

# 【代码审计】ThinkPhp6任意文件写入

原创 染尘 saulGoodman 今天

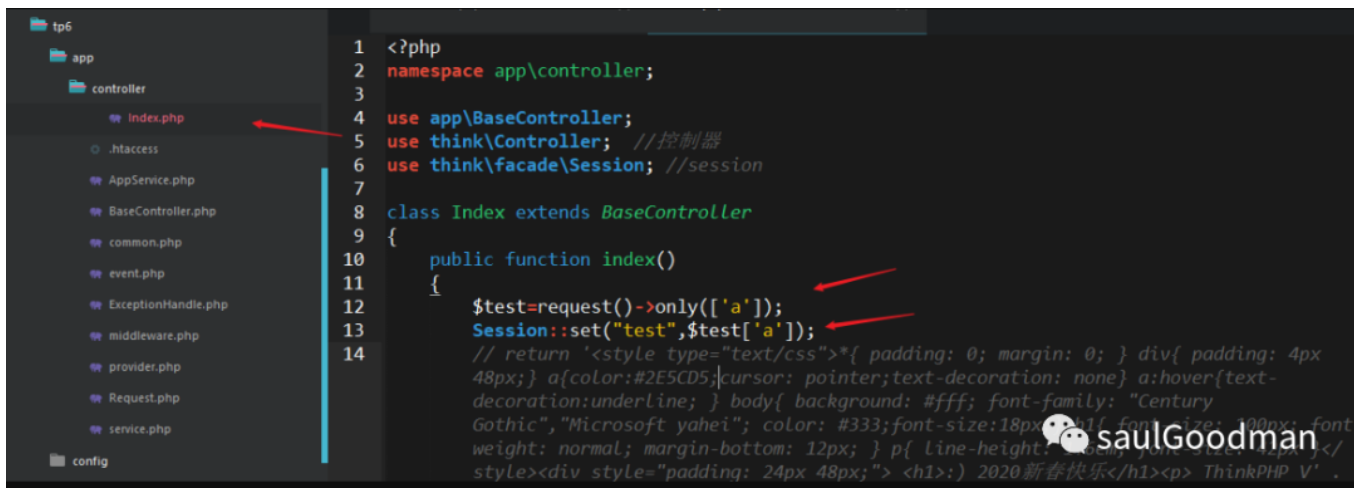
## ThinkPhp6任意文件写入

### Thinkphp6:任意文件写入

版本：v6.0.0-v6.0.1

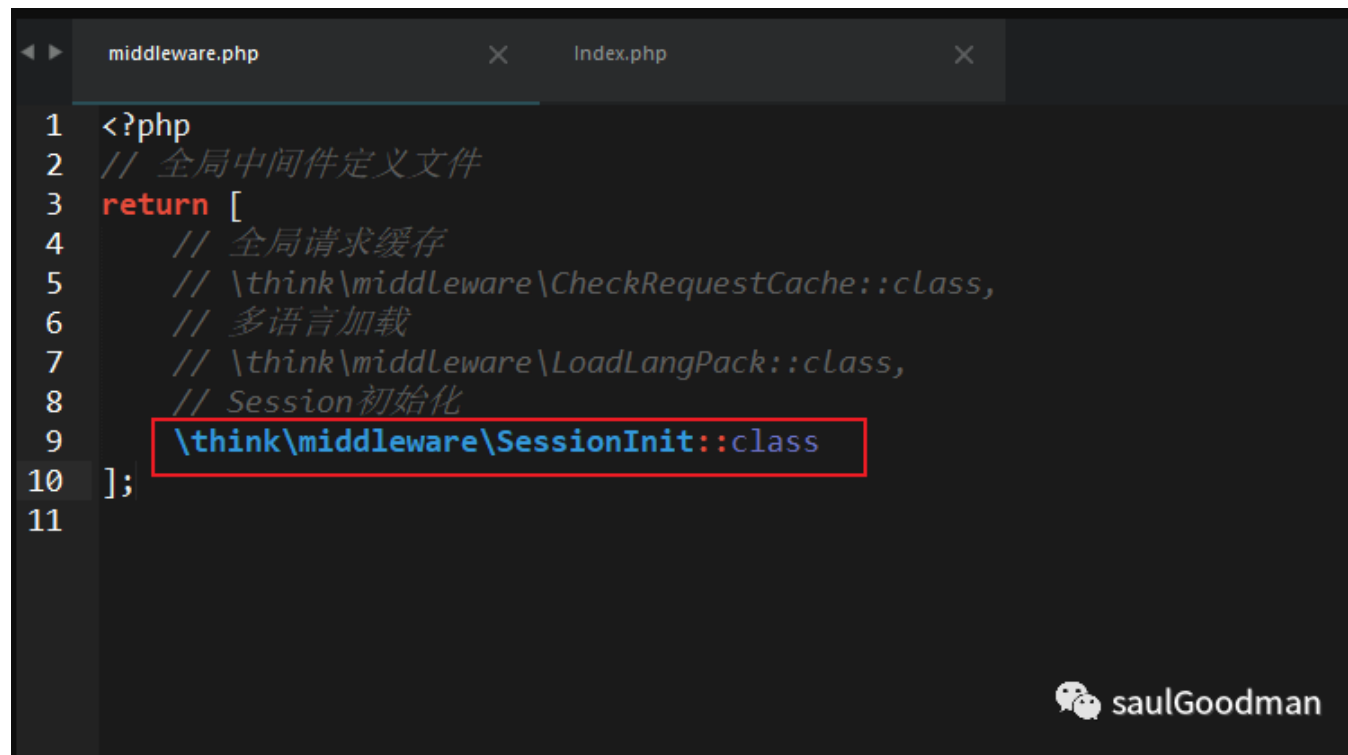
复现过程：

下载配置好tp6 然后，在\tp6\app\controller\Index.php 中。写好，漏洞代码。



```
1 <?php
2 namespace app\controller;
3
4 use app\BaseController;
5 use think\Controller; //控制器
6 use think\facade\Session; //session
7
8 class Index extends BaseController
9 {
10     public function index()
11     {
12         $test=request()->only(['a']);
13         Session::set("test",$test['a']);
14         // return 'style type="text/css">{* padding: 0; margin: 0; } div{ padding: 4px 48px;} a{color:#2E5CD5;cursor: pointer;text-decoration: none} a:hover{text-decoration:underline; } body{ background: #fff; font-family: "Century Gothic","Microsoft yahei"; color: #333;font-size:18px;}p{font-size: 100px; font-weight: normal; margin-bottom: 12px; } p{ line-height: 1.5; }</style><div style="padding: 24px 48px;"> <h1>:) 2020 新春快乐</h1><p> ThinkPHP V' .
```

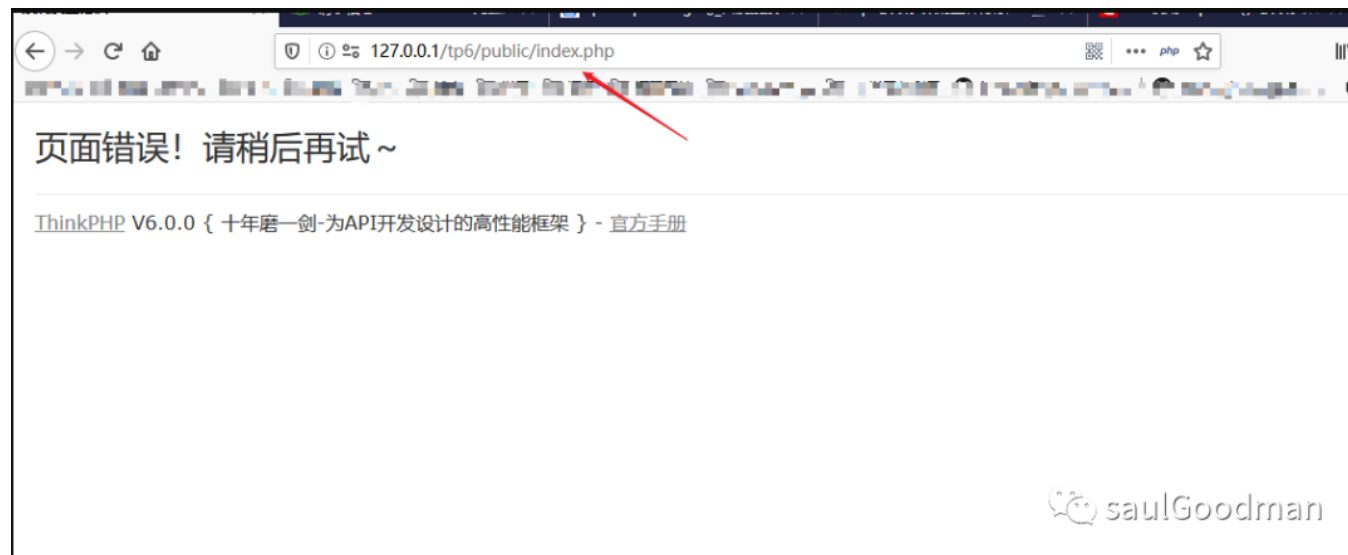
然后再去，tp6\app\middleware.php中吧session 写入打开



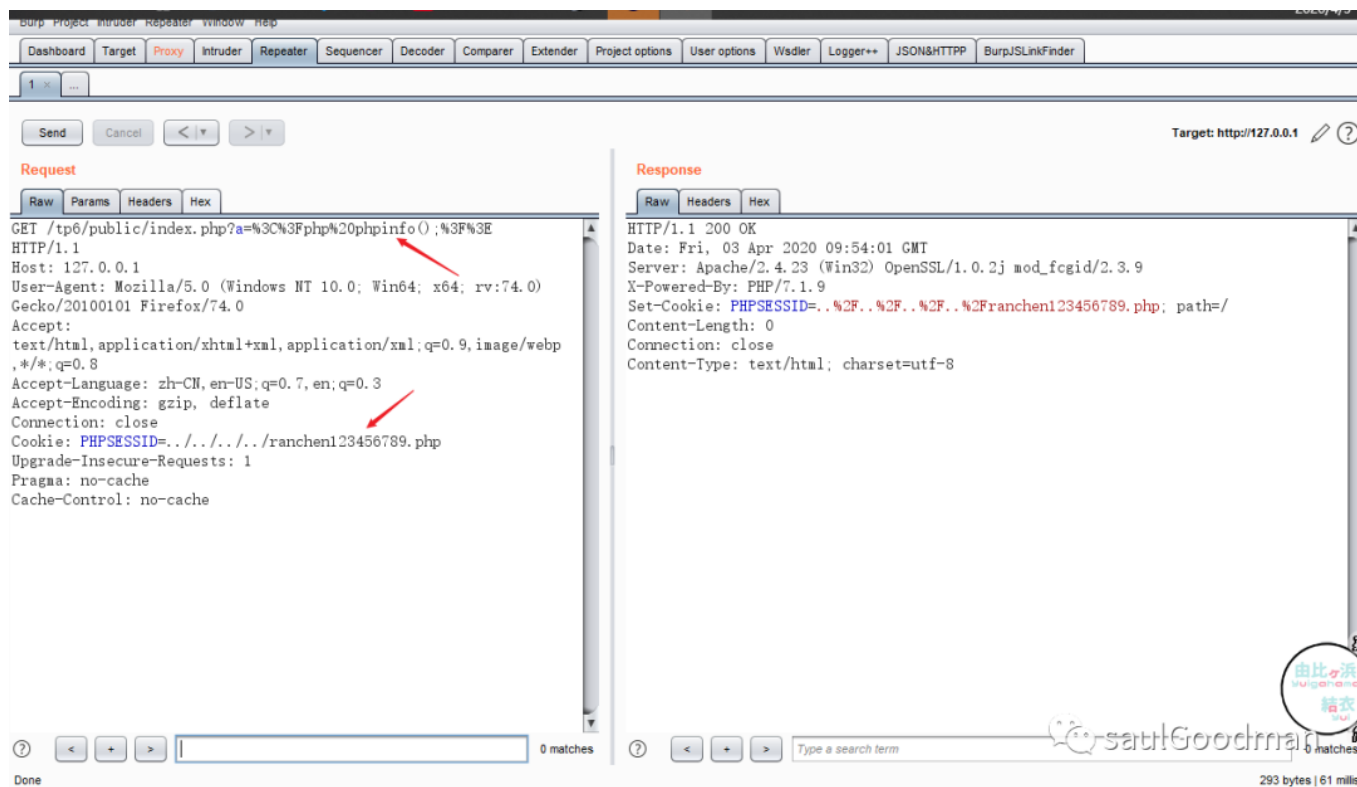
```
1 <?php
2 // 全局中间件定义文件
3 return [
4     // 全局请求缓存
5     // \think\middleware\CheckRequestCache::class,
6     // 多语言加载
7     // \think\middleware\LoadLangPack::class,
8     // Session初始化
9     \think\middleware\SessionInit::class
10 ];
11
```

saulGoodman

访问index 控制器,查看



打开正常，传入a 参数并且抓包



在a参数写入php代码。Phpsessid 写入一个文件名和路径总长度32位的内容。  
发送

名称	修改日期	类型	大小
app	2020/4/3 17:33	文件夹	
config	2020/4/3 17:29	文件夹	
extend	2020/4/3 17:29	文件夹	
public	2020/4/3 17:29	文件夹	
route	2020/4/3 17:29	文件夹	
runtime	2020/4/3 17:35	文件夹	
vendor	2020/4/3 17:29	文件夹	
view	2020/4/3 17:29	文件夹	
.example.env	2020/4/3 17:02	ENV 文件	1 KB
.gitignore	2020/4/3 17:02	文本文档	1 KB
.travis.yml	2020/4/3 17:02	YML 文件	2 KB
composer.json	2020/4/3 17:13	JSON 文件	2 KB
composer.lock	2020/4/3 17:22	LOCK 文件	29 KB
LICENSE.txt	2020/4/3 17:02	文本文档	2 KB
ranchen123456789.php	2020/4/3 17:54	PHP 文件	1 KB
README.md	2020/4/3 17:02	MD 文件	2 KB
think	2020/4/3 17:02	文件	1 KB

saulGoodman

成功写入

漏洞分析:

根据网上信息漏洞位置在:

tp6/vendor/topthink/framework/src/think/session/Store.php

所以我们直接定位到漏洞位置:

```

119 public function setId($id = null): void
120 {
121     $this->id = is_string($id) && strlen($id) === 32 ? $id : md5( str: microtime( get_as_float: (0.0) ) . session_create_id());
122 }
123

```

saulGoodman

判断phpsessid的值是否是字符串并且是否长度为32

向下走, 找到session保存的位置:

```

249  /**
250   * 保存session数据
251   * @access public
252   * @return void
253   */
254  public function save(): void
255  {
256      $this->clearFlashData();
257      $sessionId = $this->getId();
258
259      if (!empty($this->data)) {
260          $data = $this->serialize($this->data);
261          $this->handler->write($sessionId, $data);
262      } else {
263          $this->handler->delete($sessionId);
264      }
265
266      $this->init = false;
267  }
268

```

saulGoodman

关键代码就是 259-261，这里他判断session的值如果不为空就是进行序列化然后写入.其中也没有过滤。所以如果session 的值 我们能控制（就比如图1那样），就会直接造成一个任意文件写入的漏洞。

（在src/think/middleware/SessionInit.php文件里有phpsessid的获取方法。这里就不追了，有兴趣的可以自己去看）