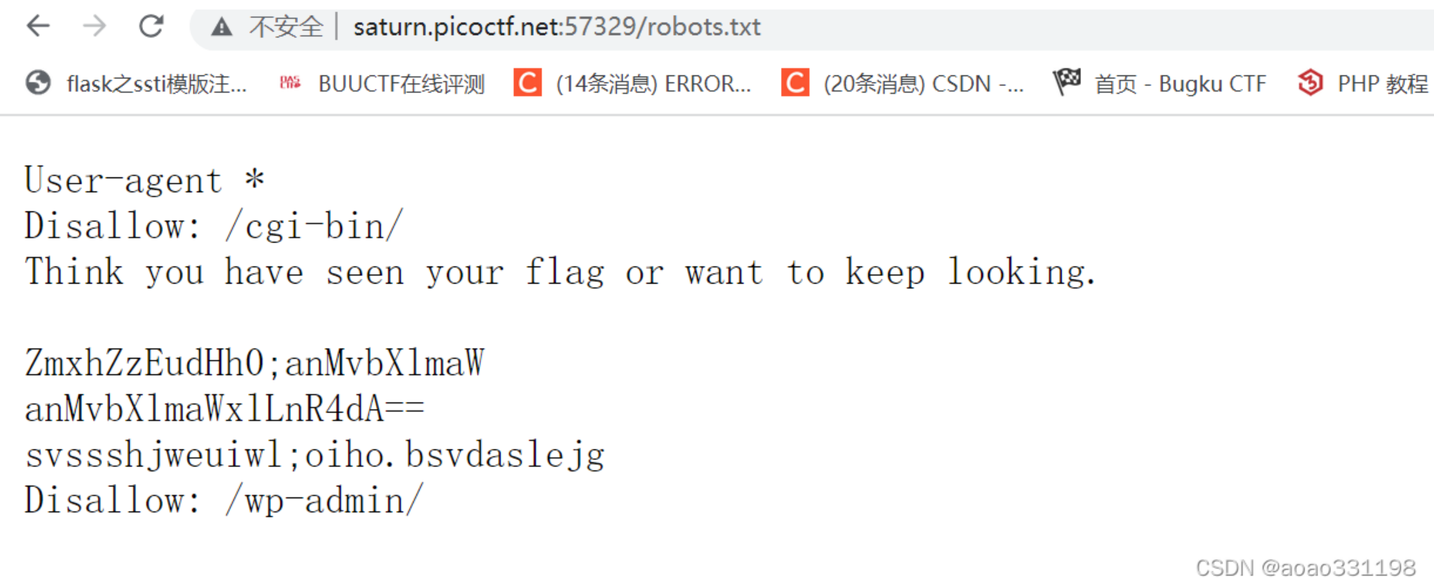
URL: http://saturn.picoctf.net:63397/check.php

Enable POST

Name	Value
☑ Cookie	isAdmin=1

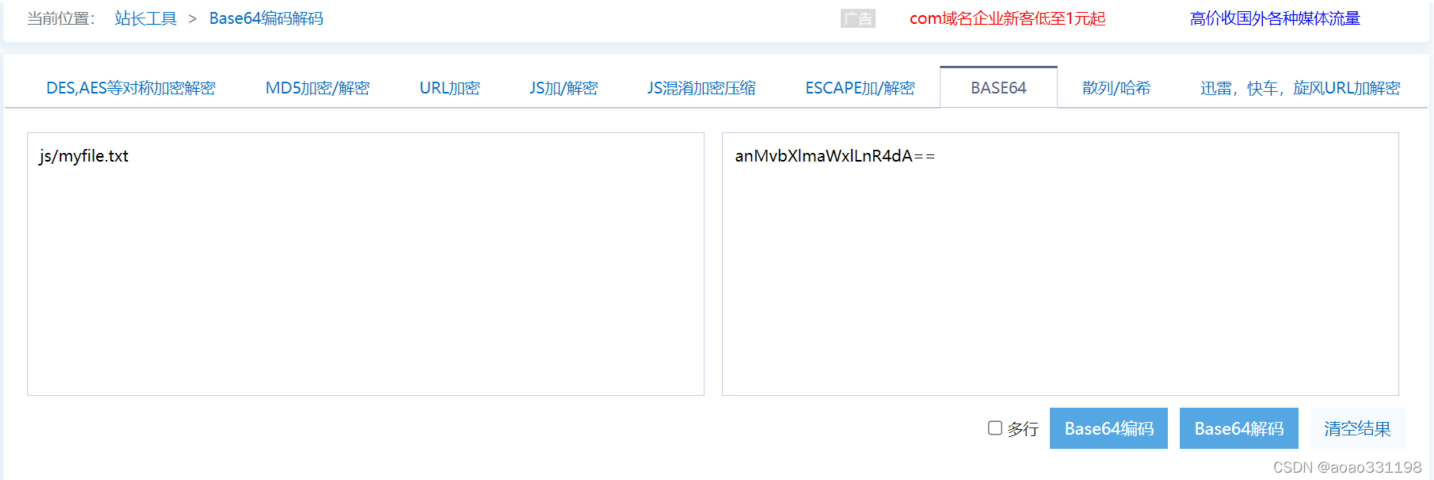
7.Roboto Sans

查看robots.txt



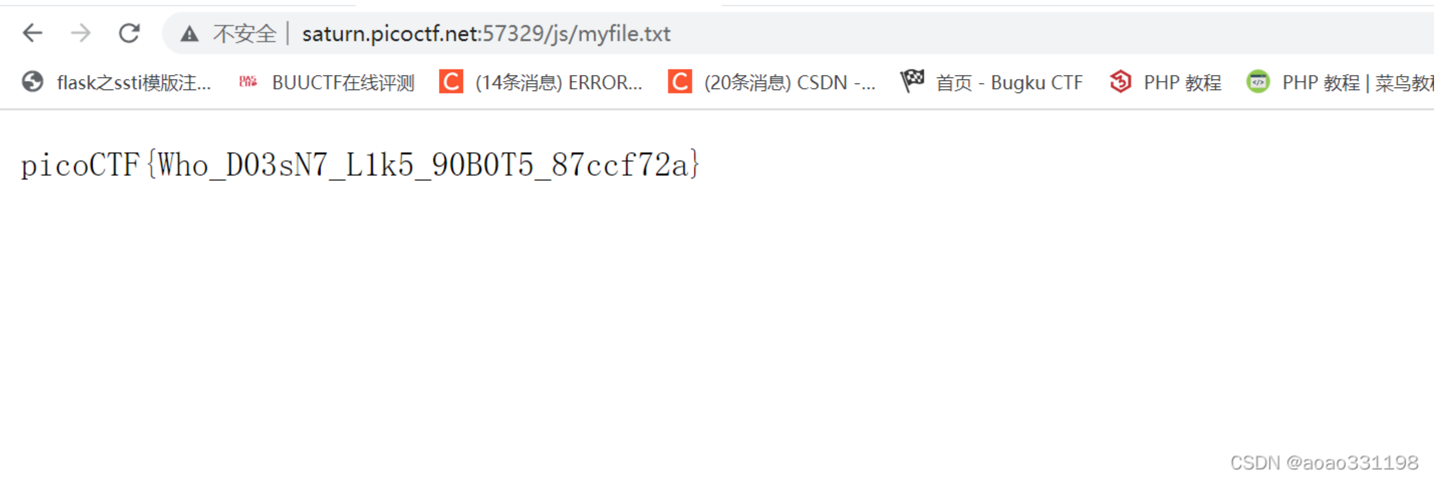
```
User-agent *
Disallow: /cgi-bin/
Think you have seen your flag or want to keep looking.

ZmxhZzEudHh0;anMvbXlmaW
anMvbXlmaWxlLnR4dA==
svssshjweuiwl;oiho.bsvdaslejg
Disallow: /wp-admin/
```

看到一行明显是base64加密



js/myfile.txt

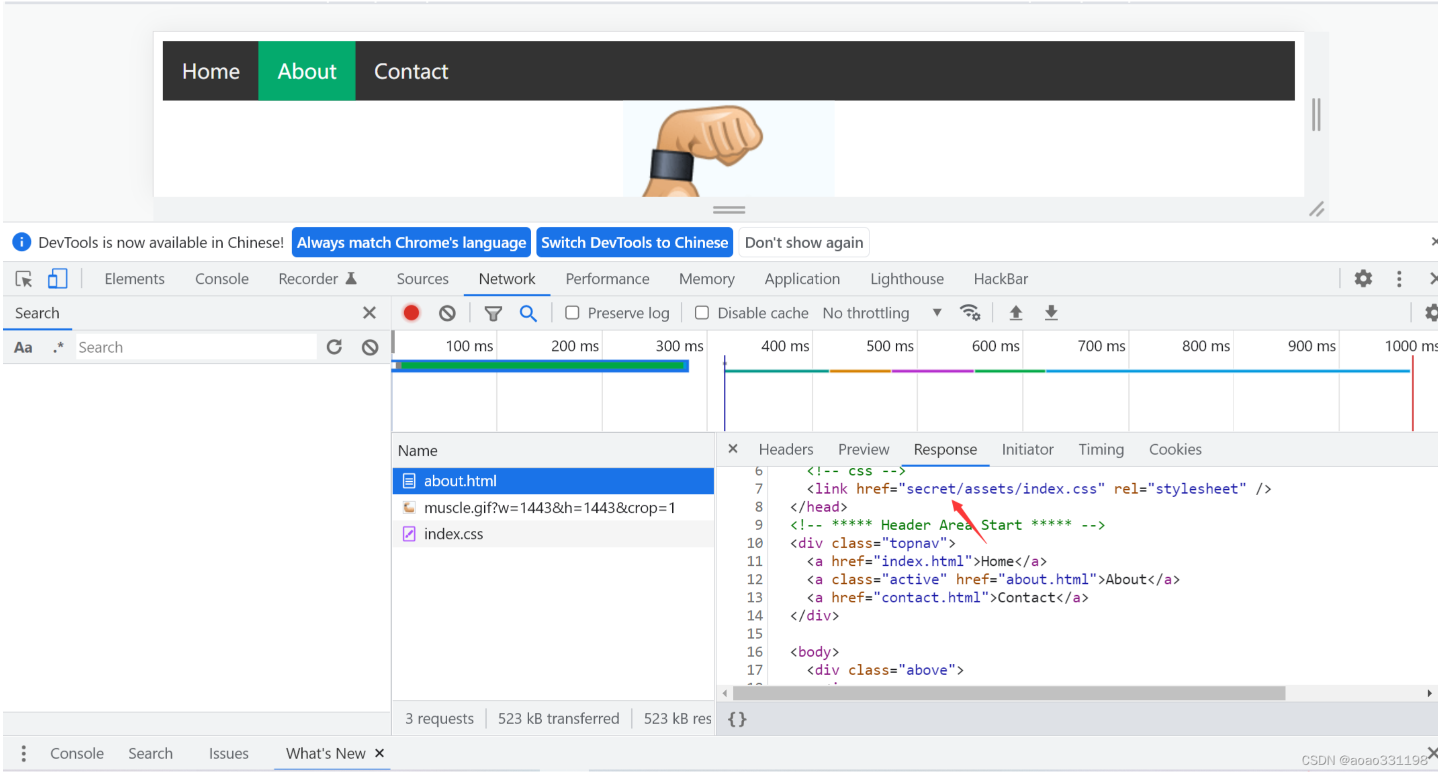
anMvbXlmaWxlLnR4dA==



```
picoCTF{Who_D03sN7_L1k5_90B0T5_87ccf72a}
```

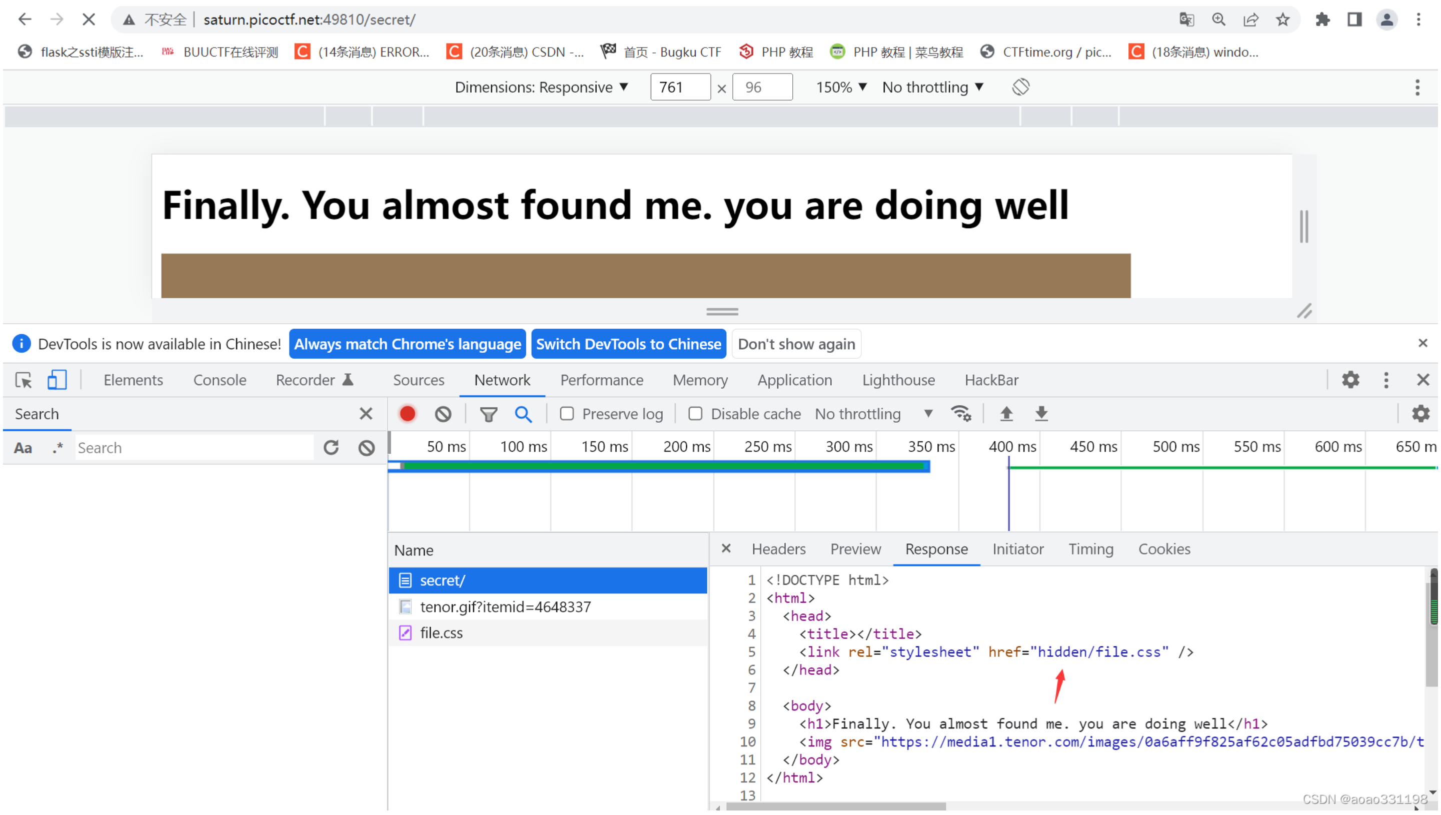
8.Secrets

不停的找就行了



about.html
muscle.gif?w=1443&h=1443&crop=1
index.css

```
  <!-- css -->
  <link href="secret/assets/index.css" rel="stylesheet" />
</head>
<!-- ***** Header Area Start ***** -->
<div class="topnav">
  <a href="index.html">Home</a>
  <a class="active" href="about.html">About</a>
  <a href="contact.html">Contact</a>
</div>

<body>
  <div class="above">
```

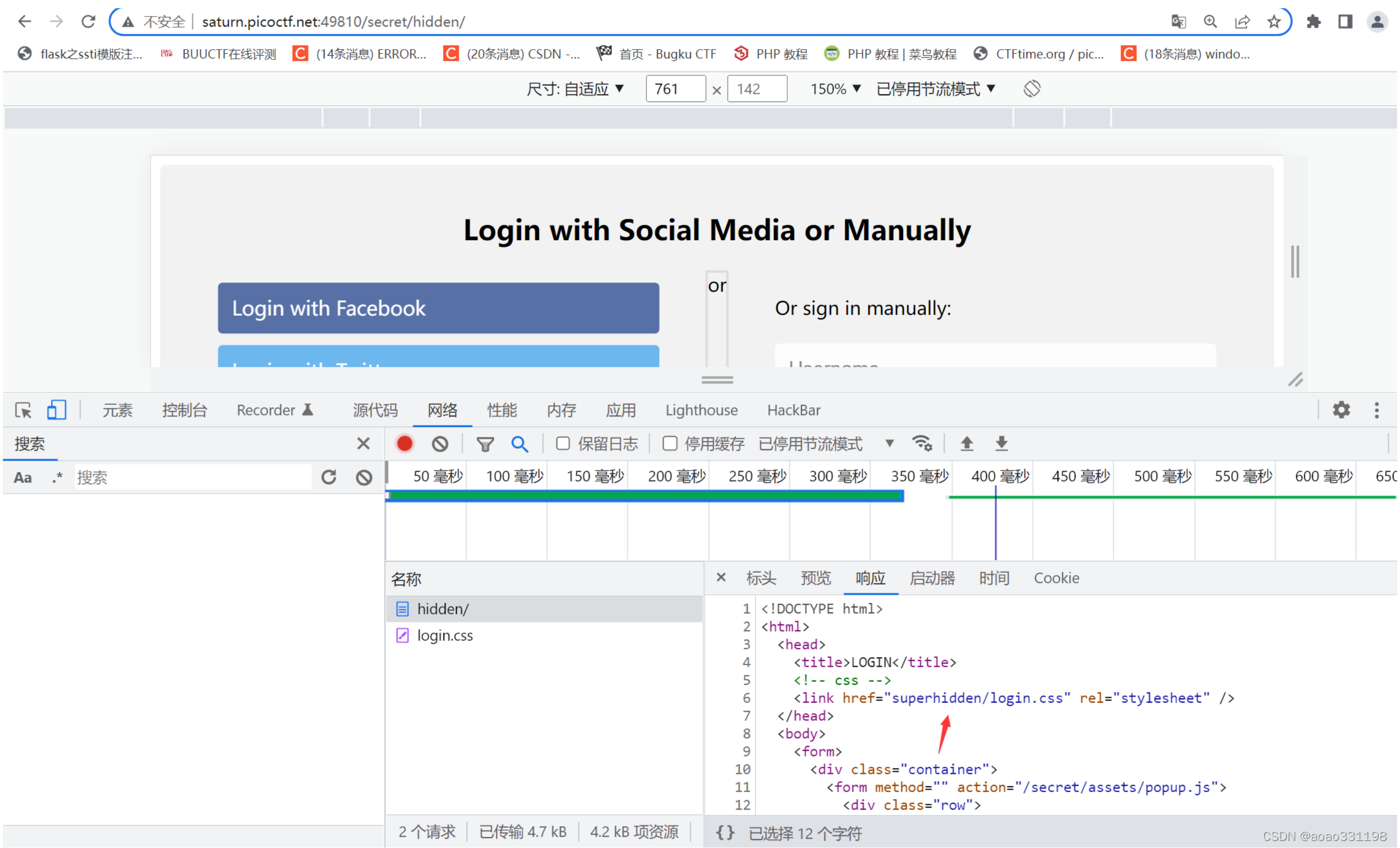


saturn.picoctf.net:49810/secret/

Finally. You almost found me. you are doing well

secret/
tenor.gif?itemid=4648337
file.css

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
    <link rel="stylesheet" href="hidden/file.css" />
  </head>

  <body>
    <h1>Finally. You almost found me. you are doing well</h1>
    <img src="https://media1.tenor.com/images/0a6aff9f825af62c05adfbd75039cc7b/t
  </body>
</html>
```


saturn.picoctf.net:49810/secret/hidden/
尺寸: 自适应 761 × 142 150% 已停用节流模式
元素 控制台 Recorder 源代码 网络 性能 内存 应用 Lighthouse HackBar
名称
hidden/
login.css
标头 预览 响应 启动器 时间 Cookie
2 个请求 已传输 4.7 kB 4.2 kB 项资源
已选择 12 个字符

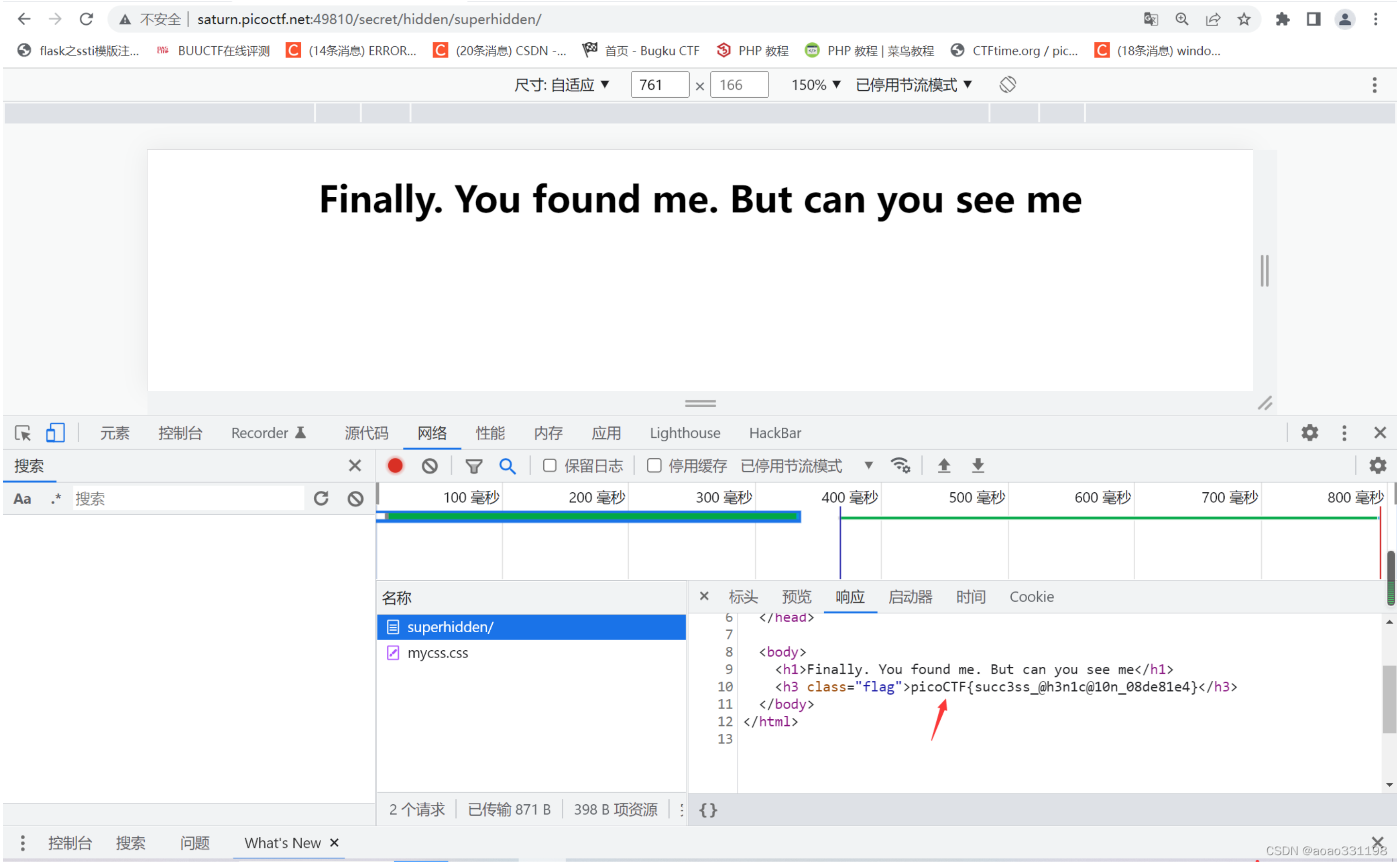
Login with Social Media or Manually

Login with Facebook

or

Or sign in manually:

```
<!DOCTYPE html>
<html>
  <head>
    <title>LOGIN</title>
    <!-- css -->
    <link href="superhidden/login.css" rel="stylesheet" />
  </head>
  <body>
    <form>
      <div class="container">
        <form method="" action="/secret/assets/popup.js">
          <div class="row">
```

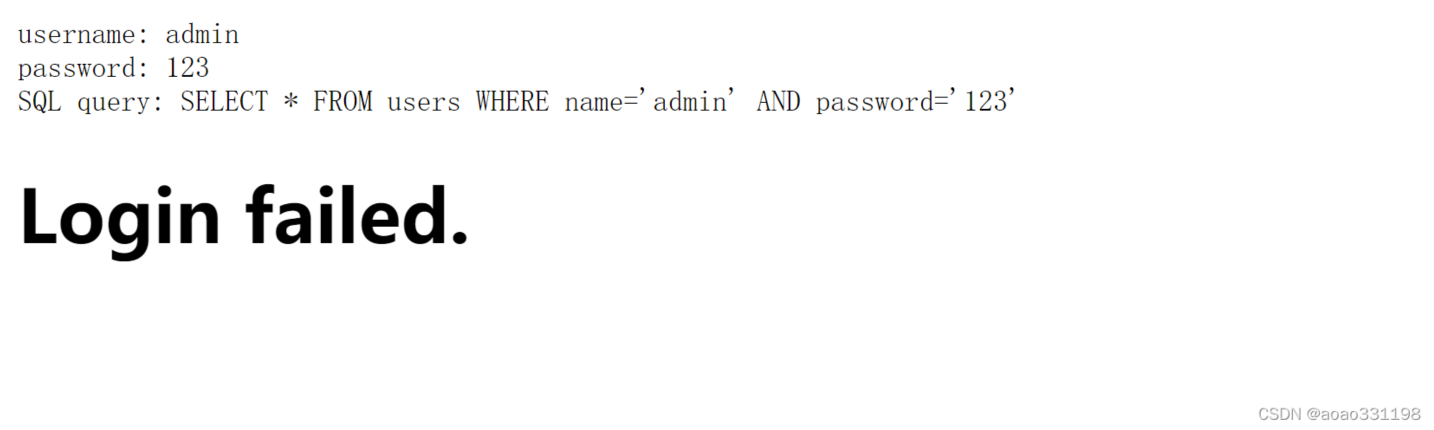

saturn.picoctf.net:49810/secret/hidden/superhidden/
尺寸: 自适应 761 × 166 150% 已停用节流模式
元素 控制台 Recorder 源代码 网络 性能 内存 应用 Lighthouse HackBar
名称
superhidden/
mycss.css
标头 预览 响应 启动器 时间 Cookie
2 个请求 已传输 871 B 398 B 项资源
控制台 搜索 问题 What's New

Finally. You found me. But can you see me

```
  </head>

  <body>
    <h1>Finally. You found me. But can you see me</h1>
    <h3 class="flag">picoCTF{succ3ss_@h3n1c@10n_08de81e4}</h3>
  </body>
</html>
```

9.SQLiLite

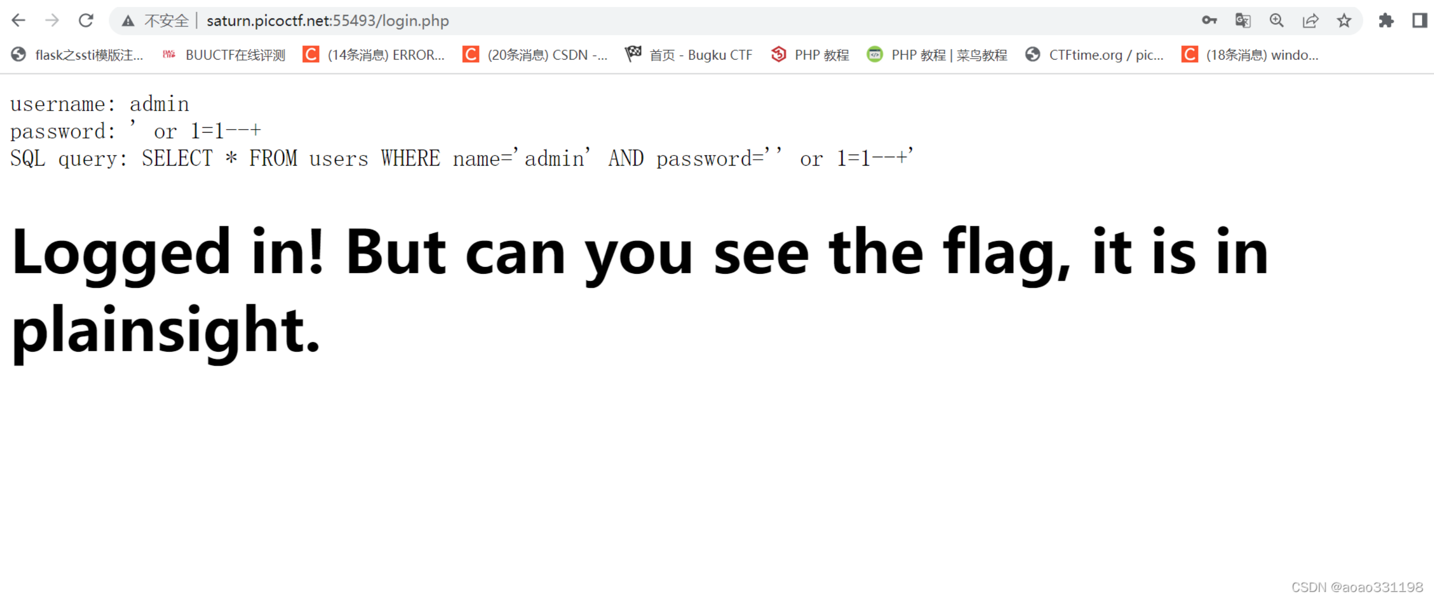
baby级别的SQL注入



```
username: admin
password: 123
SQL query: SELECT * FROM users WHERE name='admin' AND password='123'
```

Login failed.

万能密码



```
username: admin
password: ' or 1=1--+
SQL query: SELECT * FROM users WHERE name='admin' AND password='' or 1=1--+'
```

Logged in! But can you see the flag, it is in plainsight.

Crypto

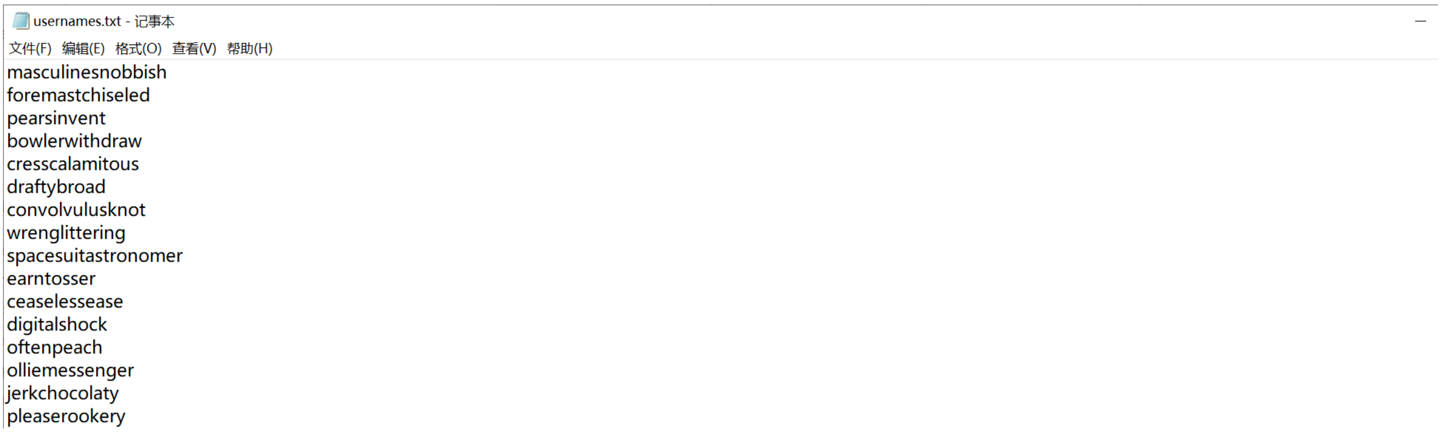
1.basic-mod1 & basic-mod2

第一个就是纯纯的会取模就行了，直接放第二个的脚本

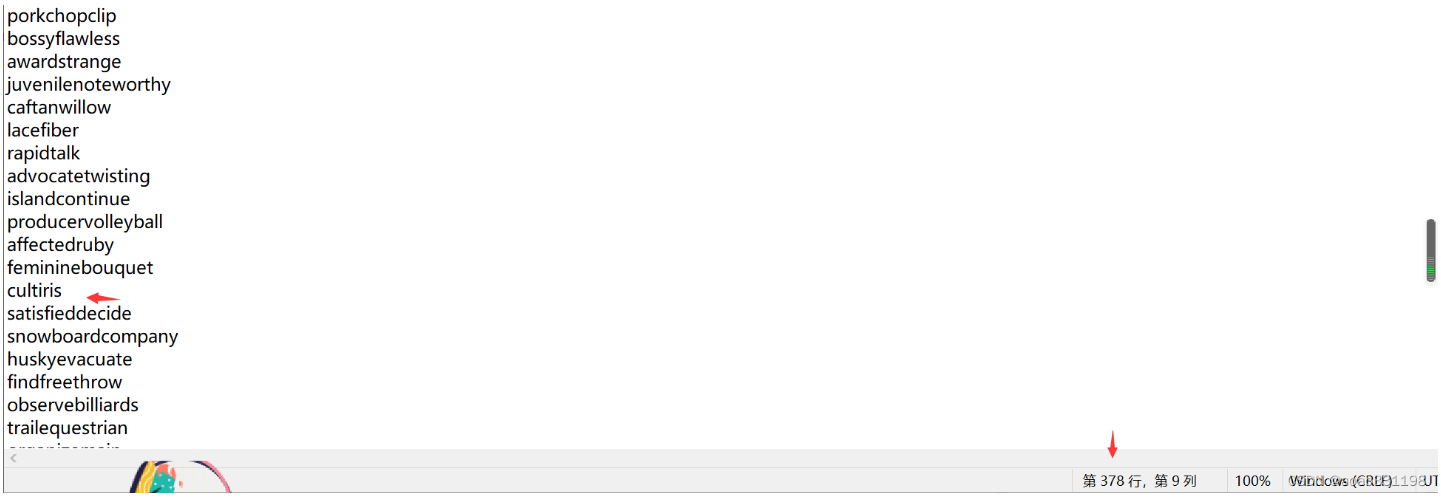
```python
def extended_euclid_gcd(a: int, b: int) -> list:
    """
    Returns [gcd(a, b), x, y] where ax + by = gcd(a, b)
    """
    s, old_s = 0, 1
    t, old_t = 1, 0
    r, old_r = b, a
    while r != 0:
        quotient = old_r // r
        old_r, r = r, old_r - quotient * r
        old_s, s = s, old_s - quotient * s
        old_t, t = t, old_t - quotient * t
    return [old_r, old_s, old_t]


def modular_multiplicative_inverse(a: int, n: int) -> int:
    """
    Assumes that a and n are co-prime, returns modular multiplicative inverse of a under n
    """
    # Find gcd using Extended Euclid's Algorithm
    gcd, x, y = extended_euclid_gcd(a, n)
    # In case x is negative, we handle it by adding extra n
    # Because we know that modular multiplicative inverse of a in range n lies in the range [0, n-1]
    if x < 0:
        x += n
    return x


if __name__ == '__main__':
    list=[186,249,356, 395, 303, 337, 190, 393, 146, 174, 446, 127, 385, 400, 420 ,226, 76, 294, 144 ,90 ,291 ,445 ,137 ]
    for item in list:
        # print(item)
        tmp=modular_multiplicative_inverse(item%41,41)
        #print(tmp)
        if tmp >= 1 and tmp <= 26:
            print(chr(tmp+64),end="")
        elif tmp>=27 and tmp<=36:
            print(int(tmp-27),end="")
        else :
            print("_",end="")
```

2.credstuff

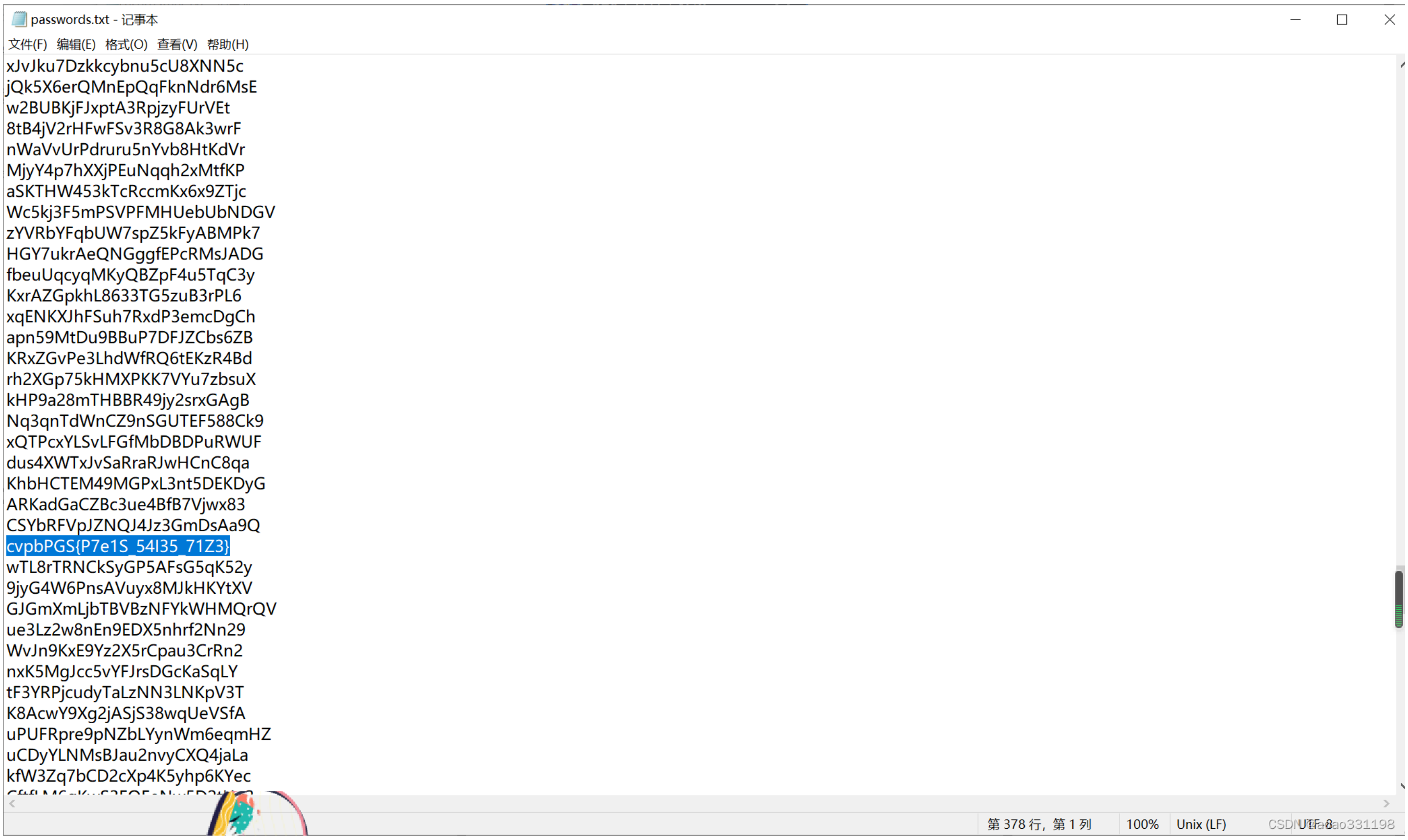
先在username.txt中找到cultiris，是在378行，于是对应的在password.txt中找到对应密码



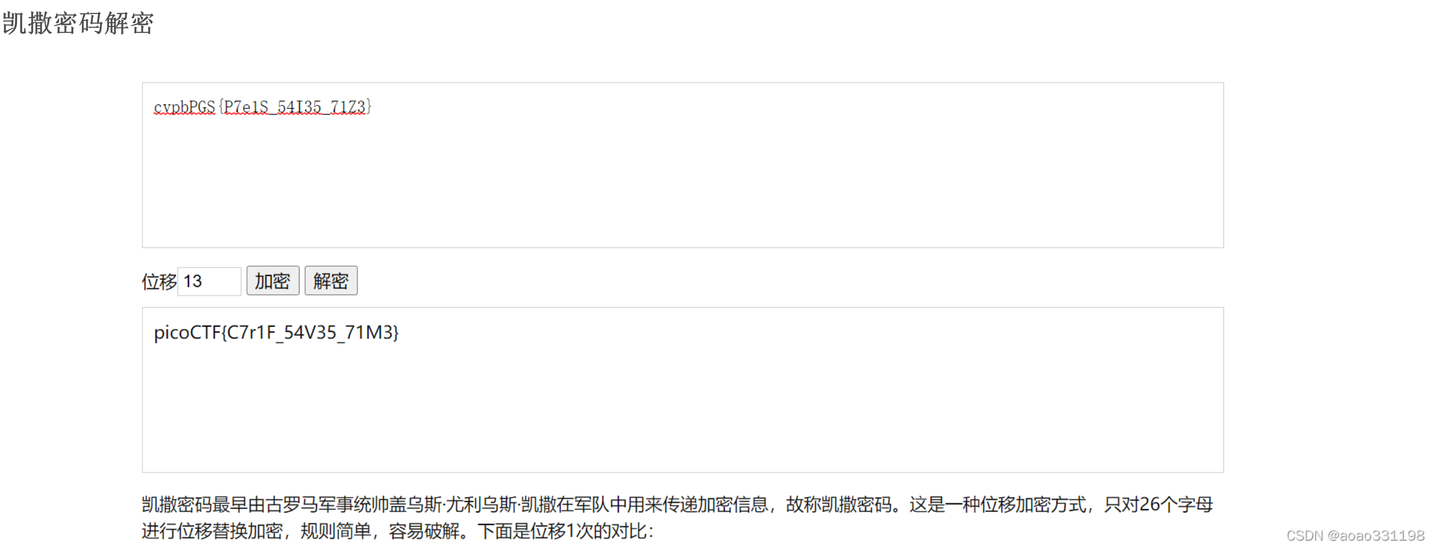
masculinesnobbish
foremastchiseled
pearsinvent
bowlerwithdraw
cresscalamitous
draftybroad
convolvulusknot
wrenglittering
spacesuitastronomer
earntosser
ceaselessease
digitalshock
oftenpeach
olliemessenger
jerkchocolaty
pleaserookery



```
porkchopclip
bossyflawless
awardstrange
juvenilenoteworthy
caftanwillow
lacefiber
rapidtalk
advocatetwisting
islandcontinue
producervolleyball
affectedruby
femininebouquet
cultiris
satisfieddecide
snowboardcompany
huskyevacuate
findfreethrow
observebilliards
trailequestrian
```



```
xJvJku7Dzkkcybnu5cU8XNN5c
jQk5X6erQMnEpQqFknNdr6MsE
w2BUBKjFJxptA3RpjzyFUrVEt
8tB4jV2rHFwFSv3R8G8Ak3wrF
nWaVvUrPdruru5nYvb8HtKdVr
MjyY4p7hXXjPEuNqqh2xMtfKP
aSKTHW453kTcRccmKx6x9ZTjc
Wc5kj3F5mPSVPFMHUebUbNDGV
zYVRbYFqbUW7spZ5kFyABMPk7
HGY7ukrAeQNGggfEPcRMsJADG
fbeuUqcyqMKyQBZpF4u5TqC3y
KxrAZGpkhL8633TG5zuB3rPL6
xqENKXJhFSuh7RxdP3emcDgCh
apn59MtDu9BBuP7DFJZCbs6ZB
KRxZGvPe3LhdWfRQ6tEKzR4Bd
rh2XGp75kHMXPKK7VYu7zbsuX
kHP9a28mTHBBR49jy2srxGAgB
Nq3qnTdWnCZ9nSGUTEF588Ck9
xQTPcxYLSvLFGfMbDBDPuRWUF
dus4XWTxJvSaRraRJwHCnC8qa
KhbHCTEM49MGPxL3nt5DEKDyG
ARKadGaCZBc3ue4BfB7Vjwx83
CSYbRFVpJZNQJ4Jz3GmDsAa9Q
cvpbPGS{P7e1S_54I35_71Z3}
wTL8rTRNCkSyGP5AFsG5qK52y
9jyG4W6PnsAVuyx8MJkHKYtXV
GJGmXmLjbTBVBzNFYkWHMQrQV
ue3Lz2w8nEn9EDX5nhrf2Nn29
WvJn9KxE9Yz2X5rCpau3CrRn2
nxK5MgJcc5vYFJrsDGcKaSqLY
tF3YRPjcudyTaLzNN3LNKpV3T
K8AcwY9Xg2jASjS38wqUeVSfA
uPUFRpre9pNZbLYynWm6eqmHZ
uCDyYLNMsBJau2nvyCXQ4jaLa
kfW3Zq7bCD2cXp4K5yhp6KYec
```



凯撒密码解密

```
cvpbPGS{P7e1S_54I35_71Z3}
```

位移 13

```
picoCTF{C7r1F_54V35_71M3}
```

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：

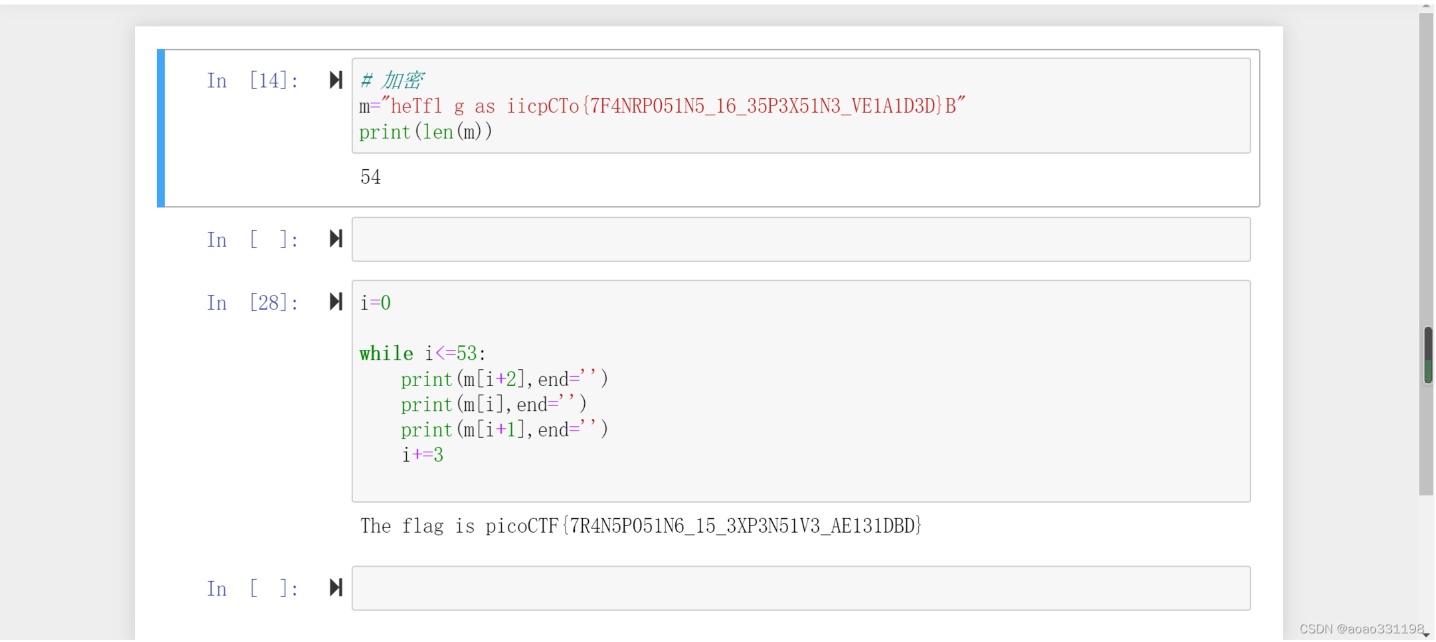
3.substitution0

替换密码
贴上c++脚本

```
#include <iostream>
#include<cstdio>
#include<cstring>



using namespace std;

int main() {
    char str[30]="IADNMLPFYEJSWBZVXUHKGROCQT";

    char str1[30]="ABCDEFGHIJKLMNOPQRSTUVWXYZ";

   char q[]="Kfm vydzDKL{5GA5717G710B_3R0SG710B_A1N36772}";
   char ch;

    for(int i=0;i<strlen(q);i++){
        char ch=q[i];
        if(isupper(ch))
        for(int j=0;j<=25;j++)
        {
            if(str[j]==ch)

        cout<<str1[j];
        }
        else if(islower(ch)){
            for(int j=0;j<=25;j++){
                if(str[j]==ch-32)
                cout<<char(str1[j]+32);
            }
        }
        else cout<<ch;
    }
    return 0;
}
```

4.transposition-trial

三个字符为一组，交换次序



```
In [14]:  # 加密
          m="heTfl g as iicpCTo{7F4NRP051N5_16_35P3X51N3_VE1A1D3D}B"
          print(len(m))

          54

In [ ]:

In [28]:  i=0

          while i<=53:
              print(m[i+2],end='')
              print(m[i],end='')
              print(m[i+1],end='')
              i+=3

          The flag is picoCTF{7R4N5P051N6_15_3XP3N51V3_AE131DBD}

In [ ]:
```

5.Vigenere

维基利亚密码

6.Sum-O-Primes

很水的RSA题目，就利用一个小数学转换

> (p-1)(q-1)=p*q-(p+q)+1

脚本

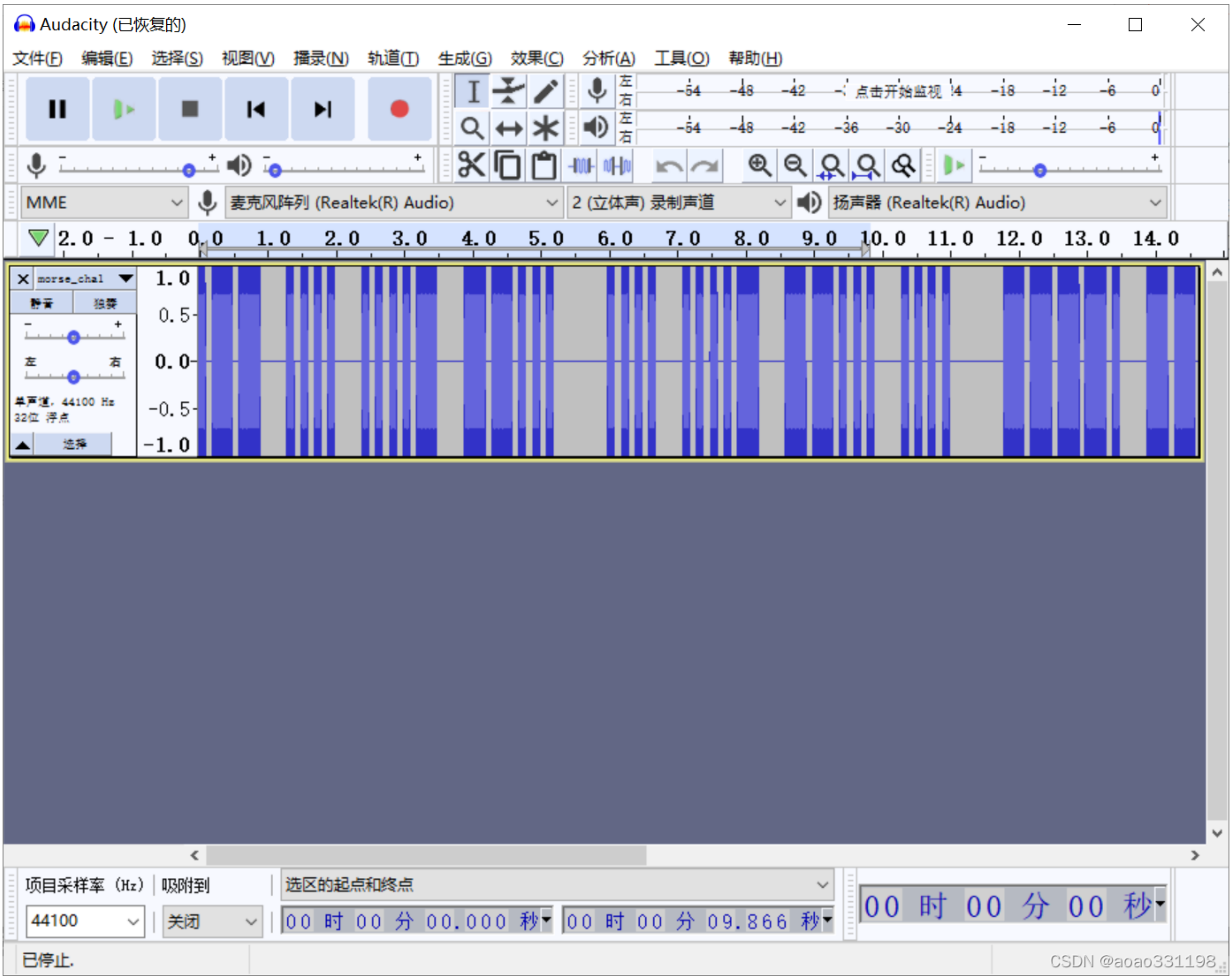
```
import gmpy2
import random
import hashlib
from hashlib import sha256
from Crypto.Util.number import *
import os

x = 0x1603fc8d929cb31edf62bcce2d06794f3efd095accb163e6f2b78941bd8c646d746369636a582aaac77c16a9486881a9e3db26d742
e48c4adcc417ef98f310a0c5433ab077dd872530c3c3c77fe0c080d84154bfdb4c920df9617e986999104d9284516c7babc80dc53718d590
32aefdf41b9be53957dea3f00a386b2666d446e
n = 0x75302ba292dc4bf47ffd690b8edc70ef1fcca5e148b2b9c1b60227788afcfe77a0097929ed3789fe51ac66f678c558244890a09ae4
af3e7d098fd366a1c859edabbff1c9e164d5354968798107ae8518fcaab3743de58a141ffd26c1e16cb09fed1f6b0d68536ec7fba744ed12
0fea8c3a7ac1ebfa55d664d2f321fb44e814650147a9031f3bfa8f69d87393c7d88976d28d147398a355020bcb8e5613f0b29028b77db710
e163ca1019fd3c3a065465ea457adec45243c385d12d3a1de3178f6ca05964be92e8b5bc24d420956de96ccc9ce39e70705660eb6b2f4e67
5aac7d6d7ba45c84223fc5819b37aa85beff1382f1c2c3b97603150f30c17f7e674441
c = 0x562888c70ce9a5c5ed9a0be1b6196f854ba2efcdb6dd0f79319ee9e1142659f90a6bae67481eb0f635f445d3c9889da84639beb84f
f7159dcf4d3a389873dc90163270d80dbb9503cbc32992cb592069ba5b3eb2bbe410a3121d658f18e100f7bd878a25c27ab8c6c15b690fce
1ca43288163c544bfce344bcd089a5f4733acc7dc4b6160718e3c627e81a58f650281413bb5bf7bad5c15b00c5a2ef7dbe7a44cce85ed5b1
becd5273a26453cb84d327aa04ad8783f46d22d61b96c501515913ca88937475603437067ce9dc10d68efc3da282cd64acaf8f1368c1c098
00cb51f70f784bd0f94e067af541ae8d20ab7bfc5569e1213ccdf69d8a81c4746e90c1
e=65537
phi=n-x+1


d=gmpy2.invert(e,phi)
print(d)

m=pow(c,d,n)
print(long_to_bytes(m))
```

7.rail-fence

栅栏密码加密/解密【W型】

8.morse-code

鉴定为纯纯的摩尔斯密码，用hint里面的软件可以较好的分析，不用听。



```
.-- ... ...- --...... ...- --... ...----. ----- -..._.-- ...— ----- ...- ----. ... --...
```

转在线网站破解
在线破解摩尔斯

9.diffie-hellman

DH密钥交换（Diffie–Hellman key exchange）
这道题比较简单，算出密钥为5，作为凯撒密码的移位。
注意的是这道题凯撒密码是移动-5，需要对数字也移位，而且最后flag用小写字母

```python
a = 'H98A9W_H6UM8W_6A_9_D6C_5ZCI9C8I_DI9D987F'
b = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789'
ans = ''
for i in a:
    if i == '_':
        ans += i
    else:
        ans += (b[(b.find(i) - 5) % len(b)])

print('picoCTF{' + ans.lower() + '}')
```

10.Very Smooth

CTF-CYRPTO-RSA-Smooth