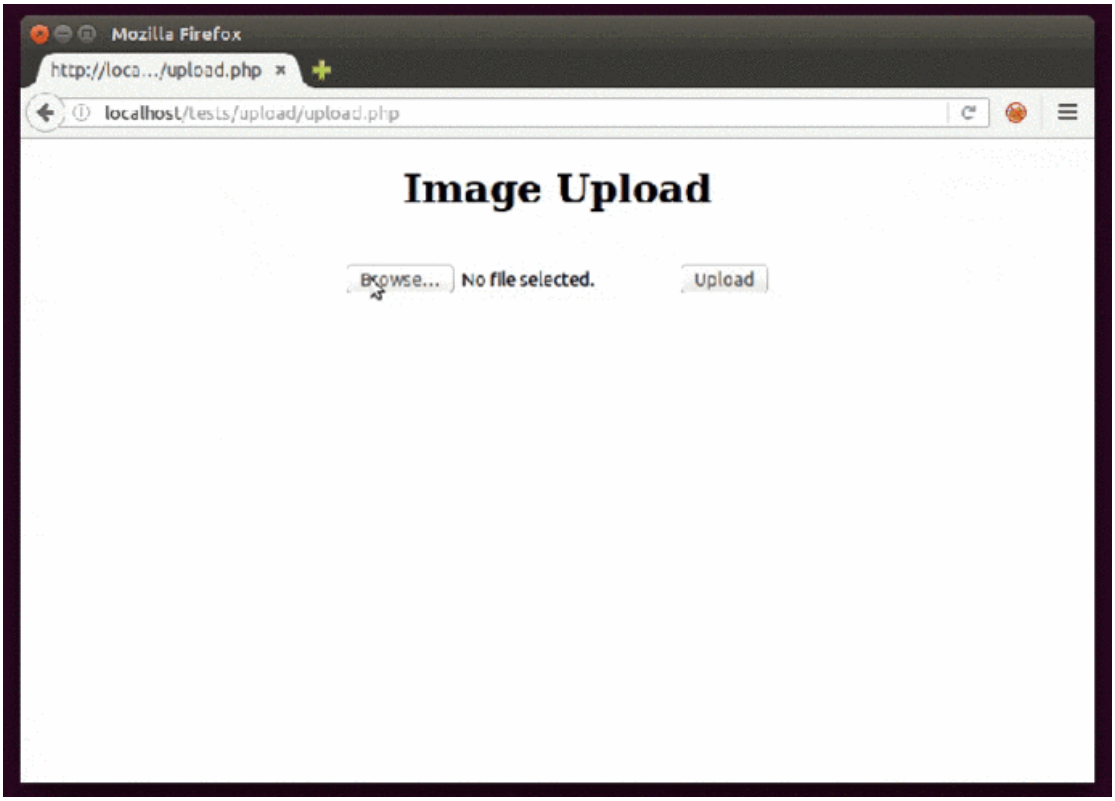
如何利用文件上传执行 XSS？

利用文件上传实现 XSS 攻击是一个 Hacking Web 应用的很好机会，特别是无处不在的用户头像上传案例中，这就给予我们很多发现开发者错误的机会。基本的文件上传 XSS 攻击有以下几种。

1、 文件名

文件名本身可能就是网页的一部分可以造成反射，所以可以通过将 XSS 语句插入文件名中来触发反射。



尽管不是有意为之，但是这个 XSS 可以实践在 W3Schools。

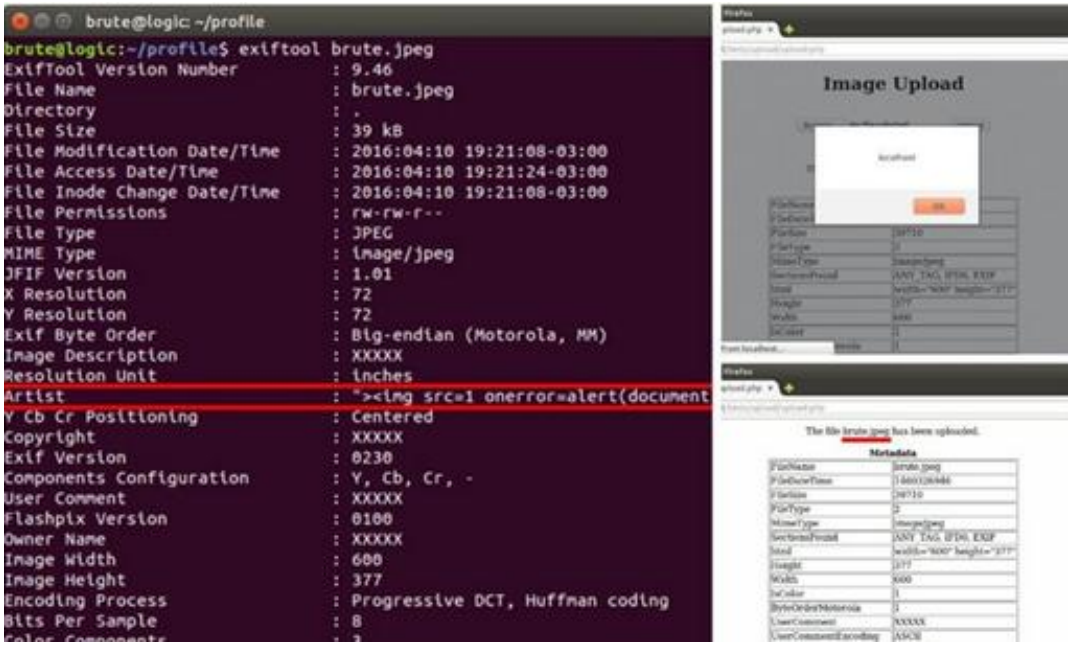
2、元数据

使用 exiftool 工具可以修改 EXIF 元数据，从而在某些地方造成反射：

```
$ exiftool -FIELD=XSS FILE
```

例子：

```
$ exiftool -Artist=’ “><img src=1
onerror=alert(document.domain)>’ brute.jpeg
```



3、 内容

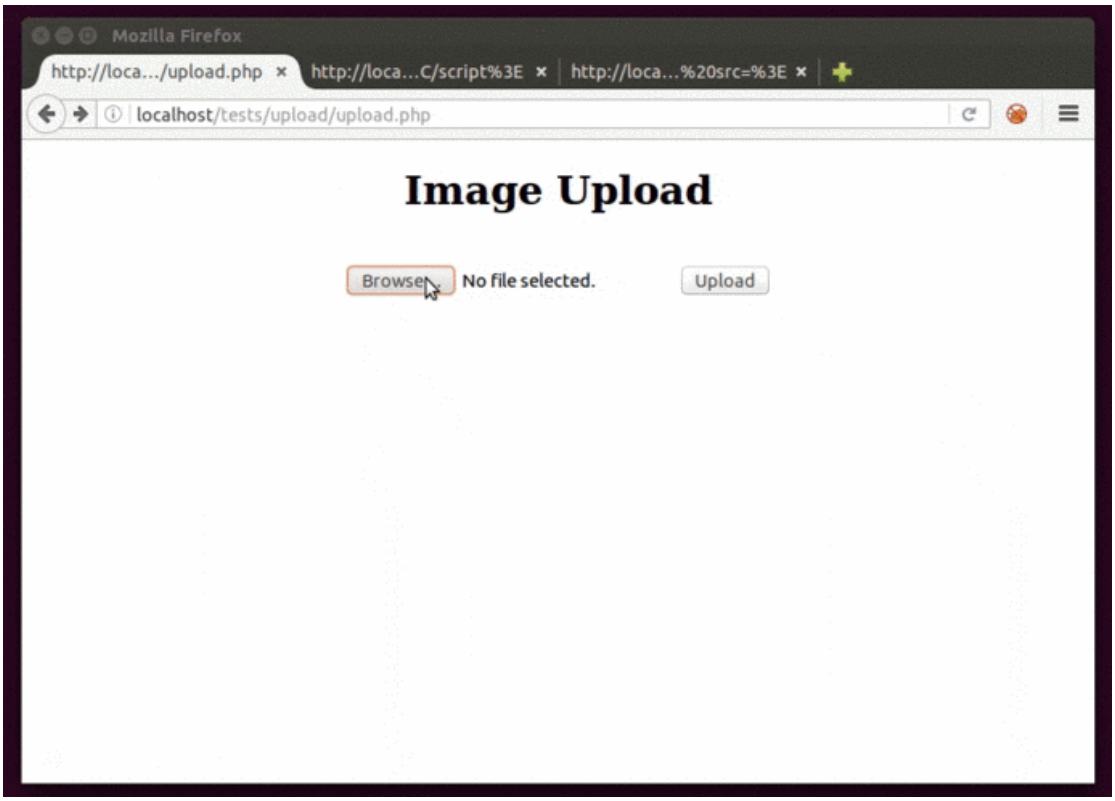
如果 Web 应用允许上传 SVG（一种图像类型）扩展名，则以下内容可以用来触发 XSS：

```
<svg xmlns="http://www.w3.org/2000/svg"
onload="alert(document.domain)"/>
```

一个 POC 可以在这里看到 brutelogic.com.br/poc.svg。

4、源码

我们可以很容易的创建一张包含 javascript payload 的 GIF 图片，然后将这张图片当做源码加以引用。如果我们可以成功的注入相同的域名，如下所示，则这样可以有效的帮我们绕过 CSP（内容安全策略）防护（其不允许执行例如 `<script>alert(1)</script>`）



创建这样一张图片可以使用如下内容并将文件命名为 .gif 后缀：

```
GIF89a/*<svg/onload=alert(1)>*/=alert(document.domain)//;
```

GIF 文件标识 GIF89a 做为一个 javascript 的变量分配给 alert 函数。中间注释部分的 XSS 是为了以防图像被检索为 text/HTML MIME 类型时，通过请求文件来执行 payload。

我们通过下图可以发现，类 UNIX 命令的 PHP 函数 exif_imagetype() 和 getimagesize() 都会将这个文件识别为 GIF 文件。而一般的 Web 应用都是使用这些函数来验证图像类型的，所以这样一个文件是可以被上传的（但上传后可能会被杀毒软件查杀）。

更多可以用作 javascript 变量赋值的 ASCII 文件类型标识符，请参见这里。

更多关于文件上传 XSS 的详尽示例，包括绕过像 GD 库等过滤的例子，请参见这里。