ThinkPHP5.0.10-3.2.3缓存函数设计缺陷可导致Getshell

phpoop / 2017-08-09 18:31:00 / 浏览数 31470 技术文章 技术文章 顶(0) 踩(0)

0x00 框架运行环境

ThinkPHP是一个免费开源的，快速、简单的面向对象的轻量级PHP开发框架，是为了敏捷WEB应用开发和简化企业应用开发而诞生的。ThinkPHP从诞生以来一直秉承简洁
Thinkphp在使用缓存的时候是将数据 序列化 然后存进一个php文件中这就导致我们了我们在一些情况下可以直接getshell

0x01漏洞利用

该漏洞形成最关键的一点是
需要使用框架时，有使用缓存，才能利用这个漏洞

我们这里使用缓存查看官网对这个缓存类的说明以及利用方法

本地按照官方给的文档安装成功后，
根据官网给的缓存使用方法，
新建一个
方法，我们都清楚缓存一般是为了减少数据库的开销为设置的，所以缓存的数据一般也是从数据库获取到的
为了模拟线上，我们这里先查数据库数据在写入缓存。

这里我们写了一个
add添加数据的方法

%2F%2F%0D%0A = //+回车

执行完以后查看方法缓存目录

这里需要特别说的一点是
TP的缓存名字是不变的，所以我们在审计的时候不用怕缓存文件名猜不到的情况。

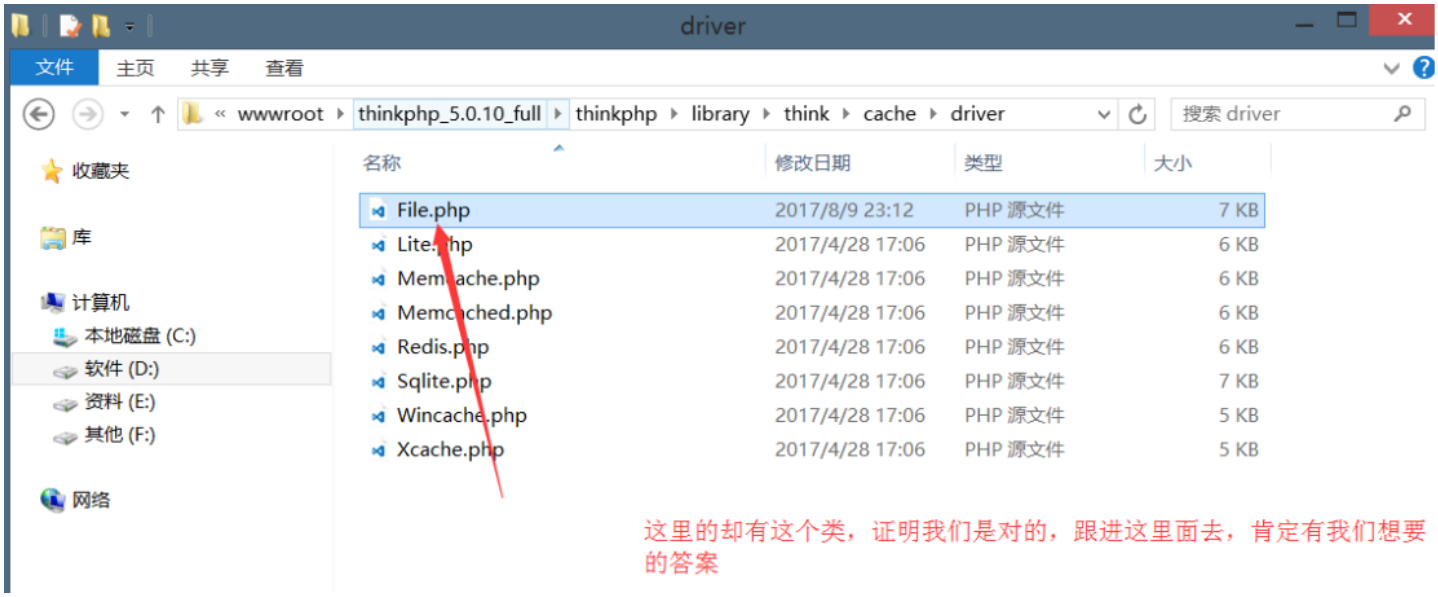
0x02漏洞分析

上面我们展示了漏洞利用方法，这里我们对这个漏洞进行分析

为了证明我们的逻辑是对的我们这里打印一下返回的数据

通过这个我们就可以知道了下面这个截图的意思

实例化
\thinkp\cache\driver\ 文件里面的
File类 并且调用
set
方法



缓存文件名称的获取方法

Thinkphp3.2 缓存函数设计缺陷

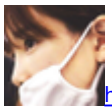
这个感觉没什么可以说的，和上面的原理是一样的，我们只演示攻击的方法

修复方案

通过上面的过程与分析我们可以清楚了解造成这个漏洞的主要原因就是换行与回车导致绕过了注释。那么我们修复的方法就很简单了只要删除这些即可
修复方法：
1，打开文件：thinkphp\library\think\cache\driver\File.php
2，找到：public function set($name, $value, $expire = null) 方法
3，添加：$data = str_replace(PHP_EOL, '', $data);

上一篇：【玩转linux系统】Linux内网渗透 下一篇：MSSQL注入技巧两则

1. 54 条回复


hades 2017-08-10 01:01:51

我来给你编辑出来~~赞

0 回复Ta


aaaaaaaaaaa 2017-08-10 01:28:57

据说是个0day。

0 回复Ta


answer 2017-08-10 02:20:20

据说是个0day

0 回复Ta


test1234 2017-08-10 02:21:22

0day在哪？

0 回复Ta


wooyun 2017-08-10 02:44:52

屌屌屌

0 回复Ta


hades 2017-08-10 02:46:15

晚上看公众号把~~

0 回复Ta

 ze7o 2017-08-10 02:50:52

0 回复Ta

 gdgdgdgd 2017-08-10 06:06:33

略微有些没懂

0 回复Ta

 hades 2017-08-10 06:38:03

文章晚上会公布文章

0 回复Ta

 anivia 2017-08-10 09:48:46

tp3.2.3？？？

0 回复Ta

浮萍 2017-08-10 10:20:42

先来膜拜一下

0 回复Ta


simeon 2017-08-10 13:02:43

看不了啊。

0 回复Ta


xxooad 2017-08-10 13:42:45

为什么没有？？？？？

0 回复Ta


shinpachi8 2017-08-10 16:17:56

据说是个0DAY

0 回复Ta


合肥滨湖虎子 2017-08-11 01:01:39

据说是个0DAY

0 回复Ta


mr.hao 2017-08-11 03:44:03

%0A%0A%24a%3D%24_GET%5B3%5D%3B%2F%2F%0A%24a%3Deval(%24_POST%5B'a3'%5D)%3B%23

3.2.3 加上replace 也可以用。上面的payload

需要猜temp目录下的文件名。。。

攻击点不高。。

0 回复Ta


null2017 2017-08-11 04:23:02

Thinkphp默认是否开启了cache功能？

0 回复Ta


流年 2017-08-11 05:44:34

几个前提：
1、缓存使用文件方式并且缓存目录暴露在web目录下面
2、攻击者要能猜到开发者使用的缓存key

TP5的缓存目录并非暴露在web目录下面 怎么执行？
TP3早就过了官方的维护生命周期已经不再更新维护了。

再说这个处理方案也有失偏颇啊 违背缓存数据的一致性。

0 回复Ta

 流年 2017-08-11 06:33:53

顺便在提下，TP3可以设置 DATA_CACHE_KEY 参数来避免被猜到缓存文件名 这些都是15年就反馈过的问题

0 回复Ta

 紫霞仙子 2017-08-11 06:39:51

0 回复Ta

 紫霞仙子 2017-08-11 06:41:48

。

0 回复Ta

 朱老黑 2017-08-11 07:12:50

<?php if (!defined('THINK_PATH')) exit();?>这个能绕过么？

0 回复Ta


独孤圣人 2017-08-11 07:35:39

不知道楼主看过这句话么
https://www.kancloud.cn/manual/thinkphp5/118008
5.0的部署建议是public目录作为web目录访问内容，其它都是web目录之外，当然，你必须要修改public/index.php中的相关路径。如果没法做到这点，请记得设置目录

0 回复Ta


null2017 2017-08-11 07:58:05

不用看这些文档，就单纯的看这个漏洞，在默认环境下是否可以黑盒复现？

0 回复Ta


__zz 2017-08-11 08:33:20

!!!厉害了，学习了

0 回复Ta


fzer0 2017-08-11 09:23:02

膜拜一下！
但实战中该怎么应用还不是很清楚

0 回复Ta


天使 2017-08-12 07:54:06

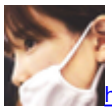
phpoop 大表哥注册了个小号来给你加油

0 回复Ta


anivia 2017-08-13 15:20:20

这是TP3.2.3 这个版本的漏洞吗？

0 回复Ta


hades 2017-08-14 01:09:25

文章里面不已经有说明了么？？

0 回复Ta


r00t4dm 2017-08-22 04:43:41

这个漏洞实战中是有一些问题的，大家可以来讨论一下

首先你要能确定使用cache的控制器，然后这个控制器还可以添加数据

就算找到了，还要找b0这个文件夹

实战中发现，无法确定使用cache的控制器，或者找不到cache操作方法。

其次，找不到b0文件夹。

利用难点：

无法确定相关目录

0 回复Ta

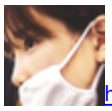

不知浮生愁 2017-08-27 11:15:32

楼主你好,我按这样改了一下以后反序列化时出错
$data = serialize($value);

$data = str_replace(PHP_EOL, '', $data);

初步想法是因为序列化以后将\n替换为空,导致序列化的字符串s:xx数值与替换以后的字符串长度不同引起的

0 回复Ta

 hades 2017-08-27 11:50:52

帖子上传图片的时候不要关闭上传页面 否则图片上传会失败

0 回复Ta

phpoop 2017-08-27 18:06:50

你好呢，之后也发现这个问题了，我的建议是不使用我文中的修复方法。
可以使用下面这个方法
1，thinkphp3.2的版本请选择开启：DATA_CACHE_KEY 这样就算你使用的cms是开源的人家发现了这个也无法使用。
2，tp3.2-tp5 做好目录权限，除公共目录绝对不要让外部可访问。

0 回复Ta

 不知浮生愁 2017-08-27 23:59:40

引用第33楼phpoop于2017-08-28 02:06发表的 回 31楼(不知浮生愁) 的帖子：
你好呢，之后也发现这个问题了，我的建议是不使用我文中的修复方法。
可以使用下面这个方法
1，thinkphp3.2的版本请选择开启：DATA_CACHE_KEY 这样就算你使用的cms是开源的人家发现了这个也无法使用。
2，tp3.2-tp5 做好目录权限，除公共目录绝对不要让外部可访问。
[url=https://xianzhi.aliyun.com/forum/job.php?action=topost&tid=1973&pid=5321][/url]

恩,好的,谢谢楼主

0 回复Ta

 流年 2017-08-28 02:50:00

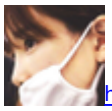
既然已经知道了框架内置了解决方案 说明框架设计之初就考虑到了，这个漏洞还有什么意义？无非是造成小白的恐慌

0 回复Ta

 hades 2017-08-28 03:18:11

我觉得只做技术交流 小白有误解

0 回复Ta

 hades 2017-08-28 03:20:04

这周会放一篇思路篇 有兴趣的人可以期待一下

0 回复Ta

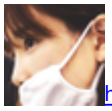
 流年 2017-08-28 06:19:41

可能你觉得是技术交流 但这样的交流被阿里云利用了， 这个漏洞经过阿里云的两次推送
波及较大。会造成很多的新人和用户的恐慌，官方从来没有承认这个漏洞，也无需要任何的修正补丁，我们需要给很多反馈的用户解释。

再来回答你的问题 就算源代码泄露，TP3.2本身就提供了参数设置 TP5.0只要按照官方的部署方案部署 哪里来的问题？

框架提供了机制，开发人员什么都不处理或者不使用正确的方法就做产品
这到底是产品（网站）的漏洞还是框架的漏洞？你用PHP写的网站有漏洞你会说PHP有漏洞么？你用TP开发的网站有漏洞
就说是框架有漏洞？有很多白帽子这点界限都分不清楚，不是说你能本地重现就是漏洞了，就算你能远程重现那也是某个网站有问题。

0 回复Ta


hades 2017-08-28 06:29:30

恩 给个联系方式 我们私下沟通一下

0 回复Ta


一讯 2017-08-28 11:03:01

其实就是看一下自己想怎么搞

0 回复Ta


我的昵称 2017-08-29 02:00:18

谁会把缓存文件放在用户可访问的目录呢？

0 回复Ta

 悠然岁月 2017-08-29 06:40:50

发帖子交流还可以，推送感觉就呵呵哒。

0 回复Ta

 itssme 2017-08-29 11:07:30

另外TP3.2对数据进行过滤处理之后，第一步写入数据库根本就无法进行下去...
此外，tp缓存文件第一行请注意：“<?php if (!defined('THINK_PATH')) exit();?>”
就算你猜到了缓存文件名称，打开之后也只会是空白，无法得到信息；

所以，这个只能算是极端理想情况下的案例吧，需要满足：
1、你猜到了key；
2、对方网站完全没有数据过虑。

PS：以TP3.2.2为例。

以上只有开发小白才有概率发生。

0 回复Ta

 hades 2017-08-29 11:23:55

现在的安全场景已经不是以前那种，随便就可以注入的时代了，出发的本意只是做技术探讨，并没有误导新手的情况，懂的人自然是懂的

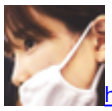
0 回复Ta


loophole 2017-08-30 02:36:53

请教一下大家，这样的缓存设计有什么好处吗？
写入的时候是使用序列化，然后读取的时候用的是先读取文件，然后反序列化。

为什么不直接 var_export 存储呢，这样会有安全隐患吗？

0 回复Ta


hades 2017-08-30 03:39:24

使用PHP的站点系统，在面对大数据量的时候不得不引入缓存机制。有一种简单有效的办法是将PHP的对象缓存到文件里。下面我来对这3种缓存方法进行说明和比较。

第一种方法：JSON

JSON缓存变量的方式主要是使用json_encode和json_decode两个php函数。json_encode可以将变量变成文本格式，这样就可以存到文件里。

使用样例如下：

```
// Store cache
file_put_contents($cachePath, json_encode($myDataArray));
// Retrieve cache
$myDataArray = json_decode(file_get_contents($cachePath));
```

优势：

- 变量序列化后依然可读

- 可以给其他系统使用，因为JSON格式是标准的

劣势：

- 只对UFT-8的数据有效，其他编码可能不能很好工作
- 只对stdClass类的示例有效

第二种方法：序列化

序列化的方式主要使用serialize和unserialize这2个函数，序列化的方式和JSON都是，都是以文本方式存储。

使用示例

```
// Store cachefile_put_contents($cachePath, serialize($myDataArray));// Retrieve cache$myDataArray = unserialize(file_get_c
```

优势：

- 允许非UTF-8的变量
- 支持除了stdClass 示例外的其他实例

劣势：

- 编码后的文本对人来说是不可读的
- 无法被其他语言的系统引用

第三种方法：Var_export

这种方式是用var_export函数将变量内容打印到一个PHP文件里，使用include的方式来重新获取变量内容。因此生成的缓存文件时一个php文件，内容如下

```
<?php
return /*var_export■■■*/;
?>
```

使用示例：

```
// Store cache
file_put_contents($cachePath, "<?php\nreturn " . var_export($myDataArray, true) . ";");
// Retrieve cache
$myDataArray = include($cachePath);
```

优势：

- 对编码格式无要求，允许非UTF-8的编码
- 缓存文件易读
- 获取变量的时候直接使用语言特性，而非函数
- 当使用opcode的时候，缓存php文件会放在opcode的缓存里（这实际上是一个劣势)

劣势：

- 不能缓存不带__set_state 方法的对象
- var_export出来的变量里不能带有影响php语法解析的内容，触发语法错误，可能影响你的php应用

性能测试

是用5组不同大小（904B, ~18kB, ~250kB, ~4.5MB and ~72.5MB）的数组，进行以下测试。

1. 使用编码函数对数据进行10次编码
2. 计算编码后的数据的大小
3. 对编码后的数据进行10次解码

结论

2.53GHz, 4GB, Ubuntu linux, PHP 5.3.0RC4.这样配置的笔记本上，测试的结果如下：

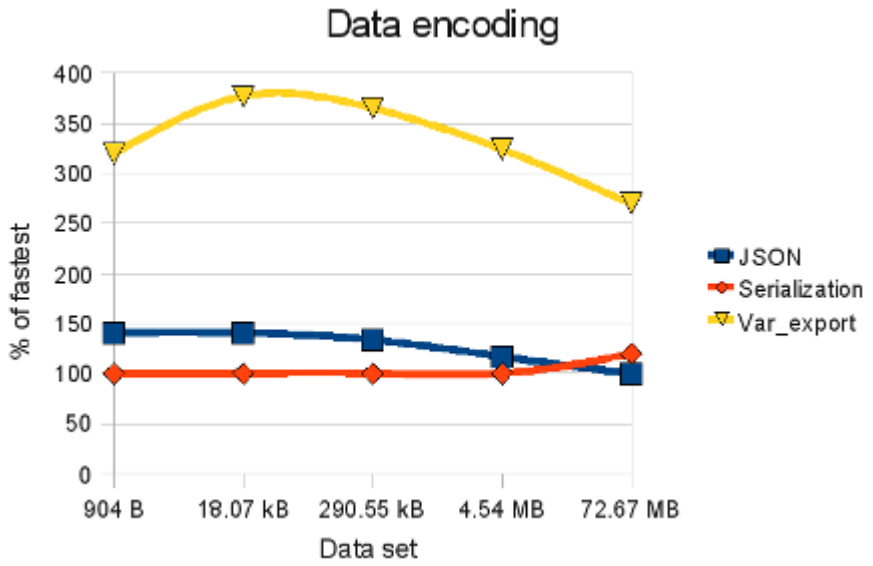
904 B array	JSON	Serialization	var_export / include
Length	105	150	151
Encoding	0.0000660419464111	0.00004696846008301	0.00014996528625488
Decoding	0.0011160373687744	0.00092697143554688	0.0010221004486084

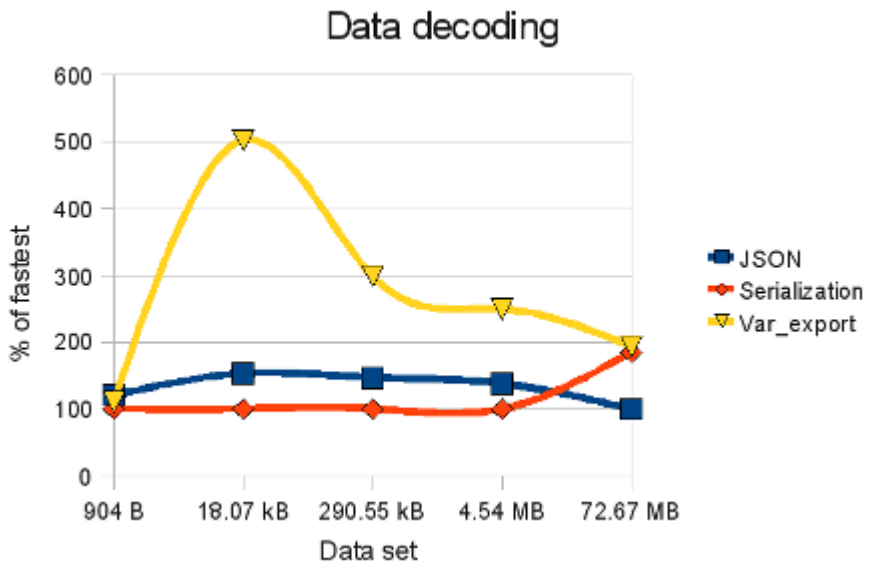
18.07 kB array	JSON	Serialization	var_export / include
Length	1965	2790	3103
Encoding	0.0005040168762207	0.00035905838012695	0.001352071762085
Decoding	0.0017290115356445	0.0011298656463623	0.0056741237640381

290.59 kB array	JSON	Serialization	var_export / include
Length	31725	45030	58015
Encoding	0.0076849460601807	0.0057480335235596	0.02099609375
Decoding	0.014955997467041	0.010177850723267	0.030472993850708

4.54 MB array	JSON	Serialization	var_export / include
Length	507885	720870	1059487
Encoding	0.13873195648193	0.11841702461243	0.38376498222351
Decoding	0.29870986938477	0.21590781211853	0.53850317001343

72.67 MB array	JSON	Serialization	var_export / include
Length	8126445	11534310	19049119
Encoding	2.3055040836334	2.7609040737152	6.2211949825287
Decoding	4.5191099643707	8.351490020752	8.7873070240021





上面2个图表示100%是最佳的，var_export在编码和解码的性能上不佳。建议在数据量小的时候使用序列化的方法，如果数据量非常大，那就要考虑数据结构的问题了。

文章来源：http://www.opstool.com/article/262

0 回复Ta

你好小兵 2017-08-30 15:52:18

我不知道大家有没有仔细阅读这篇文章,这边文章我看了,我的理解是,博主说的是thinkphp3.2 和thinkphp5.0
对用户像服务器发送请求时的缓存存在漏洞,这个地方thinkphp3.2 和thinkphp5.0
默认配置是没有缓存用户像服务器发送请求,这个只是个人根据这篇文的个个人见解,大家可以验证一下你们使用thinkphp 开发的项目

0 回复Ta


风之旅人 2017-09-19 05:41:27

请问大牛这个3.2.2不行吗，为什么只到3.2.3呢

0 回复Ta


hades 2017-09-19 08:23:17

可以自行测试~~

0 回复Ta


shutdown_r 2017-09-25 11:17:30

有可以不用回车的办法，在content中直接把?>闭合了，然后再写一个<?php
content=%3f%3e%3c%3fphp%24a%3Deval(%24_POST%5B%27b%27%5D)%3B%23

0 回复Ta

phpoop 2017-09-26 08:55:11

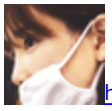
你比我聪明，2333，系列化坑很多。

0 回复Ta


我来了呵呵 2017-10-17 15:26:45

还别说，真的有php同事被阿里的这篇TP “漏洞” 吓尿了。
我感觉按照这种思路，使用任何框架直接 $_GET
接收数据，演示给大家看，什么注入啊，xss，Getshell等等，一年可以找出365个漏洞，阿里可以天天推送吓新手，提升自身高大尚形象。“哇噻，阿里真厉害，你看你写

0 回复Ta


hades 2017-10-17 16:24:24

看来以后隐藏掉所有厂商信息吧 自己猜去吧

0 回复Ta

xb1ng 2017-10-28 14:59:24

没有home目录怎么办。。

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板