# 文件上传漏洞概述

**什么是上传**

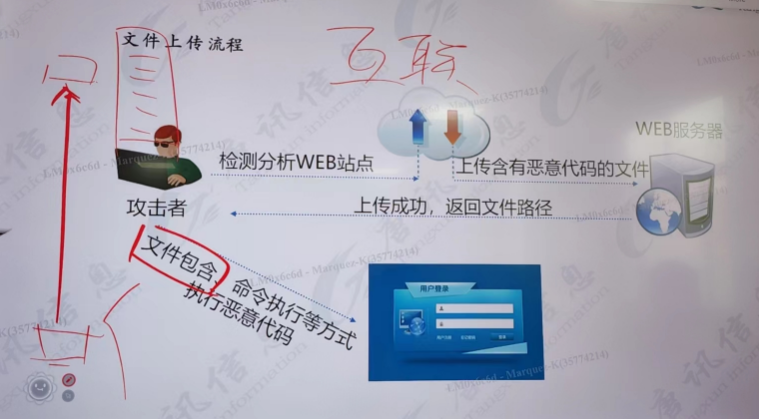
* 上传指用户提交文档到web服务器平台，是web系统中最常见的功能之一
* 文件上传功能本身没有问题，问题是文件被上传到哪里；上传之后，服务器如何处理、解释文件

**文件上传漏洞产生的原因**

文件上传漏洞的成因，一方面服务器配置不当会导致任意文件上传;另一方面，Web 应用开放了文件上传功能，并且对上传的文件没有进行足够的限制;再者就是，程序开发部署时候，有考虑到系统特性和验证和过滤不严格而导致限制被绕过，上传任意文件。

* Web服务器的文件上传功能在程序设计上逻辑缺陷。
* Web服务器无法识别区分上传的文件内容、格式
* Web服务器对文件上传路径或位置控制不严格
* 服务器对所上传文件的读、写、执行、继承权限设计不严格。

**文件上传流程**



**文件上传漏洞的危害**

上传漏洞最直接的威胁就是上传任意文件，包括恶意脚本、程序等。如果Web 服务器所保存上传文件的可写目录具有执行权限，那么就可以直接上传后门文件，导致网站沦陷。如果攻击者通过其他漏洞进行提权操作，拿到系统管理权限，那么直接导致服务器沦陷。同服务器下的其他网站无一幸免，均会被攻击者控制。

通过上传漏洞获得的网站后门，就是Webshell。上传漏洞与SQL注入或XSS相比，其风险更大，如果 Web应用程序存在上传漏洞，攻击者上传的文件是Web脚本语言，服务器的Web容器解释并执行了用户上传的脚本，导致代码执行。如果上传的文件是Flash的策略文件crossdomain.xml，黑客用以控制Flash在该域下的行为。

如果上传的文件是病毒、木马文件，黑客用以诱骗用户或者管理员下载执行。如果上传的文件是钓鱼图片或为包含了脚本的图片，在某些版本的浏览器中会被作为脚本执行，被用于钓鱼和欺诈甚至攻击者可以直接上传一个webshell到服务器上完全控制系统或致使系统瘫痪。

**webshell概述**

Webshell 就是以asp、php、jsp或者cgi（common gateway interface，通用网关接口）等网页文件形式存在的一种命令执行环境，也可以将其称做为一种网页后门。

拆分开来看，“web”的含义显然是服务器开放的web服务，“shelI”的含义是取得对服务器某种程度上操作权限。在计算机科学中，Shell 俗称壳 (用来区别于“核”) ，是指“为使用者提供操作界面”的软件 (命令解释器)。

由于webshell其大多是以动态脚本的形式出现，也有人称之为网站的后门工具，也是一个命令解释器，不过是以Web 方式(HTTP 协议) 通信 (传递命令消息) ，继承了Web 用户的权限。WebShell本质上是在服务器端可运行的脚本文件，后缀名为.php/.asp/.aspx/.jsp 等，也就是说WebShell 接收来自于Web 用户的命令，然后再服务器端执行。

# webshell&木马

**一句话木马**

ASP: <%eval request("cmd")%>

ASP.NET：<%@ Page Language="Jscript"%>

<%eval(Request.Item["cmd"],"unsafe");%>

PHP: <?php @eval(S\_REQUEST[‘cmd’);?>

JSP: <% out.println(system(request.getParameter("cmd")));%>

# WebShel防范

**webshell防御**

关于文件上传的防御，防住危险的脚本类型是最基本的防御，最理想的是能够过滤掉图片马中的恶意代码。如果一个Web 应用能够上传图片木马，那么我们认为这个Web 应用是不安全的。文件上传漏洞的防御主要从以下几个方面考虑。

\*代码角度

* 采用白名单策略，严格限制上传文件的后缀名
* 进行二次渲染，过滤掉图片马中的恶意代码。0
* 上传文件重命名，尽量少的从客户端获取信息
* 避免文件包含漏洞。
* 严格处理文件路径，防御00.截断漏洞，避开空格、点、:SDATA等windows 特性

\*服务器角度

* 及时更新WWeb 容器，防止解析漏洞的产生。
* 可写目录不给执行权限。

# webshell工具

# 文件上传漏洞绕过

# 文件上传漏洞防范