

# Writeup\_BugkuCTF\_Web

原创

秦小乙的工作台  于 2019-03-17 20:08:20 发布  140  收藏

分类专栏: [bugku](#) 文章标签: [writeup bugkuctf web](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: [https://blog.csdn.net/m0\\_37980456/article/details/88383560](https://blog.csdn.net/m0_37980456/article/details/88383560)

版权



[bugku](#) 专栏收录该内容

1 篇文章 0 订阅

订阅专栏

## web篇

### 前言

1. web2
2. 计算题
3. web基础\$\_GET
4. web基础\$\_POST
5. 矛盾
6. web3
7. 域名解析

## 前言

0基础上手CTF(现在也很菜), 刷题时也会感觉有些writeup在我当时还不够零基础, 希望能帮助到一些想要涉及该方面的同学。本篇针对Web题, 尽量用通俗易懂的方式讲解, 或者会点出相关知识点方便自主查找, 比较优质的题目会有一些自己的心得。

## 1. web2

打开是动态的滑稽图, 没有头绪直接右键查看网页源代码(或者Ctrl+U)

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <meta name="viewport" content="width=device-width,height=device-height,minimum-scale=1.0,maximum-scale=1.0,ser-scalable=none"/>
6 <title>BK-CTF-WEB2</title>
7
8 <style type="text/css">
9 body { margin: 0; padding: 0; position: relative; background-image: url(images/xh.jpg); background-position: center; /*background-repeat: no-repeat;*/ width: 100%; height: 100%; background-size: 100% 100%; }
10
11
12
13 </style>
14
15 </head>
16 <body id="body" onload="init()">
17 <!--[[[ag_KEV (Web-2-bugKas0Nk1e9100)]]-->
18 <script type="text/javascript" src="js/ThreeCanvas.js"></script>
19 <script type="text/javascript" src="js/Show.js"></script>
20
21
22 <script type="text/javascript">
23 var SCREEN_WIDTH = window.innerWidth; //
24 var SCREEN_HEIGHT = window.innerHeight;
25 var container;
26 var particle; //粒子
```

[https://blog.csdn.net/m0\\_37980456](https://blog.csdn.net/m0_37980456)

注释内得到flag。

## 2. 计算题

输入时发现只能输入一位数，很明显是网页css设置了最大长度(JavaWeb知识)。

66+35=?

来源: [BugKu-ctf](#)

```
Elements Console Sources Network Performance Memory Application Security Audits
<!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>...</head>
<body>
  <span id="code" class="code" style="background: rgb(202, 131, 214); color: rgb(65, 119, 65);">66+35=?</span>
  <input type="text" class="input" maxlength="1" == $0
  <button id="check">验证</button>
  <div style="text-align:center;">...</div>
  <script src="js/jquery-1.12.3.min.js"></script>
  <script type="text/javascript" src="js/code.js"></script>
</body>
</html>
```

[https://blog.csdn.net/m0\\_37980456](https://blog.csdn.net/m0_37980456)

右键“检查”找到此处的maxlength双击改为所需位数，然后输入正确答案即可得到flag。

66+35=?

125.200.87.240:8002 业务小

flag(CTF-bugku-0032)

```
Elements Console Sources Network Performance Memory Application Security Audits
<!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>...</head>
<body>
  <span id="code" class="code" style="background: rgb(202, 131, 214); color: rgb(65, 119, 65);">66+35=?</span>
  <input type="text" class="input" maxlength="3" == $0
```

[https://blog.csdn.net/m0\\_37980456](https://blog.csdn.net/m0_37980456)

## 3. web基础\$\_GET

和第4题都属于两种页面提交方式。

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

php语言，语意大致为选择判断，get“what”时输出what指代内容，get“flag”时输出flag。因此利用URL定义，payload为：<http://123.206.87.240:8002/get/?what=flag>（即在网址后方直接加上?what=flag）即可获得flag。

## 4.web基础\$\_POST

post不能简单在网址修改，故使用在线工具实现post访问提交内容(工具地址：<http://coolaf.com/tool/post>)

The screenshot shows an online HTTP client interface. At the top, the URL is set to `http://123.206.87.240:8002/post/` and the method is `POST`. The encoding is set to `UTF-8`. Under the `Body` tab, the `x-www-form-urlencoded` radio button is selected. A form field contains `what` as the key and `flag` as the value. Below the form, there are buttons for `提交` (Submit), `生成文档` (Generate Document), `导出历史` (Export History), and `清空表单` (Clear Form). The response section shows a `Status:200 OK` and the response body is displayed in a code editor with the following content:

```
1 $what=$_POST['what'];<br>  
2 echo $what;<br>  
3 if($what=='flag')<br>  
4 echo 'flag{****}';<br>  
5  
6  
7 flagflag{bugku_get_ssseint67se}
```

成功获得flag。

## 5. 矛盾

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

依旧是php代码，大意为is\_numeric函数判断num是否为数字，数字不输出，非数字再判断num若为1则输出flag。可是如果num为1在第一个判断就无法通过了！！

这里面涉及到了\*\*%00截断\*\*这个知识点：URL中的%00（形如%xx），web server会把它当作十六进制处理，然后将该十六进制数据hex（00）“翻译”成统一的ASCII码值“null”。

构建payload:

```
index1.php?num=1%00
```

这使得第一次判断时，num后有空不为数字，第二次判断时num值也为1,符合所有要求得到flag。

## 6. web3

这道题打开后是不断弹出的alert警告，玩了一会儿发现没什么动静，查看网页源代码发现注释里有一段特殊字符，不理解什么意思，去查了一下这种“&#x;”格式，获知这是Unicode编码方式。

```
alert("来找我吧");
alert("flag就在这里");
alert("来找我吧");
alert("flag就在这里");
alert("来找我吧");
alert("flag就在这里");
alert("来找我吧");
alert("flag就在这里");
alert("来找我吧");
alert("flag就在这里");
alert("来找我吧");
alert("flag就在这里");
alert("来找我吧");
alert("flag就在这里");
alert("来找我吧");
alert("flag就在这里");
alert("来找我吧");
alert("flag就在这里");
alert("来找我吧");
alert("flag就在这里");
alert("来找我吧");
alert("flag就在这里");
alert("来找我吧");
alert("flag就在这里");
alert("来找我吧");
alert("flag就在这里");
alert("来找我吧");
alert("flag就在这里");
alert("来找我吧");
<!--&#75;&#69;&#89;&#123;&#74;&#50;&#115;&#97;&#52;&#50;&#97;&#104;&#74;&#75;&#45;&#72;&#83;&#49;&#49;&#73;&#73;&#73;&#125;-->
</script>
```

同样利用工具进行Unicode转中文即可获得flag。

Unicode编码    UTF-8编码    URL编码/解码    Unix时间戳    Ascii/Native编码互转

KEY(U2sa42ahJK-HS11!!!)    KEY(U2sa42ahJK-HS11!!!)

ASCII 转 Unicode    Unicode 转 ASCII    Unicode 转 中文    中文 转 Unicode    清空结果

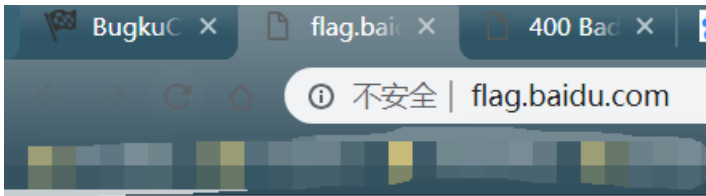
[https://blog.csdn.net/m0\\_9799499](https://blog.csdn.net/m0_9799499)

## 7. 域名解析

题目提示很清楚，需要将flag.baidu.com解析到123.206.87.240  
第一次打开无连接，即DNS访问无法连接，所以利用hosts文件修改，

```
34 .....
35 123.206.87.240 flag.baidu.com
```

这里相当于添加指定数字ip路径到字符串网址，通过这种方式即可进入该页面，即可获得flag。



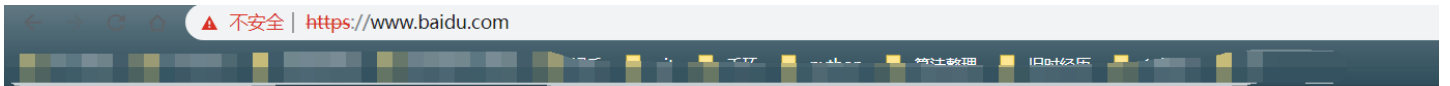
KEY{DSAHDSJ82HDS2211}

[https://blog.csdn.net/m0\\_37980456](https://blog.csdn.net/m0_37980456)

这道题解题关键在于搞清楚域名解析、hosts、IP地址、DNS解析的原理，为什么hosts里指定的数字ip可以打开对应字符串网址（域名）？直接输入相应数字ip为什么没有效果？

hosts的特点是优先于DNS解析查询域名的IP地址，而每一个IP地址唯一对应网络中众多的主机之一（同时又可以对应多个域名）。因此当IP地址未与域名通过DNS绑定时，如果单纯输入IP就可能因为一个IP对应多个网站，没办法查找到所需网站额，而这时候的域名和网站又是一一对应的，所以才可以出现这种需要IP先与域名绑定（这里是利用hosts文件实现绑定）再通过域名查找到真正所需网站。

相应的也引申出hosts文件的问题，我们常修改hosts文件来实现翻墙或者使用其他脚本，但hosts中毒可能导致的DNS劫持也必须提防，可能会出现网页地址没有问题但实际使用的ip地址指向其他网站。当然，现在的大网站一般也都有相应的审核机制，譬如：



## 您的连接不是私密连接

攻击者可能会试图从 [www.baidu.com](https://www.baidu.com) 窃取您的信息（例如：密码、通讯内容或信用卡信息）。[了解详情](#)

NET::ERR\_CERT\_AUTHORITY\_INVALID

您可以选择向 Google 发送一些系统信息和网页内容，以帮助我们改进安全浏览功能。[隐私权政策](#)

[隐藏详情](#)

[重新加载](#)

[www.baidu.com](https://www.baidu.com) 通常会使用加密技术来保护您的信息。Google Chrome 此次尝试连接到 [www.baidu.com](https://www.baidu.com) 时，此网站发回了异常的错误凭据。这可能是因为有攻击者在试图冒充 [www.baidu.com](https://www.baidu.com)，或 Wi-Fi 登录屏幕中断了此次连接。请放心，您的信息仍然是安全的，因为 Google Chrome 尚未进行任何数据交换便停止了连接。

您目前无法访问 [www.baidu.com](https://www.baidu.com)，因为此网站使用了 HSTS。网络错误和攻击通常是暂时的，因此，此网页稍后可能会恢复正常。

[https://blog.csdn.net/m0\\_37980456](https://blog.csdn.net/m0_37980456)