

攻防世界——web新手区（全解）

原创

小白一枚多多关注  于 2020-12-20 21:48:43 发布  4170  收藏 19

分类专栏: [CTF](#) 文章标签: [安全 web 信息安全](#)

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_45766004/article/details/111222553

版权



[CTF 专栏收录该内容](#)

10 篇文章 2 订阅

订阅专栏

当前网络安全形式越来越严重, 我国也越来越重视, 现在国内乃至国际上各个网络攻防大赛层出不穷, 但是练习平台却还是很稀缺, 可以说目前网上能够练习的平台也就只有几家, 大多数的院校它们有自己的练习平台但并不公开!

今天小白介绍的是业内很有名气的练习平台——攻防世界, 它里面有web、pwn、music、reverse、crypto和mobile六个大类, 因为小白主要做的是web渗透和后渗透, 所以我们今天来介绍的是web区, 先从新手开始吧!

实验环境:

火狐浏览器

burp

攻防世界——web新手区

目录

view_source

robots

backup

cookie

disabled_button

simple_php

get_post

xff_referer

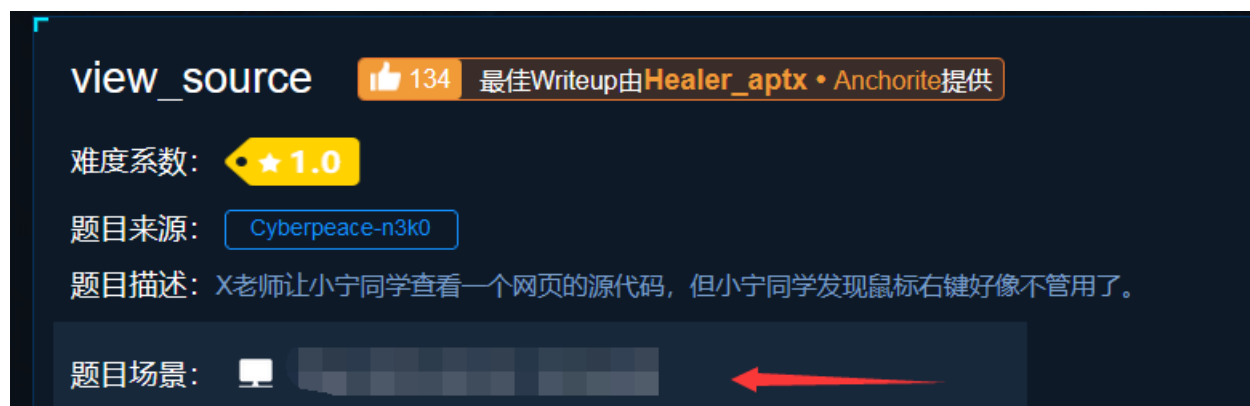
webshell

command_execution


simple_js

第一题 view_source

点击获取在线场景后等它加载完毕后点击URL进入实验环境:

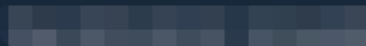



view_source  134 最佳Writeup由 [Healer_aptx](#) • [Anchorite](#) 提供

难度系数:  1.0

题目来源: [Cyberpeace-n3k0](#)

题目描述: X老师让小宁同学查看一个网页的源代码, 但小宁同学发现鼠标右键好像不管用了。

题目场景:  

删除场景

倒计时: 03:59:38 延时

题目附件: 暂无

https://blog.csdn.net/qq_45766004

1.进入环境后先由题意知道了在该页面中鼠标右击是被锁定了,所以我们右击鼠标是没有任何作用的,所以我们大胆推测flag会不会就在“检查元素”哪里呢,所以我们抱着侥幸心理看看管理员是否将F12也给禁掉了,但巧的是管理员有可能有点忙他并没有禁止掉F12,我们进入F12后很成功的看见了被注释了的flag

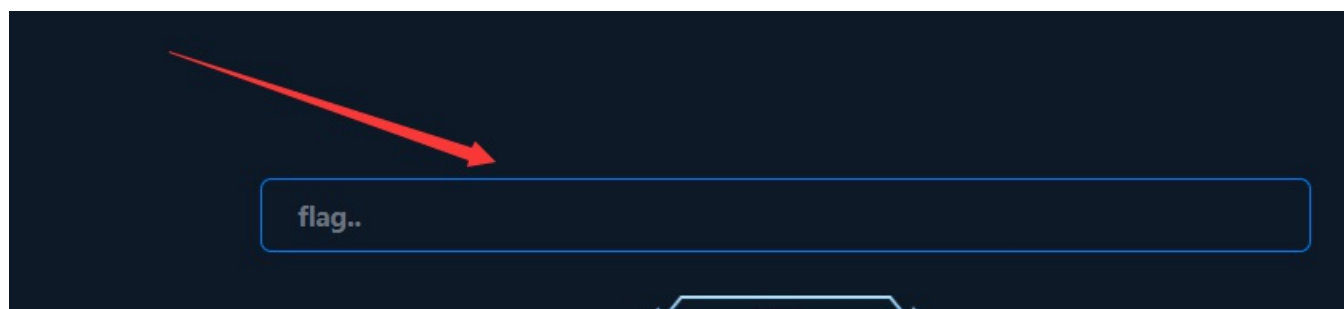
FLAG is not here



```
<!DOCTYPE html>
<html lang="en">
  <head>
  </head>
  <body>
    <script>
      <h1>FLAG is not here</h1>
      <!--cyberpeace{cfd325aa3c0c8c8f8626e0ffe85ce23}-->
    </body>
  </html>
```

https://blog.csdn.net/qq_45766004

2.接下来我们只需要双击flag那串字符后复制并粘贴到题目中的提交位置后点击提交即可



提交

https://blog.csdn.net/qq_45766004

view_source

👍 134 最佳Writeup由Healer_aptx • Anchorite提供

难度系数: ★ 1.0

题目来源: Cyberpeace-n3k0

题目描述: X老师让小宁同学查看一个网页的源代码, 但小宁同学发现鼠标右键好像不管用了。

题目场景:



删除场景

倒计时: 03:53:39

延时

题目附件: 暂无

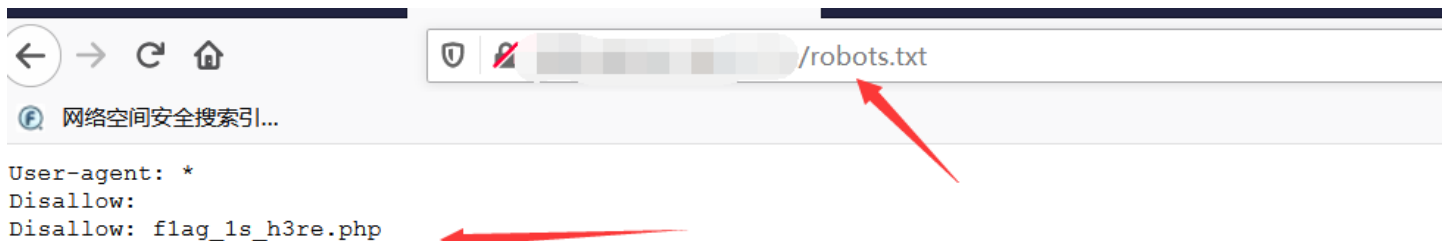
题目已答对

https://blog.csdn.net/qq_45766004

第二题robots

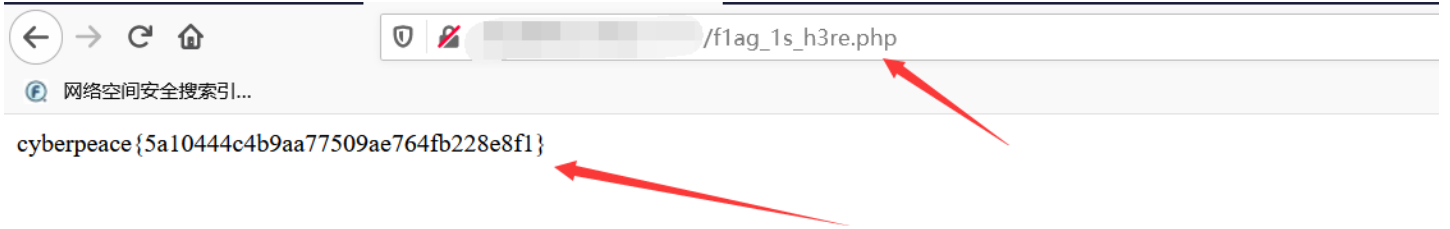
在做这道题之前我们先了解一个东西, 它叫robots协议 (又叫伪君子协议), 它是用于网站中的, 为了防止网站一些敏感目录被爬虫爬取, 所以特地建了一个文本文档用来表明那些目录是攻击者不能爬取的 (注: 非法爬取它人网站数据属于违反行为)

1. 进入实验环境后, 因为我们知道了robots是个协议, 并且存放在网站服务器上, 所以我们只需直接在URL后面输入/robots.txt即可看到当前网站都有那些页面



https://blog.csdn.net/qq_45766004

2. 我们发现有一个被标注“不能访问”的php文件在哪里, 接下来我们将刚文件复制后将URL中的robots.txt替换成flag_ls_h3re.php即可查看该文件内容



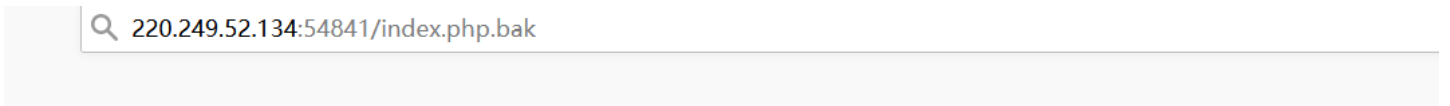
https://blog.csdn.net/qq_45766004

3.最后将flag提交即可

第三题 backup

这道题由题意可以知道，要求是将备份文件给找到并且下载下来，而大多数的管理员为了以后方便都会将备份文件的后缀写成.bak，所以，我们这里就是找到.bak的文件

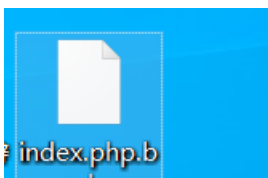
1.这里可以看见，是找到index.php的备份文件，所以我们先在url栏中输入index.php.bak试试能不能找到：



备份文

https://blog.csdn.net/qq_45766004

2.可以看见的是，这里出现了个下载框，我们再次大胆猜测，flag会不会就在这个下载文件中，现在我们点击下载：





3.可以看见，这里已经将备份文件下载下来了，但我们并不能打开该文件，因为这里最后的.bak将前面的.php给覆盖了，所以计算机自动认为这里的文件名及后缀叫：index.bak，所以我们需要将.bak给去掉：

```
index.php - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
<html>
<head>
  <meta charset="UTF-8">
  <title>备份文件</title>
  <link href="http://libs.baidu.com/bootstrap/3.0.3/css/bootstrap.min.css" rel="stylesheet" />
  <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_45766004

这样就找到了flag了

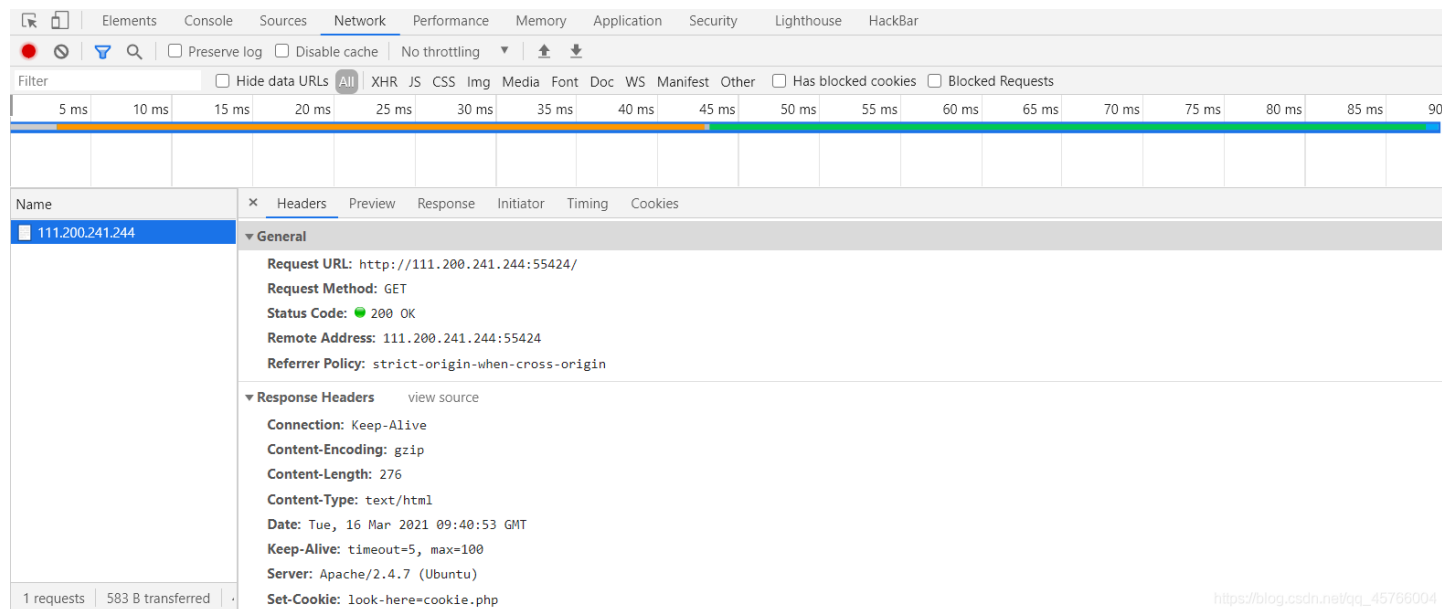
第四题 cookie

这道题主要考察用户对cookie的认知

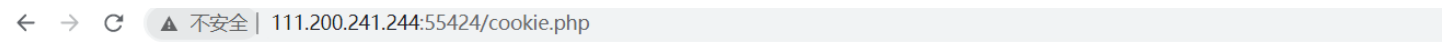
这道题有两种做法，一种是抓包后看返回包的包头，另一种是直接在浏览器上查看，我现在用的是第二种（稍微麻烦了一点）现在进入环境

你知道什么是cookie吗?

这是我们进入环境后看见的界面，感觉没有什么用，因为整个界面只有一段话，而这段话已经在题目中告诉了我们，所以我再四处找找看有没有什么有利用价值的信息



最终，被我们发现在该网页的包头中有一个名为cookie.php的网页，现在我们进行访问，看看该网页有什么作用

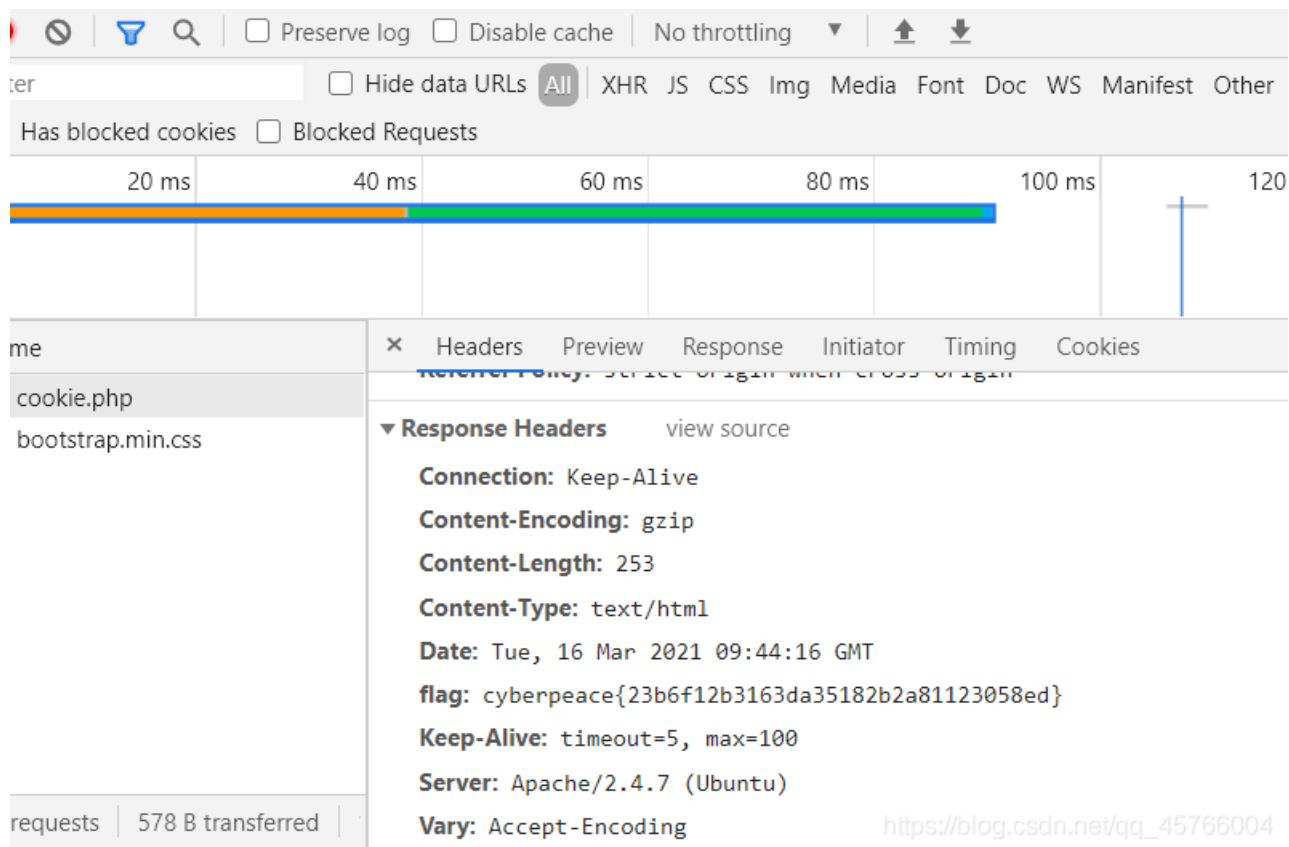


See the http response

这是我们进入后的样子，看起来没有什么作用，但突然想起，cookie它在网络中是作为包传送的，那么有没有可能这个界面的包中就存在着我们需要的信息呢，现在我们来试试

See the http response



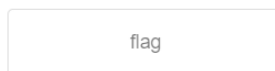


果然，我们在它的包中找到了flag，这道题就是这样的解放，当然抓包的那个解法也很简单，但需要各位自己做一下，不懂的可以留言，我看见后会答复的，现在进入下一道题把

第五题 disabled_button

这道题给的题目说的是不能按下的按钮，进入环境后是这样的

一个不能按的按钮



https://blog.csdn.net/qq_45766004

可以发现，这里有个flag的按钮框，但点击后却没有按钮的效果，所以可以判断只要将该按钮设置成活按钮就能够出flag，既然这样就又可以判断出，这里是使用代码的方式来修改该按钮的属性，现在我们来看一下该按钮的源代码

```
1 <html>
2 <head>
3   <meta charset="UTF-8">
4   <title>一个不能按的按钮</title>
5   <link href="http://libs.baidu.com/bootstrap/3.0.3/css/bootstrap.min.css" rel="stylesheet" />
6   <style>
7     body{
8       margin-left:auto;
9       margin-right:auto;
10      margin-top:200px;
11      width:20em;
```

```

2 |     }
3 |     </style>
4 | </head>
5 | <body>
6 | <h3>一个不能按的按钮</h3>
7 |
8 | <form action="" method="post" >
9 | <input disabled class="btn btn-default" style="height:50px;width:200px;" type="submit" value="flag" name="auth" />
10 | </form>
11 |
12 | </body>
13 | </html>
14 |

```

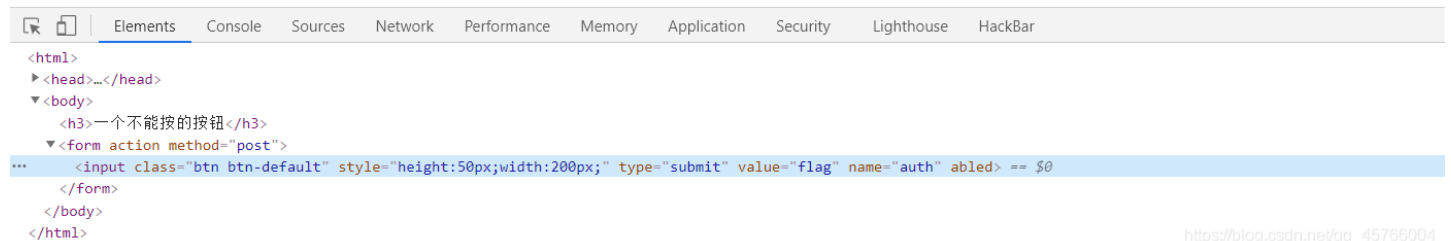
https://blog.csdn.net/qq_45766004

这里可以看见，该按钮是用input标签中的submit属性所做的提交按钮，但是呢在这里有一个特别刺眼的地方，那就是有个地方使用了否定前缀使得一个本来是活的元素变成了死的元素

```
<input disabled class="btn btn-default" ;
```

那么到这里相信大家大概知道怎么做了吧，在英语中dis是否定前缀，那么我们将否定前缀给去掉，那这个按钮又有什么变化呢，来看看

一个不能按的按钮



https://blog.csdn.net/qq_45766004

这样是不是感觉没什么变化？那我们再试试点击一下是否会有变化

一个不能按的按钮

cyberpeace{56d995070a87544d1954c8d04079b1c9}

https://blog.csdn.net/qq_45766004

哎呀，一不小心flag就被点出来了，鹅鹅鹅，这道题也有另一种解法，不过还是一样需要各位自行摸索

第六题 weak_auth

这道题呢考验的是各位对hurn这个工具的熟悉程度。讲入环境后是这样的界面

Login

https://blog.csdn.net/qq_45766004

我们试着什么都不输入的状态点击一下login，看看回显中有什么有利用价值的信息没有

111.200.241.244:35868/check.php

111.200.241.244:35868 显示
please login as admin

https://blog.csdn.net/qq_45766004

在这里可以明显的看出这里的登录用户名是admin，既然得到了登录名，那么这道题就有可能考的是用burp对密码进行爆破

Burp Project 测试器 重发器 窗口 帮助

仪表盘 目标 代理 测试器 重发器 定序器 编码器 对比器 插件扩展 项目选项 用户选项

1 × 2 × ...

目标 位置 有效载荷 选项

有效载荷位置

设置在基本请求中插入有效载荷的位置。攻击类型指定如何将有效载荷分配给有效载荷位置。 - 有关详细信息，请参阅帮助。

攻击类型: **狙击手 (Sniper)**

```
POST /check.php HTTP/1.1
Host: 111.200.241.244:35868
Content-Length: 27
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://111.200.241.244:35868
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://111.200.241.244:35868/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close

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

https://blog.csdn.net/qq_45766004

有效载荷集 开始攻击

您可以定义一个或多个有效载荷集。有效载荷集的数量取决于“位置”选项卡中定义的攻击类型。每个有效载荷集可以使用各种有效载荷类型，并且可以以各种方式定制每种有效载荷类型。

有效载荷集: 1 有效载荷数量: 7,501
 有效载荷类型: 简单词典 请求数量: 7,501

有效载荷选项[简单列表]

设置用于有效内容的简单字符串列表。

- ↳ admin
- admin12
- admin888
- admin8
- admin123
- sysadmin
- adminxxx

输入新项目

https://blog.csdn.net/qq_45766004

有效载荷处理

结果	目标	位置	有效载荷	选项		
过滤器: 显示所有项目						
请求	有效载荷	状态	错误	超时	长	评论
31	123456	200	<input type="checkbox"/>	<input type="checkbox"/>	437	
0		200	<input type="checkbox"/>	<input type="checkbox"/>	434	
1	↳ admin	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
2	admin12	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
4	admin8	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
5	admin123	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
3	admin888	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
6	sysadmin	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
7	adminxxx	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
9	6kadmin	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
11	feitium	200	<input type="checkbox"/>	<input type="checkbox"/>	434	

请求 响应

Raw 头 Hex HTML Render

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>weak auth</title>
</head>
<body>

cyberpeace{838f0c02ab292937a2c47874fc8054e3}<!--maybe you need a dictionary-->

</body>
</html>

```

https://blog.csdn.net/qq_45766004

ok, 最后的答案已经出来了

第七题 simple_php

这道题是一个php的代码审计题，进入环境后我们只会看见一个php代码

```

<?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();
}

```

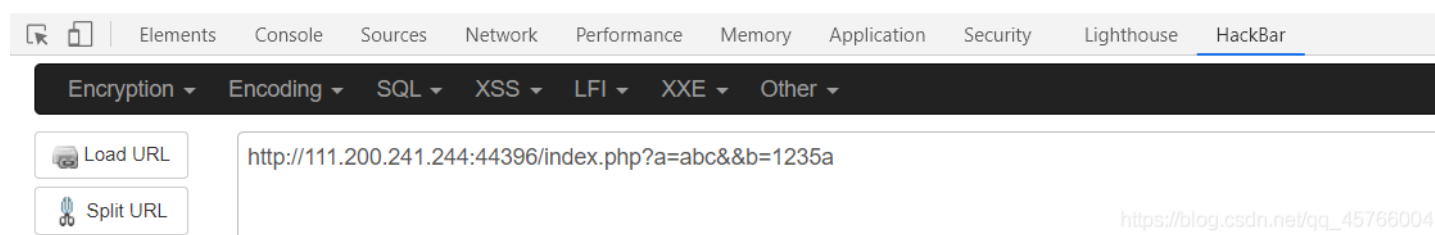
```
}
if($b>1234){
    echo $flag2;
}
?>
```

https://blog.csdn.net/qq_45766004

这里相信有一点点程序基础的人都看得懂，就是两个变量和三个if判断，但是真正起作用的却只有两个，好了，后面的做法我也不做解释了，直接上图

```
<?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;
}
?>
```

Cyberpeace{647E37C7627CC3E4019EC69324F66C7C}



这道题就解出来了，鹅鹅鹅

第八题 get_post

这道题考察的是用户对post和get协议的熟悉程度，这道题有两种解体方法，一种是利用hackbar，另一个种就是使用burp，我们这道题采用的是hackbar的方式做的，现在进入环境

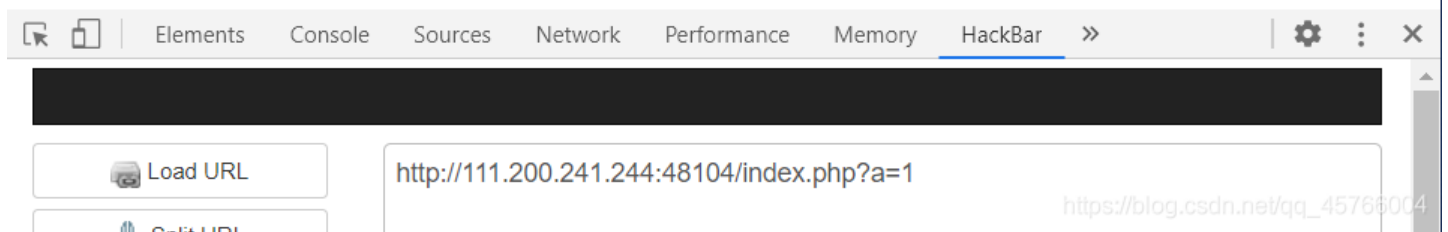
请用GET方式提交一个名为a,值为1的变量

https://blog.csdn.net/qq_45766004

现在我们使用hackbar用get方式传输a变量为1的值

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

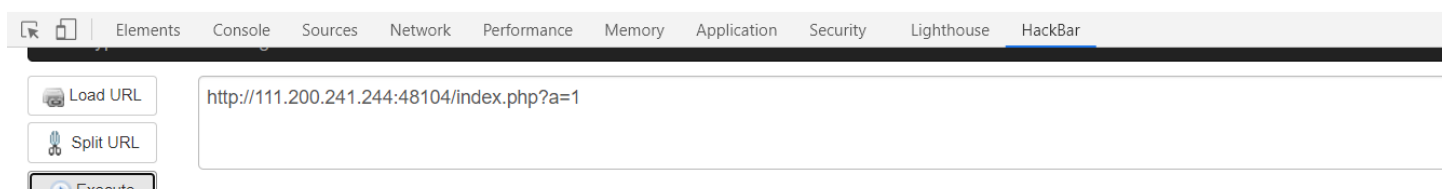


可以看见第一个条件满足后第二个条件就自己出来了，现在我们再用post的方式传输b变量为2的值

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{bf3d44c50994c9ed01154f9cbc35734f}





Post data Referer User Agent Cookies [Clear All](#)

b=2

https://blog.csdn.net/tqq_45766004

这样就完成了