文件上传的危害

一、 文件上传危害介绍

在web应用中，每个人都会接触上传功能最为常见。例如，用户可利用上传功能上传个人照片或图片，从而自定义头像。在这个过程中，用户会上传自己的信息（通常是文件，部分情况为远程图片地址），服务器接收到用户端的上传信息后会按照业务流程进行处理，并在后续页面中显示。

上传功能是用户与服务器进行文件交互的重要手段，主要应用于系统的流程核心点，如证件信息上传，申请表格上传，自定义头像上传。通过上传功能，用户可实现自有内容的个性化修改、重点业务的开展等。同时也可通过文件上传功能实现用户交互的自定义设计，为业务开展及用户体验提供良好的实现方式。

但是，上传过程中存在重大安全隐患。攻击者的目标是取得当前Web服务器的权限。如果通过Web层面开展攻击，那么必须将攻击者的木马插人Web系统中，并在服务器端执行。这个过程就是对Web服务器进行文件注人攻击。这时，上传点可作为上传木马的有效途径，上传攻击将直接威胁当前系统的安全性。上传攻击可定义如下:

文件上传攻击是指攻击者利用Web应用对上传文件过滤不严的漏洞，将应用程序定义类型范围之外的文件上传到Web服务器，并且此类文件通常为木马，在上传成功后攻击者即可获得当前的webshell。

二、 文件上传攻击原理

在针对Web的攻击中，攻击者想要取得webshell，最直接的方式就是将Web木马插人服务器端并进行成功解析。那么如何理解成功解析?假设目标服务器为用PHP语言构建的Web系统，那么针对上传点就需要利用PHP木马，并且要求木马在服务器以后缀名 为.php进行保存。因此，上传木马的过程就是在Web系统中新增一个页面。 当木马上传成功后，攻击者就可远程访问这个木马文件，也就相当于浏览一个页面，只不过这个页面就是木马，具备读取、修改文件内容、连接数据库等功能。

了解木马的原理之后再进一一步思考，服务器肯定不能允许这种情况存在。因此，Web应用在开发时会对用户上传的文件进行过滤，如限制文件名或内容等。因此，上传漏洞存在的前提是:存在上传点且上传点用户可独立控制上传内容，同时上传文件可被顺利解析。在以上条件都具备的情况下，攻击者方可利用此漏洞远程部署木马，并获取服务器的Web执行权限，进而导致服务器的webshell被获取，并产生后续的严重危害。

总结来说，假设目标Web服务器为Apache+PHP架构，攻击者通过上传功能上传“木马php”到服务器，再访问“木马.php”所在的目录，由此“木马php”会被当作php文件执行，进而木马生效。

三、 文件上传攻击的条件

回到攻击者视角，攻击者利用上传功能的目标是将Web木马上传至服务器并能成功执行。因此，攻击者成功实施文件上传攻击并获得服务器webshell的前提条件如下：

1. **目标网站具有上传功能：**上传攻击实现的前提是：目标网站具有上传功能，可以上传文件，并且文件上传到服务器后可被存储。
2. **上传目标文件能够被Web服务器解析执行**：由于上传文件需要依靠中间件解析执行，因此上传文件后缀应为可执行格式。在Apache+PHP环境下，要求上传的Web木马采用`.php`后缀名（或能有以PHP方式解析的后缀名），丙炔存在上传文件的目标要有执行脚本的权限。以上两种条件缺一不可。
3. **需要知道文件上传到服务器后存在路径和文件名称：**许多Web应用都会修改上传文件的文件名称，这时就需要结合其它漏洞获取这些信息。如果不知道上传文件的存放路径和文件名称，即使上传成功也无法访问。因此如果上传成功但不知道真实路径，那么攻击过程没用任何意义。

4. **目标文件可别用户访问：**如果文件上传后，却不能通过web访问，或真实路径无法获得，木马则无法被攻击者打开，那么就不能成功实施攻击。

总结：以上是上传攻击成功的4个必要条件。

四、 文件上传攻击绕过

在正常情况下， 服务器会对上传文件的格式或内容进行校验，几乎不可能允许直接上传文件。攻击者需要对不同的防御方式进行绕过，才能上传文件。常见的上传绕过方法如下：

- 前端绕过：文件上传前端验证的原理，被上传的文件还没有传到服务器端，而是在前端进行拦截的。
- 服务端绕过：文件类型、文件头、文件后缀。
- 配合文件包含漏洞绕过。
- 配合服务器解析漏洞绕过。
- CMS、编辑器漏洞绕过。
- 配合操作系统文件命名规则绕过。

五、 webshell-介绍

简介：简单来说一句话木马就是通过向服务端提交一句简短的代码来达到向服务器插入木马并最终获得webshell的方法。对于不同的语言有不同的构造方法，基本构造是首先出现的是脚本开始的标记，后边跟着的 eval 或者是 execute 是核心部分，就是获取并执行后边得到的内容，而后边得到的内容，是 request 或者是 $_POST 获取的值。如果我们通过客户端向服务器发送，那么就会让服务器执行我们发送的脚本，挂马就实现了。

```
php一句话木马：   <?php @eval($_POST[value]); ?>
asp一句话木马：   <%eval request ("value")%> 或  <% execute(request("value")) %>
aspx一句话木马：  <%@ Page Language="Jscript" %> <% eval(Request.Item["value"]) %>

<?php fputs( fopen('xie.php','w') , '<? php eval($_POST[xie]) ?>' ) ; ?>
将当前目录下创建xie.php文件，并且将一句话木马写入xd.php中
```

一句话小马原理：

拿php的一句话木马说明一下原理：在PHP脚本语言中，eval(code) 的功能是将 code 组合成 php 指令，然后将指令执行，其他语言中也是使用此原理，只是函数可能不同。`<?php $a="phpinfo()"; eval("$a;");?>  #就相当于执行 phpinfo（）; 语句。`

当利用web中的漏洞将`<?php @eval($_POST[value]);?>`一句话插入到了可以被黑客访问且能被web服务器执行的文件中时，那么我们就可以向此文件提交post数据，post方式提交数据的参数就是这个一句话中的`value`，它就称为一句话木马的密码。这样提交的数据如果是正确的php语言的语句，那么就可以被一句话木马执行，从而达到黑客的恶意目的。