DVWA File Inclusion——Writeup

原创

b1gpig安全 于 2020-11-22 14:01:23 发布 39 收藏

分类专栏： DVWA 文章标签： 安全

本文链接：https://blog.csdn.net/weixin_45694388/article/details/109923153

文件包含：

即File Inclusion，意思是文件包含（漏洞），是指当服务器开启allow_url_include选项时，就可以通过php的某些特性函数（include()，require()和include_once()，require_once()）利用url去动态包含文件，此时如果没有对文件来源进行严格审查，就会导致任意文件读取或者任意命令执行。文件包含漏洞分为本地文件包含漏洞与远程文件包含漏洞，远程文件包含漏洞是因为开启了php配置中的allow_url_fopen选项（选项开启之后，服务器允许包含一个远程的文件）。服务器通过php的特性（函数）去包含任意文件时，由于要包含的这个文件来源过滤不严，从而可以去包含一个恶意文件，而我们可以构造这个恶意文件来达到自己的目的。

文件包含漏洞：

攻击者利用了包含的特性，再加上了应用本身对文件控制不严，对include进来的文件不可控，最终造成了攻击者进行任意文件包含。包含进来的文件都以当前脚本文件解析，比如，我们当前测试系统是Apache加php环境，那么被include进来的文件，不管是什么类型，比如说图片，文本文档，这些文件被包含以后，都会被当做php脚本来解析。

LOW

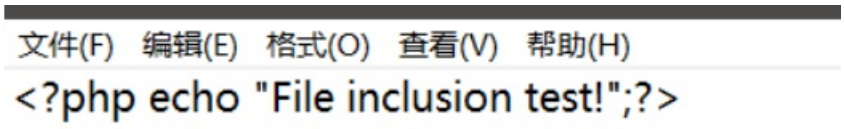
源码：

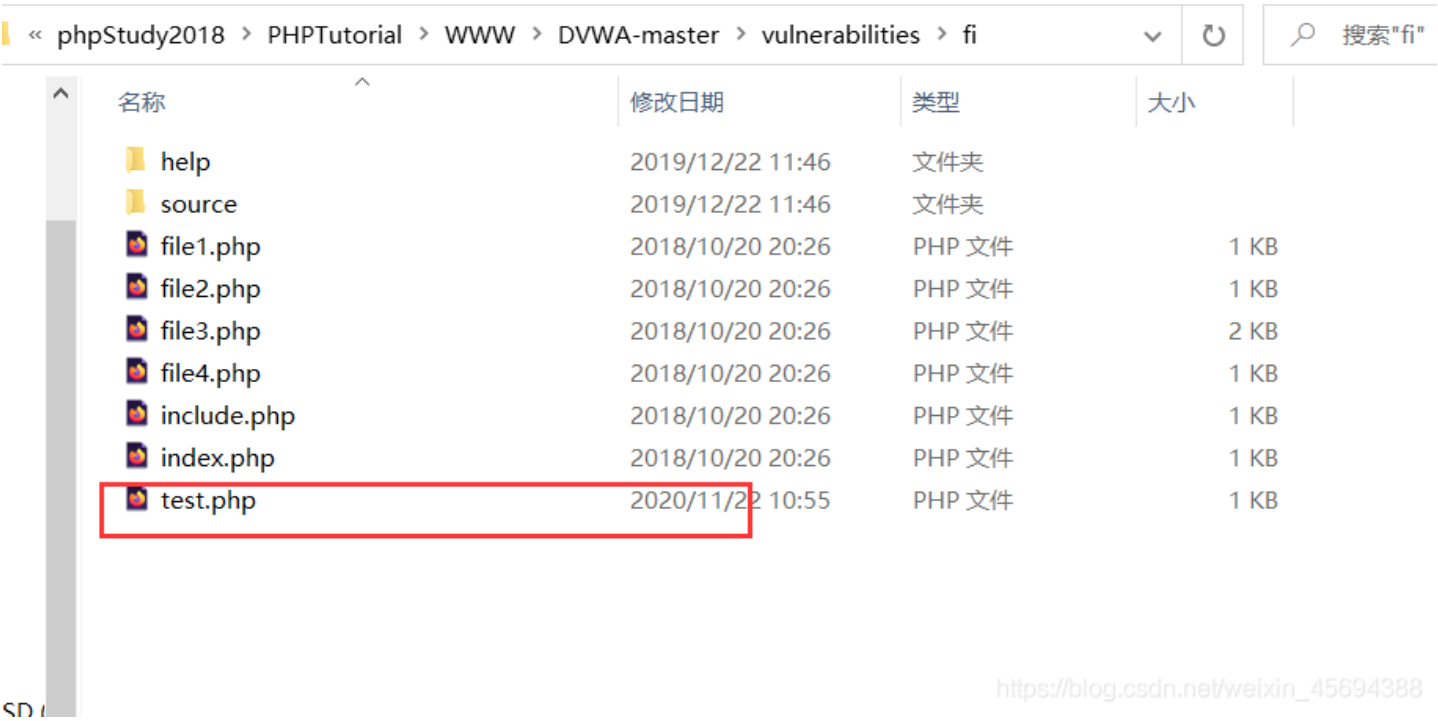
```php
<?php

// The page we wish to display
$file = $_GET[ 'page' ];

?>
```

测试：本地文件包含，在目录中加一个test.txt





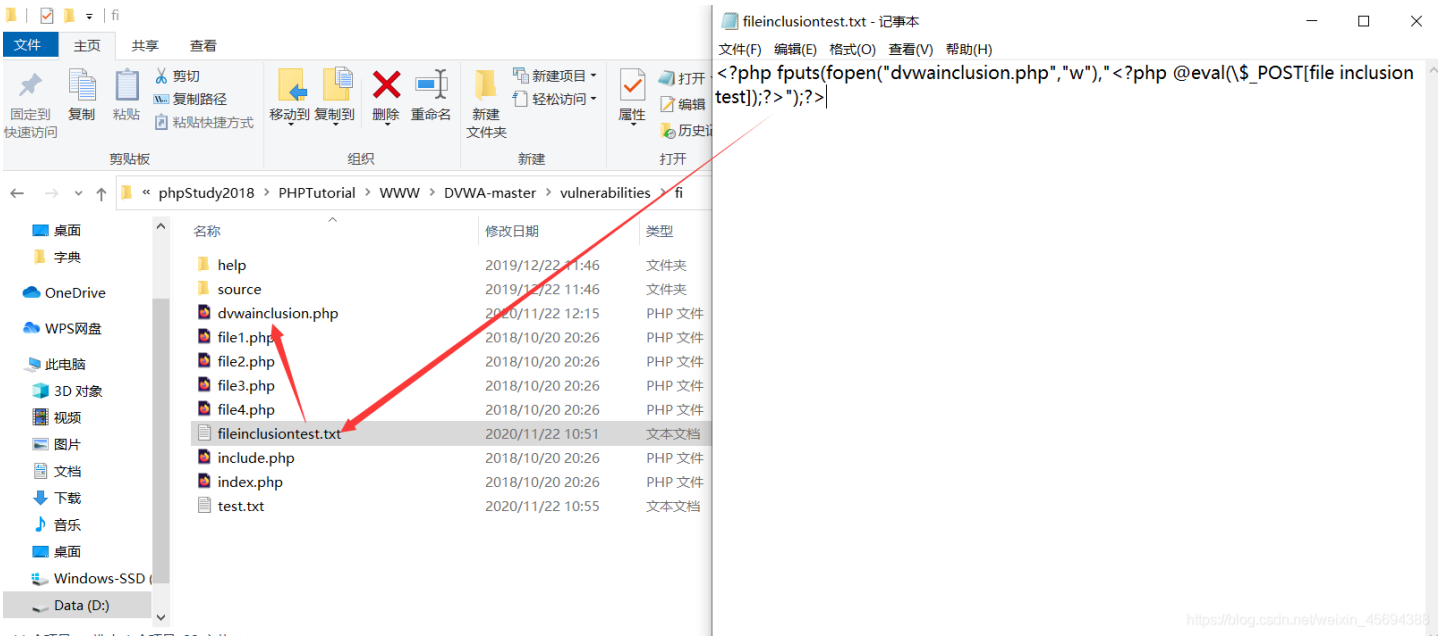
执行成功：

注意：无论文件后缀名是什么，只要文件内容为php脚本都会正确解析并且执行

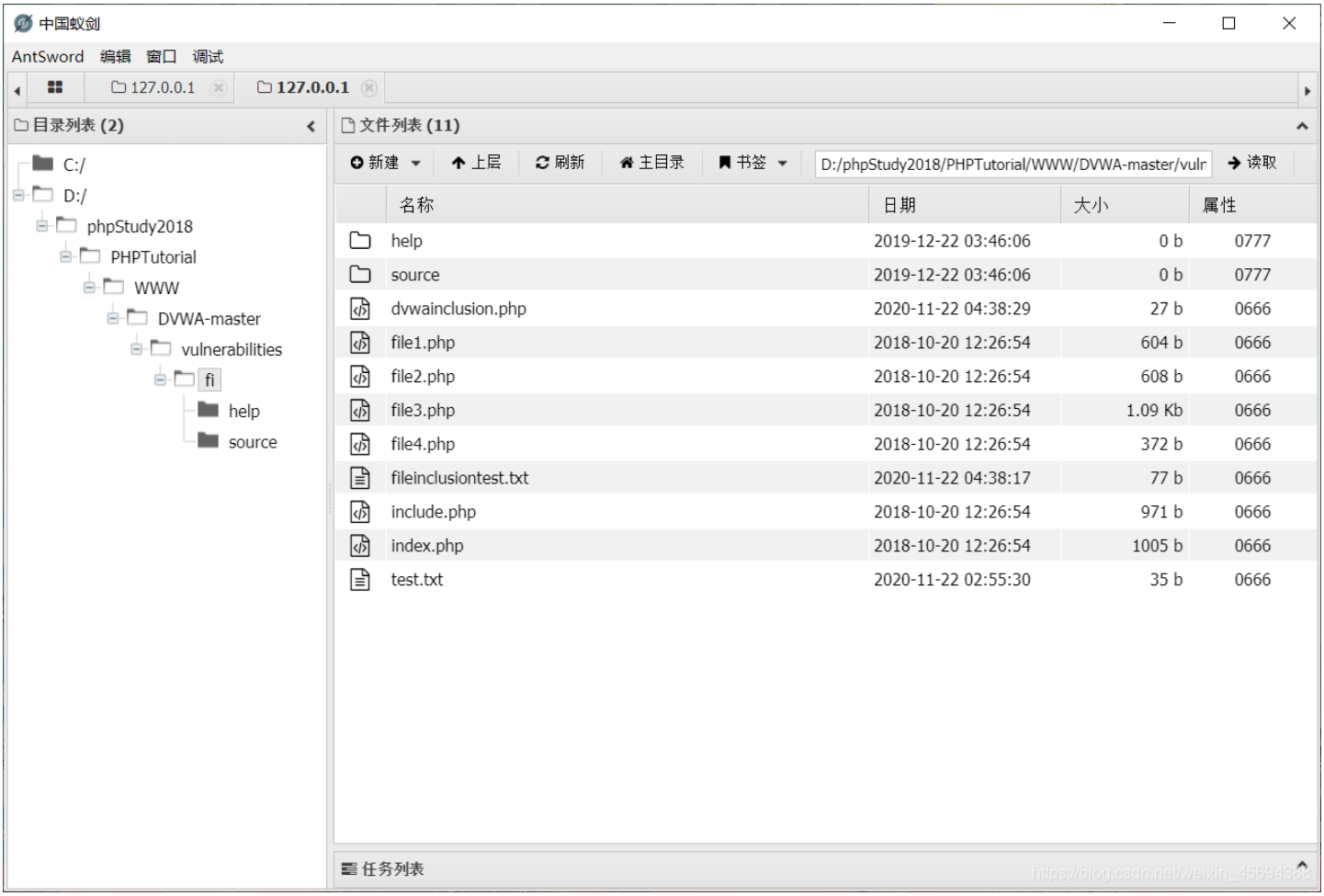
漏洞利用：

实际利用需要利用文件上传漏洞，上传木马，

fputs函数：为php函数，功能和fwrite函数一样，表示创建一个新文件，这样上传成功以后就会在当前平台文件所在目录下面创建一个含有木马的dvwainclusion.php文件



蚁剑连接，这个就很恐怖了，可以打开我电脑下所有文件



medium:

源码

```php
<?php

// The page we wish to display
$file = $_GET[ 'page' ];

// Input validation
$file = str_replace( array( "http://", "https://" ), "", $file );
$file = str_replace( array( "../", "..\"" ), "", $file );

?>
```

str_replace函数将 “http://” “https:\” “…/” “…” 替换成空

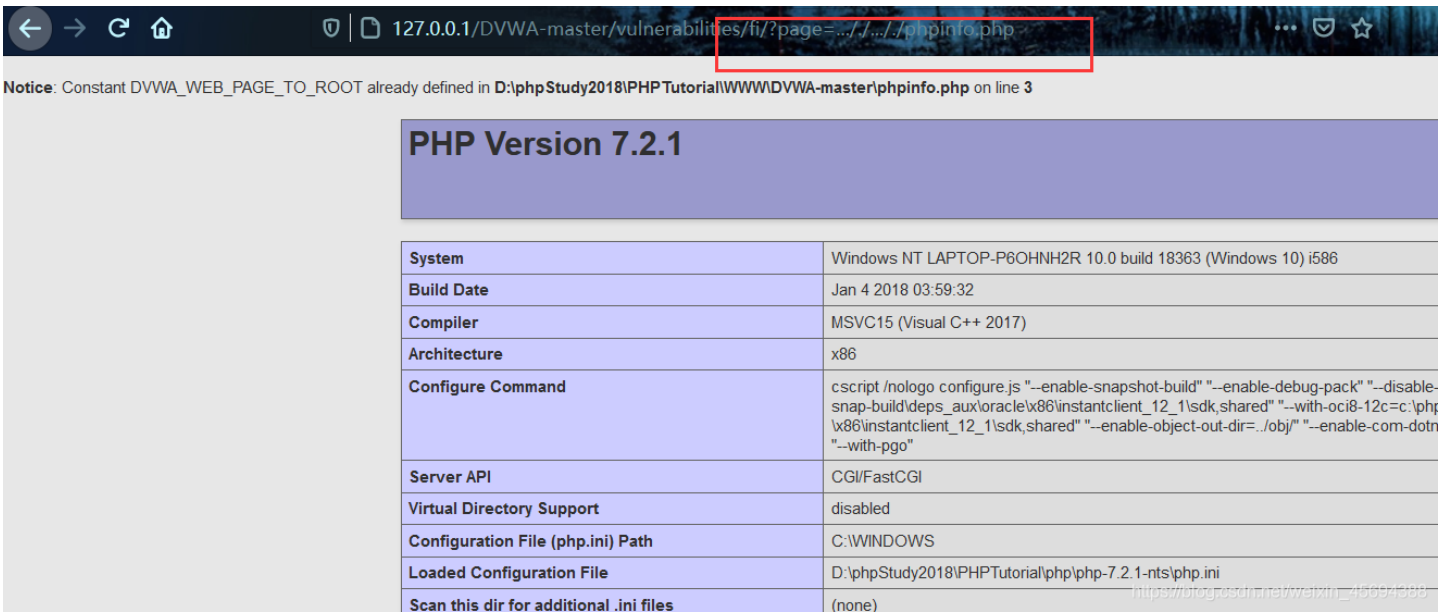
绕过：

所有 “http://” 用 “htthttp://p://” 代替

"…/“用”…/./"代替

以此类推

远程文件包含：



后面的漏洞利用就与low操作相同了

high:

```php
<?php

// The page we wish to display
$file = $_GET[ 'page' ];

// Input validation
if( !fnmatch( "file*", $file ) && $file != "include.php" ) {
        // This isn't the page we want!
        echo "ERROR: File not found!";
        exit;
}

?>
```


High级别的代码使用了fnmatch函数检查page参数，要求page参数的开头必须是file，服务器才会去包含相应的文件。

High级别的代码规定只能包含file开头的文件，看似安全，不幸的是我们依然可以利用file协议绕过防护策略。

使用file:///协议可以成功访问，其余的步骤就和前面一样了

impossible

```
<?php

// The page we wish to display
$file = $_GET[ 'page' ];

// Only allow include.php or file{1..3}.php
if( $file != "include.php" && $file != "file1.php" && $file != "file2.php" && $file != "file3.php" ) {
        // This isn't the page we want!
        echo "ERROR: File not found!";
        exit;
}

?>
```


可以看到，Impossible级别的代码使用了白名单机制进行防护，简单粗暴，page参数必须为“include.php”、“file1.php”、“file2.php”、“file3.php”之一，彻底杜绝了文件包含漏洞。

强行总结

防御从低到高依次：
无防护无过滤
str_replace函数过滤 “http://” “https:\” “…/” “…”（str_replace 函数具有危险性，比较容易绕过，例如双写）
fnmatch函数检查参数开头
白名单限制（简单有效）