跨站脚本攻击基础

讲师：空白

学院介绍

学院宗旨：专注网安人才实战技能培养

学院官网：https://edu.hetianlab.com/

合天网安实验室：https://www.hetianlab.com/

主打课程：

《web安全》：OWASP TOP 10漏洞原理及测试

《渗透测试》：渗透测试流程及工具的使用

《安全开发》：用python写一个综合的扫描器

《CTF-PWN》：CTF中的PWN相关

目录
CONTENTS

/01 认识跨站脚本攻击

1.1 跨站脚本攻击概念

跨站脚本（Cross-site scripting，简称XSS）,是指恶意攻击者往Web页面里插入恶意JavaScript代码，当用户浏览该页之时，嵌入其中Web页面的JavaScript代码会被执行，从而达到恶意攻击用户的目的。

1.2 漏洞成因

由于动态网页的Web应用对用户提交请求参数未做充分的检查过滤，允许用户在提交的数据中掺入HTML代码（最主要的是“>”、 “<”），然后未加编码地输出到第三方用户的浏览器，这些攻击者恶意提交代码（payload）会被受害用户的浏览器解释执行。

1.3 Payload概念

Payload 的中文含义是有效载荷，在 XSS 中指代攻击代码或攻击语句。

常见的 Payload 有：

正常弹窗

<script>alert(1)</script>

<img src=0 onerror=alert(1)>

弹出网站 Cookie

<script>alert(document.cookie)</script>

<img src=0 onerror=alert(document.cookie)>

1.4 形成的危害

获取用户信息；（如浏览器信息、ip 地址、cookie 信息等）

钓鱼；（利用 xss 漏洞构造出一个登录框，骗取用户账户密码，提示登录过期，模拟一个网站的登录框，将用户名、密码发送到攻击者服务器）

注入木马或广告链接；（有些在主站注入非法网站的链接，对公司的声誉有一定的影响）

后台增删改网站数据等操作；（配合 CSRF 漏洞，骗取用户点击，利用 js 模拟浏览器发包）

xss 蠕虫（微博蠕虫：只要看过某人的微博就是自动关注某人；贴吧蠕虫：看过某个帖子就自动回复）

/02 跨站脚本攻击类型及利用场景

2.1 常见分类

跨站脚本攻击可以分为三类，反射型、存储型以及DOM型。那么这三类有哪些区别呢?

反射型：Payload 经过后端，不经过数据库

存储型：Payload 经过后端，经过数据库

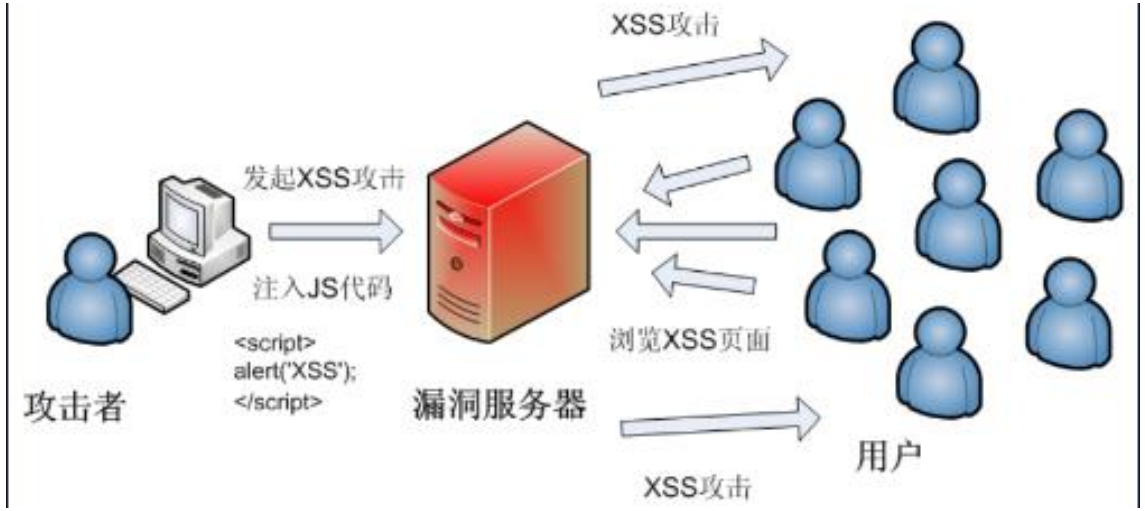
DOM：Payload 不经过后端

2.1.1 反射型XSS

也叫做非持久型XSS。攻击者将恶意脚本附加到 url 的参数中，发送给受害者，服务端未经严格过滤处理而输出在用户浏览器中，导致浏览器执行代码数据。

特点：只执行一次

常见漏洞场景：搜索处



2.1.1 代码分析

```
1  <html>
2  <head>
3  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4  <title>反射型XSS</title>
5  </head>
6  <body>
7  <form action="" method="get">
8  <input type="text" name="input">
9  <input type="submit">
10 </form>
11 <br>
12 <?php
13 $xss_r = $_GET['input'];
14 echo 'output:<br>'.$xss_r;
15 ?>
16 </body>
17 </html>
```

2.1.2 存储型XSS

也叫做持久型XSS。攻击者在数据中嵌入代码，这样当其他用户请求后，服务器从数据库中查询数据并发给用户，用户浏览此类页面时就可能受到攻击。

常见漏洞场景：多见于评论留言，个人信息等处

2.1.2 代码分析

```php
<?php
header('X-XSS-Protection: 0');
?>
<p>存储型 XSS 演示</p>
<meta http-equiv="Content-Type" content="text/html;charset=utf-8"/>
<form action="" method="post">
    <input type="text" name="xss"/>
    <input type="submit" value="test"/>
</form>
<?php
$xss=@$_POST['xss'];
mysql_connect("localhost","root","root");
mysql_select_db("xss");
if($xss!==null){
    $sql="insert into test (id,payload) values ('1','$xss')";
    $result=mysql_query($sql);
    echo $result;
}
?>
```

```php
<?php
Zoom: 180%ect("localhost","root","root");
mysql_select_db("xss");
$sql="select payload from test where id=1";
$result=mysql_query($sql);
while($row=mysql_fetch_array($result)){
    echo $row['payload'];
}
?>
```

2.1.3 DOM型XSS

基于DOM的XSS，通过对具体DOM代码进行分析，根据实际情况构造dom节点进行XSS跨站脚本攻击。

2.1.3 代码分析

#<img src=x onerror='alert(/xss/)'>。

1.html [D:\Software\PHPStudy\PHPTutorial\WWW\xss_d] - Notepad3

文件(F) 编辑(E) 查看(V) 外观(P) 设置(S) 帮助(H)

```
1  <html>
2  <head>
3          <title>DOM XSS</title>
4          <meta charset="utf-8">
5  </head>
6  <body>
7          <div id="area"></div>
8          <script>
9                  document.getElementById("area").innerHTML = unescape(location.hash);
       //location.hash函数要从uri中的#
10         </script>
11 </body>
12 </html>
```

2.2 Self XSS

顾名思义，自己输入xss脚本，输出仅自己看到，仅xss到自己。

2.2.1 代码分析

'><'

```
1  <html>
2  <head>
3  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4  <title>DOM型XSS</title>
5  <script>
6      function test(){
7          var str = document.getElementById("text").value;
8          document.getElementById("t").innerHTML = "<a href='"+str+"' >testLink</a>";
9      }
10 </script>
11 </head>
12 <body>
13 <div id="t"></div>
14 <input type="text" id="text" value="" />
15 <input type="button" id="s" value="write" onclick="test()" />
16 </body>
17 </html>
```

感谢您的聆听

学习工具、资料及课程回放

扫码免费领取