Google XSS game writeup

转载

weixin_30300523 于 2015-11-18 13:50:00 发布 42 收藏
文章标签： javascript 游戏 ViewUI
原文链接：http://www.cnblogs.com/renzongxian/p/4974308.html

用过Chrome的应该知道它的XSS Auditor，它可是灭掉了不少XSS代码呢……Google对XSS是很有研究的，不然也不敢大张旗鼓的悬赏（7500刀哦亲），还开发了一个XSS小游戏 http://xss-game.appspot.com/ （需FQ……），一共有6关，做完有“大蛋糕”奖励哦！下面就来看一下。

Level 1: Hello, world of XSS

既然是“Hello world”级别的，想必就是简简单单弹个框在输入框里输入`<script>alert(1)</script>`，然后点击按钮提交，弹框了，轻松愉快。

Level 2: Persistence is key

这个类似评论框，`<script>`被过滤了，试试插入图片标签`<img src=# onerror=alert(1) />`，提交后成功弹窗了，而且是存储型XSS，每次查看都会弹框

Level 3: That sinking feeling…

没有输入框，有三张图片可以切换，可以看到图片切换跟URL中`#`之后的数字有关系，查看源代码之后发现`<script>`标签中有这么一句`html += "<img src='/static/level3/cloud" + num + ".jpg' />";`，很明显是决定图片地址的，而num来自于`#`后的输入，我们尝试注入，注入的时候要闭合语句，不要弄出语法错误，我的语句是`1.jpg' onload=alert(1) '1`，这样就变成`<img src='/static/level3/cloud1.jpg' onload=alert(1) '1.jpg' ?>`，这样，访问http://xss-game.appspot.com/level3/frame#1.jpg‘ onload=alert(1) ‘1，即可弹框

Level 4: Context matters

有输入框，是个定时器程序，查看timer.html的源代码发现`<img src="/static/loading.gif" onload="startTimer('{{ timer }}');" />`，其中的`timer`就是输入框的输入，尝试注入并注意闭合语句，在输入框中输入`1'),alert(1) ('`便可弹框。

Level 5: Breaking protocol

这个虽然也有输入框，但是提交后发生跳转，而且跳转到的网页跟刚才输入的东西好像没啥关系……查看signup.html源代码发现，是真的没关系……不过发现这么一句`<a href="{{ next }}">Next >></a>`，Next跳转后的地址是由URL中的变量`next`决定的，于是我便想在此注入`signup" onmouseover=alert(1) "`，结果双引号被转换成了`&quot`，对confirm.html页面的注入依旧无效，页面跳来跳去的就是不弹框……无奈之下看了看提示，想起对标签还有`href="javascript:alert(1)"`这种执行js的方式啊，把next赋值成`javascript:alert(1)`，刷新该网页，然后点击“NEXT”，弹框！

Level 6: Follow the ?

这个的意思就是要加载一个外部的js代码来弹框，外部代码的地址为URL中`#`后面的内容。外部js代码可以使用XSS平台获得（搜一下就能找到），但是通过查看源码我们发现它对http(s)进行了检查过滤，相关代码如下

```
if (url.match(/^https?:\/\//)) {
 setInnerText(document.getElementById("log"),
         "Sorry, cannot load a URL containing \"http\".");
       return;
     }
```

其中的变量url就是URL中#之后的内容，仔细观察正则式我们发现它是从开始进行匹配的，如果我们在#之后加个空格，再加上我们要加载的远程js全地址，会怎么样呢？成功绕过了该检查，弹框！

之后，便出现祝贺的蛋糕啦，啊哈！

转载于:https://www.cnblogs.com/renzongxian/p/4974308.html