严格来说，文件包含漏洞是“代码注入”的一种。其原理就是注入一段用户能控制的脚本或代码，并让服务端执行。“代码注入”的典型代表就是文件包含。文件包含漏洞可能出现在 JSP、PHP、ASP 等语言中，原理都是一样的，本文只介绍 PHP 文件包含漏洞。

要想成功利用文件包含漏洞进行攻击，需要满足以下两个条件：

1.1 Web 应用采用 include()等文件包含函数通过动态变量的方式引入需要包含的文件;

1.2 用户能够控制该动态变量。

在 PHP 中，有四个用于包含文件的函数，当使用这些函数包含文件时，文件中包含的 PHP 代码会被执行。下面对它们之间的区别进行解释：

include():当使用该函数包含文件时，只有代码执行到 include()函数时才将文件包含进来，发生错误时只给出一个警告，继续向下执行。

include_once():功能和 include()相同，区别在于当重复调用同一文件时，程序只调用一次。

require():1.require()与 include()的区别在于 require()执行如果发生错误，函数会输出错误信息，并终止脚本的运行。2.使用 require()函数包含文件时，只要程序一执行，立即调用文件，而 include()只有程序执行到该函数时才调用。

require_once():它的功能与 require()相同，区别在于当重复调用同一文件时，程序只调用一次。

A **file inclusion vulnerability** is a type of vulnerability that is most commonly found to affect web applications that rely on a scripting run time. This issue is caused when an application builds a path to executable code using an attacker-controlled variable in a way that allows the attacker to control which file is executed at run time. A file include vulnerability is distinct from a generic Directory Traversal Attack, in that directory traversal is a way of gaining unauthorized file system access, and a file inclusion vulnerability subverts how an application loads code for execution. Successful exploitation of a file include vulnerability will result in remote code execution on the web server that runs the affected web application.

文件包含漏洞

Remote File Inclusion

Remote File Inclusion (RFI) occurs when the web application downloads and executes a remote file. These remote files are usually obtained in the form of an HTTP or FTP URI as a user-supplied parameter to the web application.

Local File Inclusion

Local File Inclusion (LFI) is similar to a *Remote File Inclusion* vulnerability except instead of including remote files, only local files i.e. files on the current server can be included for execution. This issue can still lead to remote code execution by including a file that contains attacker-controlled data such as the web server's access logs.