

# XCTFweb笔记

原创

[BL\\_zshaom](#) 已于 2022-02-11 14:09:56 修改 105 收藏

文章标签: [git](#)

于 2022-01-22 16:30:04 首次发布

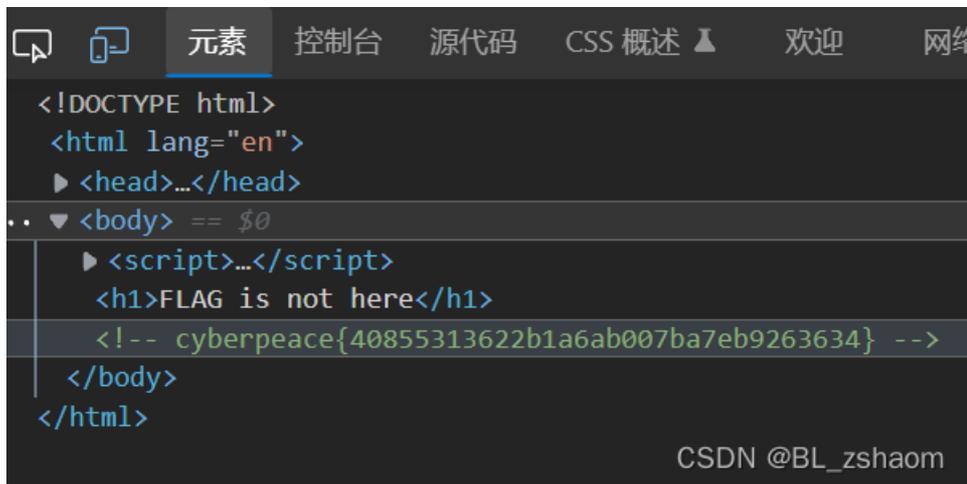
版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: [https://blog.csdn.net/BL\\_zshaom/article/details/122635825](https://blog.csdn.net/BL_zshaom/article/details/122635825)

版权

## 1、view\_source

题目上说右键不能使用, 我们直接打开开发者管理工具, 查看元素就能找到flag



```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body> == $0
    <script>...</script>
    <h1>FLAG is not here</h1>
    <!-- cyberpeace{40855313622b1a6ab007ba7eb9263634} -->
  </body>
</html>
```

CSDN @BL\_zshaom

## 2、robots

看见题目我们就要想到robots.txt协议

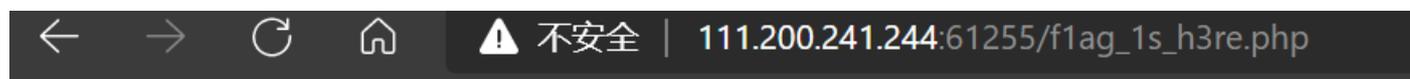


```
User-agent: *  
Disallow:  
Disallow: flag_1s_h3re.php
```

CSDN @BL\_zshaom

详细关于robots.txt协议请看:[https://blog.csdn.net/BL\\_zshaom/article/details/122634197](https://blog.csdn.net/BL_zshaom/article/details/122634197)

然后我们打开flag\_1s\_h3re.php



cyberpeace{697c286816c81a220151057efaddb281}

CSDN @BL\_zshaom

### 3、 backup

进入题目网页，显示：你知道index.php的备份文件名吗？

很显然这是备份文件。

常见的备份文件的后缀：git .svn .swp .svn .~ .bak .bash\_history。

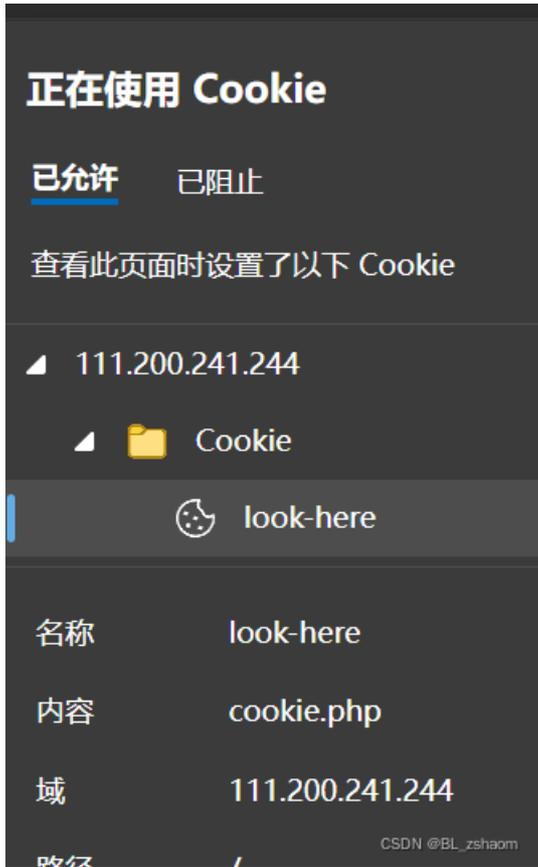
我们其实可以依次试一试，不过题目上是backup，所以我们可直接用.bak

网址里输入：/index.php.bak可以下载一个文件

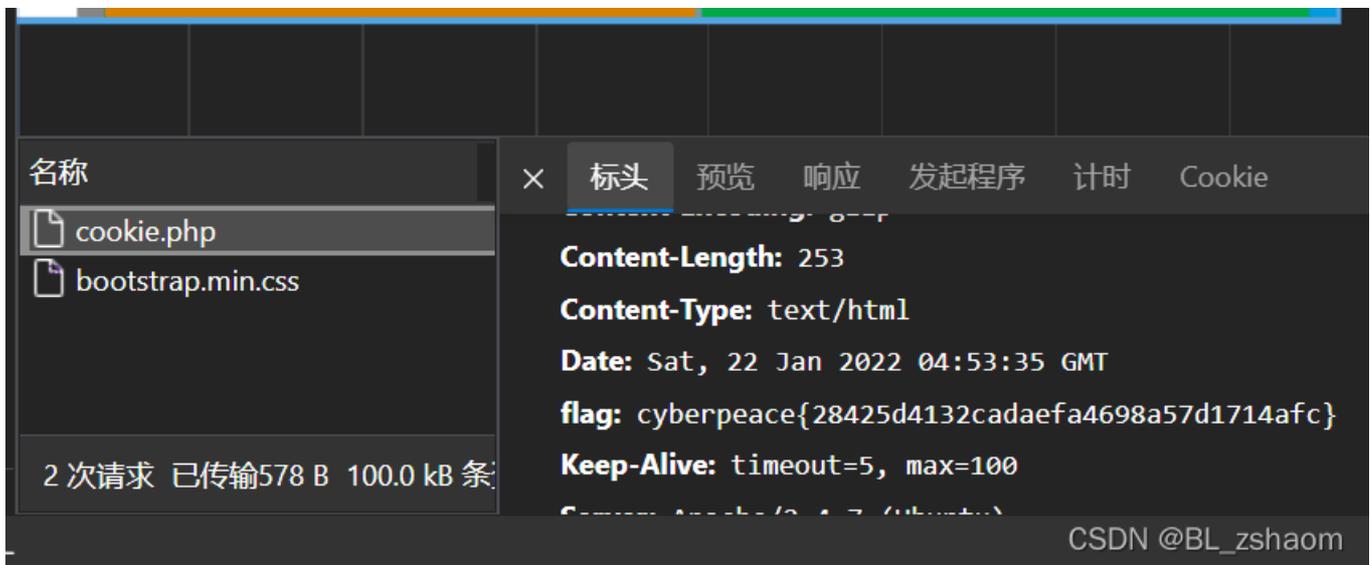
- ①：将后缀名改为html，在网页源代码中找到flag
- ②：用万能记事本打开，直接看到flag

### 4、 cookie

有题目直接查看网页cookie，可以看到



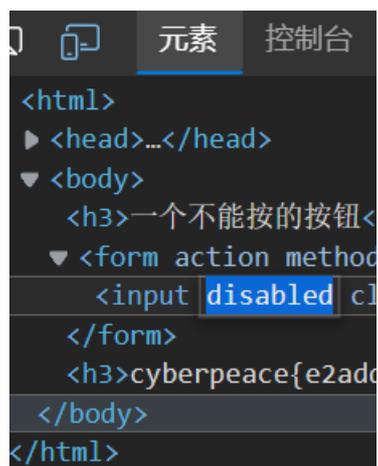
我们直接在网址里输入/cookie.php，可以进入一个网页，里面显示：See the http response  
这是让我们查看http响应，我们打开开发者管理工具——网络——标头里的响应头



就能找到flag

## 5、disabled\_button

直接打开开发者管理工具，点击元素，把disabled改为abled就可以点button了，然后得到flag（别问语法，问就是摆）



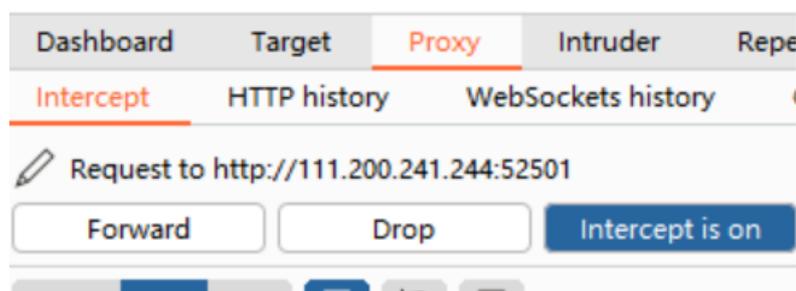
```
<html>
  <head>...</head>
  <body>
    <h3>一个不能按的按钮</h3>
    <form action method="get">
      <input disabled type="button" value="Submit" />
    </form>
    <h3>cyberpeace{e2add...}</h3>
  </body>
</html>
```

## 6、weak\_auth

这道题需要用到burpsuite软件。

①：用软件自带的浏览器打开题目网站

②：选中proxy选项，再点击intercept is off（off 变为on）之后输入张账密

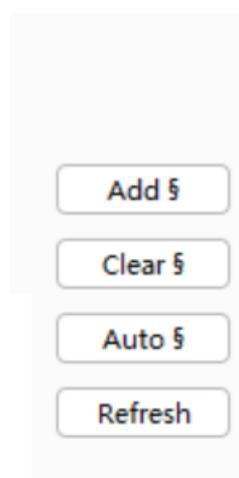


③：login之后，软件就有数据了，然后右键点击send to

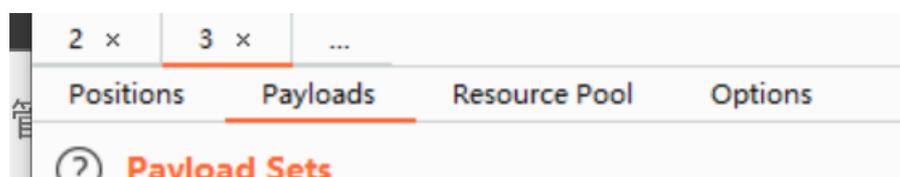


intruder。

④：之后看intruder界面，点击add



⑤：然后点击payloads，在里面payload options里点击load上传字典  
然后点击右边的start attack进行破解。破解之后看result就行



## 7、simple\_php

```
<?php
show_source(__FILE__)
include("config.php")
$a=@$_GET['a'];
$b=@$_GET['b'];
if($a==0 and $a){
    echo $flag1;
}
if(is_numeric($b)){
    exit();
}
if($b>1234){
    echo $flag2;
}
?>
```

#### 关于a:

a的值为零

直接a=a就行，因为使用==的时候会将字符串类型转为相同在进行比较，而a的值是由字符串开头的数字决定，但a前没有数字就默认为0，所以a=a。

#### 关于b:

带的字符串的值必须为数字（也就是要b=123b或者1234565b这样的）

然后b的值要大于1234（也就是12345b）

这样就能解出flag了

## 8、get\_post

get和post就是http两种请求方式（除此之外还有head、put、deldete、trace、connect、option这六种请求方式）

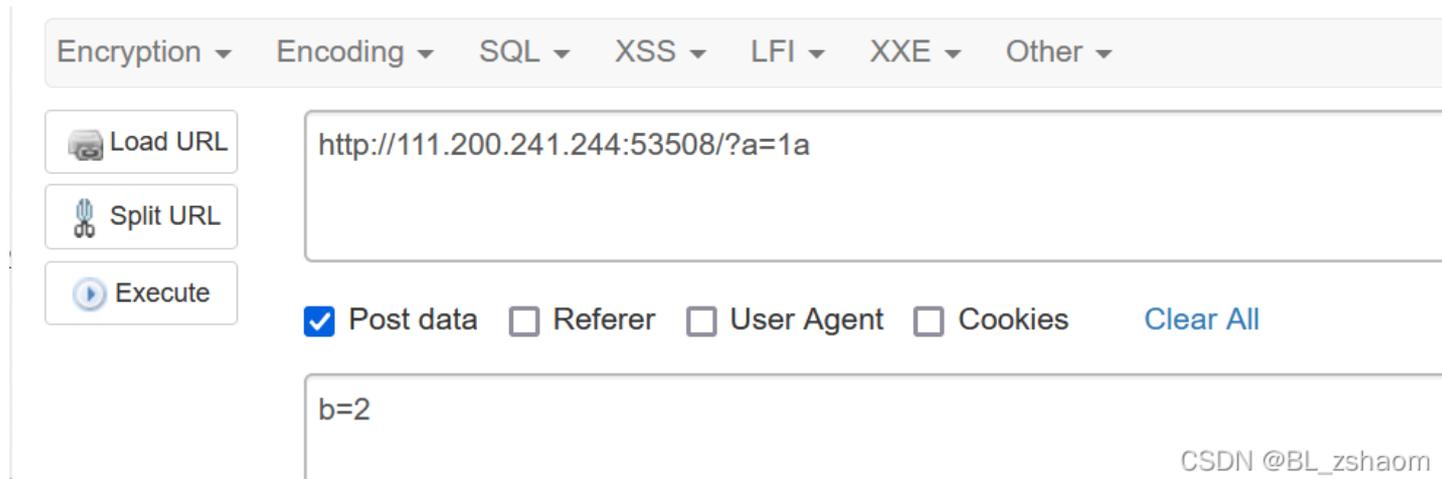
**get描述：** 请求指定的页面信息，并返回实体主体。

**post描述：** 向指定资源提交数据进行处理请求（例如提交表单或者上传文件）。数据被包含在请求体中。POST请求可能会导致新的资源的建立和/或已有资源的修改。（以上来自菜鸟教程）

## 请用GET方式提交一个名为a,值为1的变量

## 请再以POST方式随便提交一个名为b,值为2的变量

post我们需要用到火狐浏览器里的hackerbar插件



即可得到flag

## 9、xff\_referer

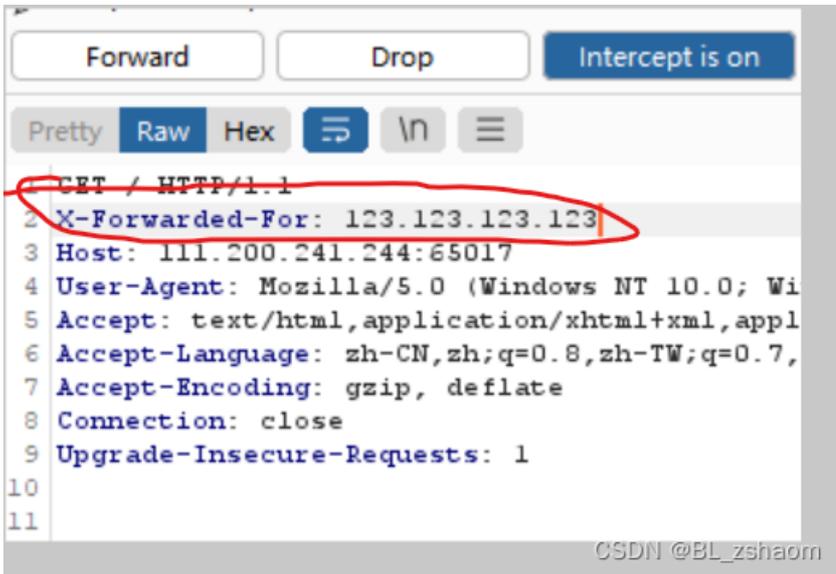
**xff:** 是header里面的一部分，也就是浏览器访问一个网站的ip地址

**referer:** 是一个链接或者网页的来源

这两个可以通过人工伪造，我们用burp抓包即可

我们在相应头的位置多增添一行：

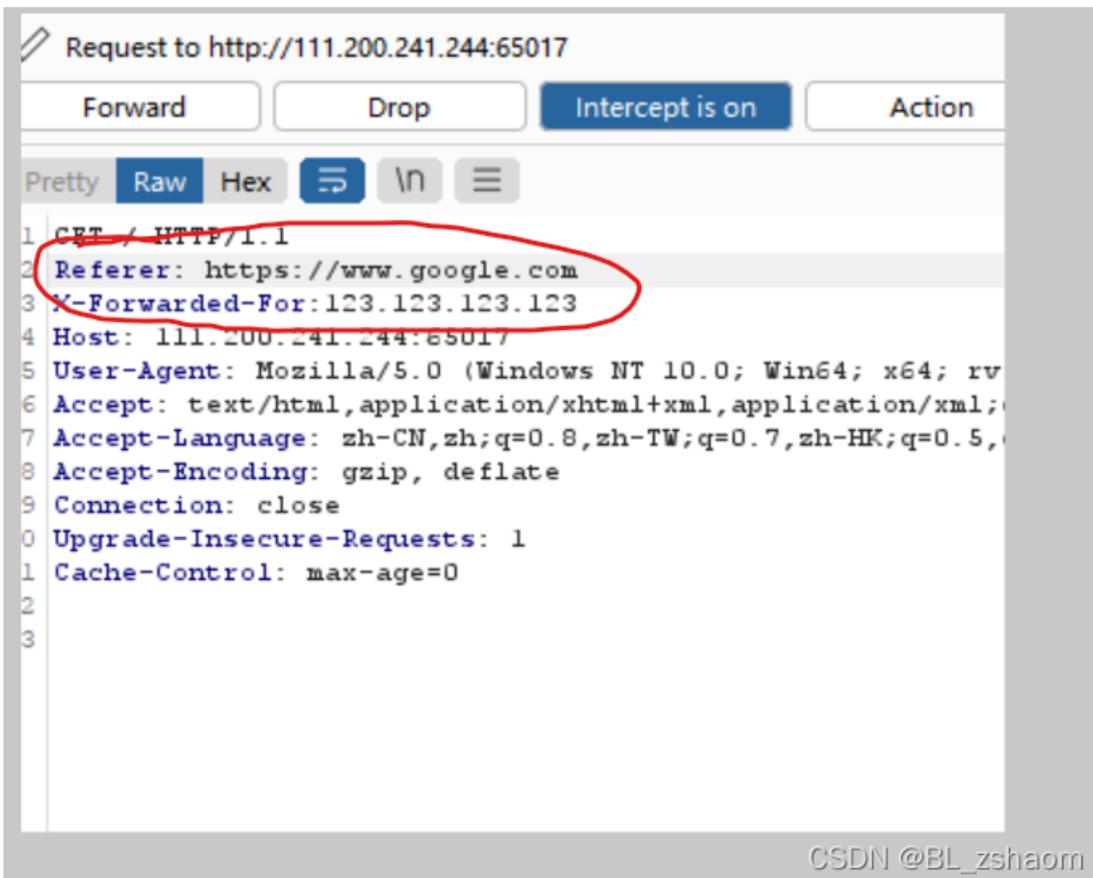
X-Forwarded-For: 123.123.123.123



然后放包，显示

必须来自<https://www.google.com>

所以我们在请求头再加上：



然后放包就能出来flag

## 10、webshell

首先看到题目，我就想到蚁剑

所以打开蚁剑，添加数据，通过网页上给的一句话木马得知连接密码是shell

URL地址 *	<input type="text" value="http://111.200.241.244:56269/"/>
连接密码 *	<input type="text" value="shell"/>

打开，发现flag

	名称
	flag.txt
	index.php

## 11、command\_execution

我们用火狐的hackbar进行post注入

①以本地的ip地址post

The screenshot shows the Hackbar tool interface. At the top, there is a text input field containing the text "请输入需要ping的地址". Below this is a button labeled "PING". The main area displays the output of a terminal command: `ping -c 3 127.0.0.1`. The output shows three successful ping requests to 127.0.0.1 with varying response times. Below the output, there is a summary: `--- 127.0.0.1 ping statistics ---`, `3 packets transmitted, 3 received, 0% packet loss, time 1999ms`, and `rtt min/avg/max/mdev = 0.050/0.069/0.089/0.015 ms`. The bottom part of the screenshot shows the Hackbar tool's navigation bar with icons for "查看器", "控制台", "调试器", "网络", "样式编辑器", "性能", and "内存". Below the navigation bar is a menu with options: "Encryption", "Encoding", "SQL", "XSS", "LFI", "XXE", and "Other". The main workspace contains a "Load URL" button, a "Split URL" button, and an "Execute" button. The URL field contains `http://111.200.241.244:60115/`. Below the URL field, there are checkboxes for "Post data" (checked), "Referer", "User Agent", and "Cookies". The "Post data" field contains `target=127.0.0.1`. In the bottom right corner, there is a watermark: "CSDN @BL\_zshaom".

②用ls命令查看目录:



③ls命令home目录发现flag



④所以我们cat home目录里面的flag.txt



## 12、simple\_js

打开网址发现给了一个输入框，随便输入，提示错误  
直接看源代码，我也看不懂js，参考了大佬们的wp  
自己就发现for语句的结果都是输出 String.fromCharCode  
而 String.fromCharCode在源代码里是这个：

```
String["fromCharCode"](dechiffre("\\x35\\x35\\x2c\\x35\\x36\\x2c\\x35\\x34\\x2c\\x37\\x39\\x2c\\x31\\x31\\x35\\x2c\\x36\\x39\\x2c\\x31\\x31\\x34\\x2c\\x31\\x31\\x36\\x2c\\x31\\x30\\x37\\x2c\\x34\\x39\\x2c\\x35\\x30"));
```

这应该就是flag，而且有的大佬说，不论输入什么，都会显示错误，那就应该是在源代码里找flag。  
然后16进制转换10进制得到：55,56,54,79,115,69,114,116,107,49,50

### 16进制转换文本 / 文本转16进制

<pre>\\x35\\x35\\x2c\\x35\\x36\\x2c\\x35\\x34\\x2c\\x37\\x39\\x2c\\x31\\x31\\x35\\x2c\\x36\\x39\\x2c\\x31\\x31\\x34\\x2c\\x31\\x31\\x36\\x2c\\x31\\x30\\x37\\x2c\\x34\\x39\\x2c\\x35\\x30</pre>	<p>字符串转16进制 &gt;&gt;</p> <p>16进制转字符串 &gt;&gt;</p>	<pre>0505,0506,0504,0709,010105,0609,010104,010106,01007,0409,0500</pre>
---	---	--

CSDN @BL\_zshaom

然后asciill码转换

<pre>刑八\寸旦子1J</pre>
<pre>55,56,54,79,115,69,114,116,107,49,50</pre>

786OsErk12

将转换后的结果放到花括号里就行