xss-labs部分题目

原创

抒情诗、 于 2020-04-14 10:47:31 发布 269 收藏

文章标签： xss

本文链接：https://blog.csdn.net/zhangxiansheng12/article/details/105506902

---

xss-labs

level-1

第一关是搜索到的教程，因为啥都不明白。

然后就是没有搜索框，就直接使用get的方式传值，归根结底还是html。让 `name="test"` 输入的内容代替到了的是test的那一处，输入 `name="test"><script>alert(1)</script>`

然后弹框 `完成的不错`

level-2

第二关是半自己做的，在第一关中明白了一点思路。

因为存在搜索框，使用的payload就和level-1的一样，也就是 `keyword="test"><script>alert(1)</script>//`

那为什么要在这个最后加上 `//` 这个东西呢？就是因为这个可以把后面的 `">` 这个东西给他注释掉。功能很强大的样子。

还是弹框 `完成的不错`

level-3

第三关也是半看相关知识半做的，得知这一关要使用的知识是JavaScript事件相关的。在菜鸟教程学习了相关的知识。菜鸟教程-JavaScript事件

这一关对 `<``>` 这两个标签采用了转义处理，然后这个引号是单引号而非双引号，所以在框内搜索 `1' onclick=alert(1)//` 就可以了

然后就是这个 `//` 的作用还是不太清楚，似乎是要注释掉后面的标签，不知道最后为啥就还可以执行这个事件反正是做出来了，还不错。

level-4

自己做的也是稀里糊涂的，用法和上一关的用法一样，就是这次的单引号换成了双引号，我正准备再尝试一次呢，就弹出完成的不错了。

输入框中的内容 `1" onclick=alert(1)//`

ps:看到try harder！还以为很难。

level-5

使用 `"> <script>alert('yes')</script>` 一般语句尝试，过滤了script标签

使用 `find a way out!" onclick=alert(1)//` 过滤掉了onclick标签

以上两个标签的过滤，都是对大小写都进行了过滤。

看了别人写的博客，发现还能使用 `<a></a>`，这个标签。

然后尝试使用伪链接方式假造一个超链接：

`"> <a href="javascript:alert('test')">hahaha</a>`，点击自己创建的链接，完成的不错。

level-6

和第一关相似的payload，然后把`script`中的`s`改为大写的`S`,对小写的整体进行了过滤，但是未对大写的部分进行过滤

payload为`break it out!"><script>alert(1)</script>//`

完成的不错。

level-7

第七关因为很多关键的词语都被屏蔽了，但是`onclick`中的`on`被直接替换为空，所以可以采用`单词嵌套`来绕过屏蔽。

payload为:`1" oonnclick=alert(1)//`

完成的不错！

level-8

这里有一个添加友情链接的功能，然后试一下用`javascript:alert(1)`然后就变成了`javas_cript:alert(1)`，又试了一下大小写，同样也被过滤掉了。

最后是使用了Unicode编码了一次，才能`绕过`过滤。

点击友情链接，完成的不错！

level-9

这里还是有一个添加友情链接的功能，然后试一下用`javascript:alert(1)`然后就不合法，很多关键词都被过滤掉了。

后台的源码应该与下面的`代码`实现的`功能`类似:

```
<!DOCTYPE html><!--STATUS OK--><html>
<head>
<meta http-equiv="content-type" content="text/html;charset=utf-8">
<script>
window.alert = function()
{
confirm("完成的不错！");
 window.location.href="level10.php?keyword=well done!";
}
</script>
<title>欢迎来到level9</title>
</head>
<body>
<h1 align=center>欢迎来到level9</h1>
<?php
ini_set("display_errors", 0);
$str = strtolower($_GET["keyword"]);
$str2=str_replace("script","scr_ipt",$str);
$str3=str_replace("on","o_n",$str2);
$str4=str_replace("src","sr_c",$str3);
$str5=str_replace("data","da_ta",$str4);
$str6=str_replace("href","hr_ef",$str5);
$str7=str_replace('"','&quot',$str6);
echo '<center>
<form action=level9.php method=GET>
<input name=keyword  value="'.htmlspecialchars($str).'">
<input type=submit name=submit value=添加友情链接 />
</form>
</center>';
?>
<?php
if(false===strpos($str7,'http://'))
{
  echo '<center><BR><a href="您的链接不合法？有没有！">友情链接</a></center>';
        }
else
{
  echo '<center><BR><a href="'.$str7.'">友情链接</a></center>';
}
?>
<center><img src=level9.png></center>
<?php
echo "<h3 align=center>payload的长度:".strlen($str7)."</h3>";
?>
</body>
</html>
```

发现要有`http://`这个关键词，就把他放在`//`的后面就行了。

因为对关键词进行了过滤，所以就算payload是`javascript:alert(1)//http://`也是不行的，可以`r`和`i`之间放入一个`%09`就是Tab制表符实现过滤。但是我并不知道具体的过滤的实现过程。

%09要放在url内实现。

点击链接，完成的不错

level-10

查看元素后发现，`page`里面有一些，准确来说是3个`隐藏`的`<input>`的标签

分别对这3个参数进行GET传参：`http://web-labs.rinue.top/xss-labs/level10.php?keyword=well%20done!&t_link=test1&t_history=test2&t_sort=test3`,查看元素的代码为

```
<input name="t_link" value="" type="hidden">
<input name="t_history" value="" type="hidden">
<input name="t_sort" value="test3" type="hidden">
```

由此可见，可对`t_sort`这个参数进行传值。

下一步就是弹框了，payload为`?t_sort=test" type="text" onclick="alert(1)`其中`test" type="text" onclick="alert(1)`为传入的内容

完成的不错！

level-11

和level-10差不多，都是hiden的框框，但是模仿着10关的过程怎么都做不出来，看了别人做的。

原来是要修改`referer`，最后使用了`HackBar2.1.3`(火狐的)然后构造payload直接就行了。

payload为`" type="text" onclick="alert(1)`,点击已经显示出来的框框就行了。

完成的不错！

level-12

和第11关的差不多，模仿着第11关，一做就出来了。

这一关修改的是`user-agent`,因为下面的`t_ua`的值就是`user-agent`的值。

`payload`不用改，还是11关的`payload`就行。

完成的不错！

level-13

和上两关的差不多，不过这一关所要修改的是cookies的值。

`payload`稍有不同，用的是前两关用到的`payload`之外，又多加了一部分。因为cookies的结构与其他的有些不一样。

payload为`user=" type="text" onclick="alert(1)`,点击显示出来的框框。

完成的不错！

level-14

不会，先跳过，会了再回来。

level-15

根据writeup，这里的注入点为`src`，

level-16

源码为：

```
<!DOCTYPE html><!--STATUS OK--><html>
<head>
<meta http-equiv="content-type" content="text/html;charset=utf-8">
<script>
window.alert = function()
{
confirm("完成的不错！");
 window.location.href="level17.php?arg01=a&arg02=b";
}
</script>
<title>欢迎来到level16</title>
</head>
<body>
<h1 align=center>欢迎来到level16</h1>
<?php
ini_set("display_errors", 0);
$str = strtolower($_GET["keyword"]);
$str2=str_replace("script"," ",$str);  /过滤'script'
$str3=str_replace(" "," ",$str2); //过滤'空格'
$str4=str_replace("/"," ",$str3);   //过滤'/'
$str5=str_replace(" "," ",$str4);
echo "<center>".$str5."</center>";
?>
<center><img src=level16.png></center>
<?php
echo "<h3 align=center>payload的长度:".strlen($str5)."</h3>";
?>
</body>
</html>
```

过滤了`script`，`空格`，`/`。使用换行符`%0A`取代空格，在html中照样可以运行。

payload为:`<img%0asrc="1.jpg"%0aonerror="alert(1)">`,这里的`1.jpg`是不存在的，所以触发了`onerror`事件，导致弹窗。

完成的不错！

level-17

`<embed src="xsf01.swf?a=b" heigth="100%" width="100%">`这一段是关键的代码。而url中`arg01=a&arg02=b`说明`arg01`和`arg02`分别取代的是`a=b`中的前面(a)和后面(b)。

所以构造的payload为`arg01=a&arg02=b onmouseover=alert(1)`，就行了，开始忙活了半天没有实现成功。原来是我的火狐浏览器的问题！

使用`Microsoft edge`的时候才能成功，因为`xsf01.swf`这个文件火狐里打不开，但是`Microsoft edge`打得开

完成得不错！

level-18

跟上一关相似，一开始空格被转化成了`%20`，然后把`%20`改成%0A就可以了,还是火狐不可以，但是`Microsoft edge`可以

payload为`arg01=a&arg02=b%0Aonmouseover=alert(1)`。

完成的不错！进入下一关！

level-19

最后两关还没想出来。

似乎跟17，18差不多。

推荐我的博客