链滴

ThinkPHP6 任意文件操作漏洞分析

作者：someone38063
原文链接：https://ld246.com/article/1579965339516
来源网站：链滴

这个洞出来也有一段时间了，看了创宇的 paper 后觉得蛮简单的，决定自己在本地搭建复现一，记录一下学习的过程。
<h2 id="-环境准备"> 环境准备</h2>
apache + thinphp(<=6.0.0 版本 <=6.0.2) + php7 以上

ThinkPHP6 起只能使用 composer 来安装，安装 composer、php、apache 的过程我就不赘了。

执行命令:<code>composer create-project topthink/think tp 6.0.0</code>，其中 tp 是你的文件夹命名，6.0.0 是版本号，6.0.1 也可。
这里说一个问题，我这个时间 Thinkphp 的最新版是 6.0.2，用上面的命令下载下来 framework 是 6.0.2 版本的，我们需要再执行一条命令：<code>composer require topthink/framework:6.0.0</code>：此时就会把将 6.0.0 的版本把 6.0.2 给替换掉

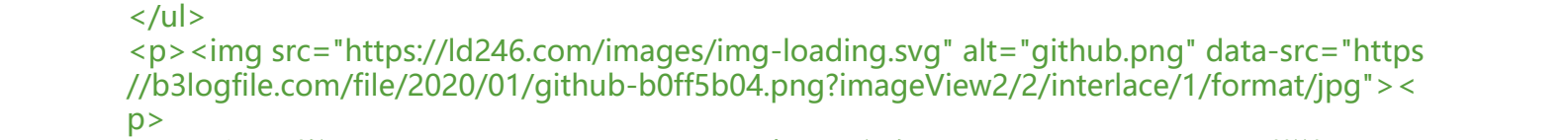
进入 tp 的安装目录，执行 <code>php think run</code>，它会开启一个临时开发环境的服务器，默认运行在 <code>localhost:8000</code>，打开浏览器访问显示正常即可

<blockquote>
漏洞复现在 apache 下进行
</blockquote>
<h2 id="-漏洞分析"> 漏洞分析</h2>
漏洞影响的版本：top-think/framework 6.x < 6.0.2

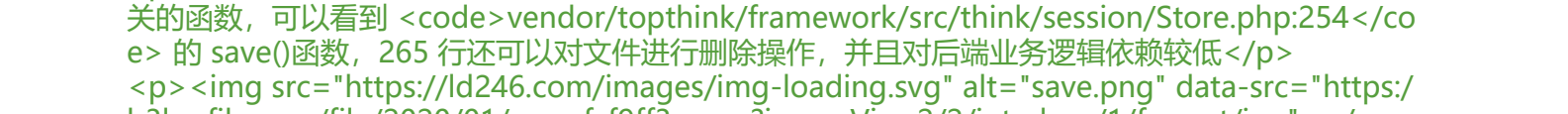
官方信息

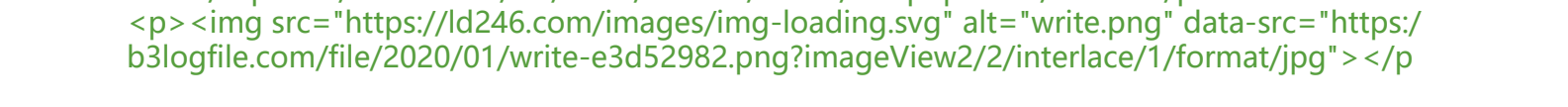
ThinkPHP 发布的补丁声称修复了一处由于不安全的 SessionId 导致的任意文件操作漏洞：在开 Session 的情况下可以导致创建任意文件以及删除任意文件，特定情况下可以 getshell

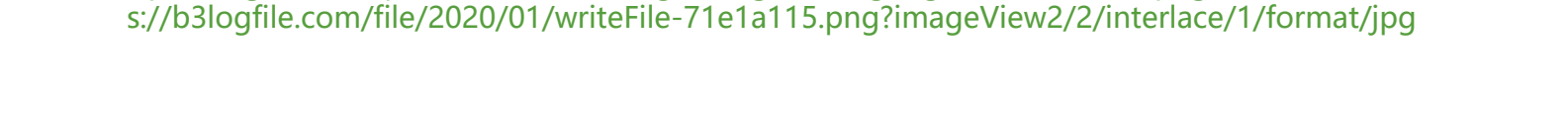
根据这些信息，我们到官方 GitHub 的 commit 页面找一下相关的提交记录：



可以看到位于 src/think/session/Store.php 中 212 行在设置 <code>id</code> 时增加了一函数：<code>ctype_alnum($text)</code>。
查一下 PHP 官方手册，这个函数是用来检测输入的 <code>$text</code> 中所有的字符全部字母和(或者)数字，返回 TRUE 否则返回 FALSE


根据文件目录和更改的函数部分猜测：可能是存储 Session 时导致的文件写入；然后跟进找一下关的函数，可以看到 <code>vendor/topthink/framework/src/think/session/Store.php:254</code> 的 save()函数，265 行还可以对文件进行删除操作，并且对后端业务逻辑依赖较低


可以看到设置了 $sessionId，并且调用了一个 write 函数，继续跟进，找到 write()函数 <code>endor/topthink/framework/src/think/session/driver/File.php:210</code>


继续跟进，找到 writeFile()函数


>
可以看到调用了 <code>file_put_contents()</code> 函数，这里是真正写入文件的操作了</p

<img src="https://ld246.com/images/img-loading.svg" alt="fileputcontents.png" data-sr
="https://b3logfile.com/file/2020/01/fileputcontents-448224fd.png?imageView2/2/interlace/
/format/jpg">

接下来我们反向分析一下，看看能不能找到可控点

 函数 <code>file_put_contents($path,$content,LOCK_EX)</code> 中参数 <code>$
ath,$content</code> 来源于函数 <code>writeFile($path,$data)</code>

函数 <code>writeFile($path,$data)</code> 中参数 <code>$path,$data</code> 来源于
数 <code>write(String $sessionID,String $sessiData)</code>

函数 <code>write(String $sessionID,String $sessiData)</code> 中参数 <cod
>$sessionID,$sessiData</code> 来源于 <code>save()</code> 中调用了 <code>write()</cod
>，同时传入的参数 <code>$sessionId</code> 的值是调用 <code>getId()</code> 传入的

综上：文件名来源于 <code>$sessionId</code>

当传入的 id 值长度为 32 并且.....etc 时，创建 <code>sessionId</code>，然后进行 <code>g
tId()</code>

<img src="https://ld246.com/images/img-loading.svg" alt="session.png" data-src="https
//b3logfile.com/file/2020/01/session-ca28bb38.png?imageView2/2/interlace/1/format/jpg">
/p>

接下来找调用 <code>setId()</code> 的地方 <code>vendor/topthink/framework/src/think
middleware/SessionInit.php:46</code>

<img src="https://ld246.com/images/img-loading.svg" alt="sessionI.png" data-src="http
://b3logfile.com/file/2020/01/sessionI-c0b945ed.png?imageView2/2/interlace/1/format/jpg"

其中 <code>cookieName</code> 的值为 <code>PHPSESSID</code>，而 <code>$sessi
nId</code> 是 <code>cookie</code> 中名为 <code>PHPSESSID</code> 的值，因此是攻击
可控的，从而导致写入的文件名可控。
但是默认环境下，<code>session</code> 的内容由 <code>vendor/topthink/framework/sr
/think/session/Store.php:261</code> 的变量 <code>$data</code> 传入：
<pre><code class="language-php highlight-chroma"><span c
ass="highlight-cl">
<span class="high
ight-nv">$data=$th
s->serialize</sp
n>($this<span clas
="highlight-o">->data<span class="highlig
t-p">);

</code></pre>

原文链接：ThinkPHP6 任意文件操作漏洞分析

<code>$data</code> 在默认环境中为空：
<pre><code class="language-php highlight-chroma"><span c
ass="highlight-cl">
<span class="high
ight-sd">/**
<span cla
s="highlight-sd">
<span cla
s="highlight-sd">   * Session数据
<span cla
s="highlight-sd">
<span cla
s="highlight-sd">   * @vararray
<span cla
s="highlight-sd">
<span cla
s="highlight-sd">   */

<span class="high
ight-k">protected$data<span class="highlight
o">=[];

</code></pre>
写入的 <code>session</code> 内容是由实际的后端业务逻辑来决定的，所以说只有苛刻的条下才能写入 webshell。并且一开始就说了需要在环境开启 <code>session</code> 的情况下才可实现任意文件操作(默认环境不开启 session)

我们在 <code>app\controller\index.php</code> 中增加一些代码后，如下：

<pre><code class="language-php highlight-chroma"><span c
ass="highlight-cl"><?<span class="highlight-nx">ph

<span class="high
ight-nx">namespaceapp\controller;

<span class="high
ight-nx">usethink\facade\Session;

<span class="high
ight-nx">useapp\BaseController;

<span class="high
ight-nx">classIndexextendsBaseController

<span class="high
ight-p">{

<span class="high
ight-nx">publicfunctionindex()

```
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high
ight-p">{</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high
ight-nx">Session</span><span class="highlight-o">::</span><span class="highlight-na">s
t</span><span class="highlight-p">(</span><span class="highlight-s1">'name'</span><s
an class="highlight-p">,</span><span class="highlight-s1">'thinkphp'</span><span class=
highlight-p">);</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high
ight-nx">return1</span><span class="highlight-p">;</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high
ight-c1">// return'&lt;style type="text/css"&gt;*{ padding: 0; marg
n: 0; } div{ padding: 4px 48px;} a{color:#2E5CD5
cursor: pointer;text-decoration: none} a:hover{text-decoration:underline;
nbsp;} body{ background: #fff; font-family: "Century 
othic","Microsoft yahei"; color: #333;font-size:18px;} h1{ font
size: 100px; font-weight: normal; margin-bottom: 12px;&nb
p;} p{ line-height: 1.6em; font-size: 42px }&lt;/style&gt
&lt;div style="padding: 24px 48px;"&gt; &lt;h1&gt;:) &lt;/h
&gt;&lt;p&gt; ThinkPHP V6&lt;br/&gt;&lt;span style="font-size:30px"&gt
13载初心不改 - 你值得信赖的PHP框架&lt;/span&gt;&lt;/p&gt;&lt;/div&gt;&lt;scrip
 type="text/javascript" src="https://tajs.qq.com/stats?sId=64890268" char
et="UTF-8"&gt;&lt;/script&gt;&lt;script type="text/javascript" src="https://e.top
hink.com/Public/static/client.js"&gt;&lt;/script&gt;&lt;think id="eab4b9f840753f8e7"&
t;&lt;/think&gt;';
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-c1"></span>
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high
ight-p">}</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high
ight-nx">publicfunctionhello</span><span class="highlight-p">(</span><span class="highl
ght-nv">$name</span><span class="highlight-o">=</span><span class="highlight-s1">'T
inkPHP6'</span><span class="highlight-p">)</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high
ight-p">{</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high
ight-k">return</span><span class="highlight-s1">'hello,'</span><span class="highlight-o"
.</span><span class="highlight-nv">$name</span><span class="highlight-p">;</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high
ight-p">}</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high
ight-p">}</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
```


原文链接：ThinkPHP6 任意文件操作漏洞分析

</code></pre>
<blockquote>
 忘了说了 thinkphp6 开启 session 的方法：删除 <code>/app/middleware.php</cod
> 最后一行的注释
</blockquote>
<img src="https://ld246.com/images/img-loading.svg" alt="kq.png" data-src="https://b3
ogfile.com/file/2020/01/kq-3e370bda.png?imageView2/2/interlace/1/format/jpg">
<h2 id="本地环境复现">本地环境复现</h2>
很简单，只需要构造 PHPSESSID 的值即可，值为 <code>string</code>&&长度为
32
<img src="https://ld246.com/images/img-loading.svg" alt="tp.png" data-src="https://b3
ogfile.com/file/2020/01/tp-34245d82.png?imageView2/2/interlace/1/format/jpg">
此时查看一下生成的 session，生成的 session 文件保存在 <code>\runtime\session</code>
下
<img src="https://ld246.com/images/img-loading.svg" alt="sessionphp.png" data-src="
ttps://b3logfile.com/file/2020/01/sessionphp-44ef96ff.png?imageView2/2/interlace/1/format/
pg">
session 里的内容:
<pre><code class="language-json highlight-chroma"><span c
ass="highlight-cl">
<span class="high
ight-err">a:1:</s
an>{s:4:<span clas
="highlight-nt">"name";s<span class="highlig
t-p">:8:<s
an class="highlight-s2">"thinkphp";<span clas
="highlight-p">}

</code></pre>
可以看到 session 的内容经过了序列化操作，只要将 session 的内容反序列化即可 getshell

如果要 getshell 的话，后端需要有类似的 <code>Session::Set('name',$_POST['i'])</code>
码才可以利用

<h2 id="总结-">总结：</h2>
在复现的过程，也遇到了不少问题：首先 ThinkPHP6 开始不支持 git 了，只能通过 composer
操作，由于从来没用过它也没经验，一开始安装环境一直下载不到旧版本，后来得到师傅的帮助终于
好了 ThinkPHP6.0.0 的环境，在这里感谢一下师傅 <a href="https://ld246.com/forward?goto=ht
ps%3A%2F%2Fweibo.com%2Fu%2F5332465356" target="_blank" rel="nofollow ugc">@P1an
 对我的帮助。
这个漏洞其实很简单，就是用户可控变量导致的，也没有对一些数据的过滤等等。需要一定条件
可以利用，也就是开启 session；写 webshell 还要看具体的后端业务逻辑等等。我觉得就这个框架
看其实可以更深入的进行挖掘，希望有大佬可以和我一起探讨学习
参考的 paper：<a href="https://ld246.com/forward?goto=https%3A%2F%2Fpaper.seebu
.org%2F1114%2F" target="_blank" rel="nofollow ugc">ThinkPHP6 任意文件操作漏洞分
