

# fastadmin(V1.0.0.20200506\_beta) 前台 getshell(文件上传解析) 漏洞分析

“ 0x1. 简介 FastAdmin 是一款基于 ThinkPHP 和 Bootstrap 的极速后台开发框架。 补天平台介绍：近日，补天漏洞响应平台监测到互联网上出现 Fastadmin 文件...

## 0x1. 简介

FastAdmin 是一款基于 ThinkPHP 和 Bootstrap 的极速后台开发框架。

补天平台介绍：近日，补天漏洞响应平台监测到互联网上出现 Fastadmin 文件上传漏洞，exp 被公开。该漏洞源于网络系统或产品的代码开发过程中存在设计或实现不当的问题，可导致文件上传并解析为可执行文件。目前厂商已发布新版本修复此漏洞，补天漏洞响应平台建议受影响的客户将框架更新至安全版本。

影响版本：V1.0.0.20180911\_beta~V1.0.0.20200506\_beta

修复建议：

升级 Fastadmin 版本到 V1.0.0.20200920\_beta，详见官网链接：

<https://www.fastadmin.net/download.html>

## 0x2. 漏洞详情

利用限制：需要开启会员中心功能，且登录会员中心。

/application/config.php 文件中：

```
'usercenter'          => true,
```

## 漏洞分析

/application/index/User.php 文件

第 58-67 行：

```
public function _empty($name)
{
    $data = Hook::listen("user_request_empty", $name);
    foreach ($data as $index => $datum) {
        $this->view->assign($datum);
    }
    return $this->view->fetch($name);
}
```

user\_request\_empty 为开发者预留的钩子可以忽视不看，主要看 return \$this->view->fetch(\$name);

此方法中的 \$name 参数可控，并且将 \$name 的值传入到了 fetch() 函数中。

fetch() 为 thinkphp 的解析模板函数，其返回模板文件渲染后的内容

fetch() 函数的关键内容如下：

```

public function fetch($template, $data = [], $config = [])
{
    if ('' == pathinfo($template, PATHINFO_EXTENSION)) {

        $template = $this->parseTemplate($template);
    }

    if (!is_file($template)) {
        throw new TemplateNotFoundException('template not exists:' . $template, $template);
    }

    App::$debug && Log::record('[ VIEW ]' . $template . ' [ ' . var_export(array_keys($data), true) . ' ]', 'info');
    $this->template->fetch($template, $data, $config);
}

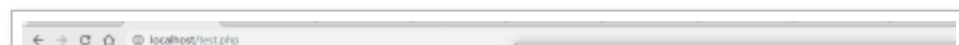
```

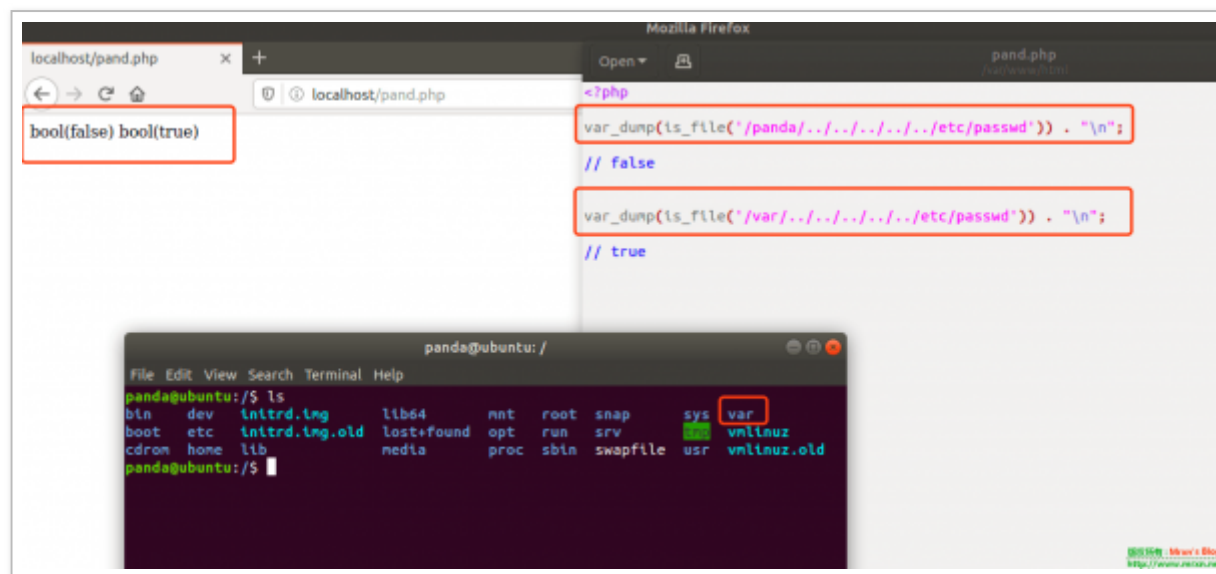
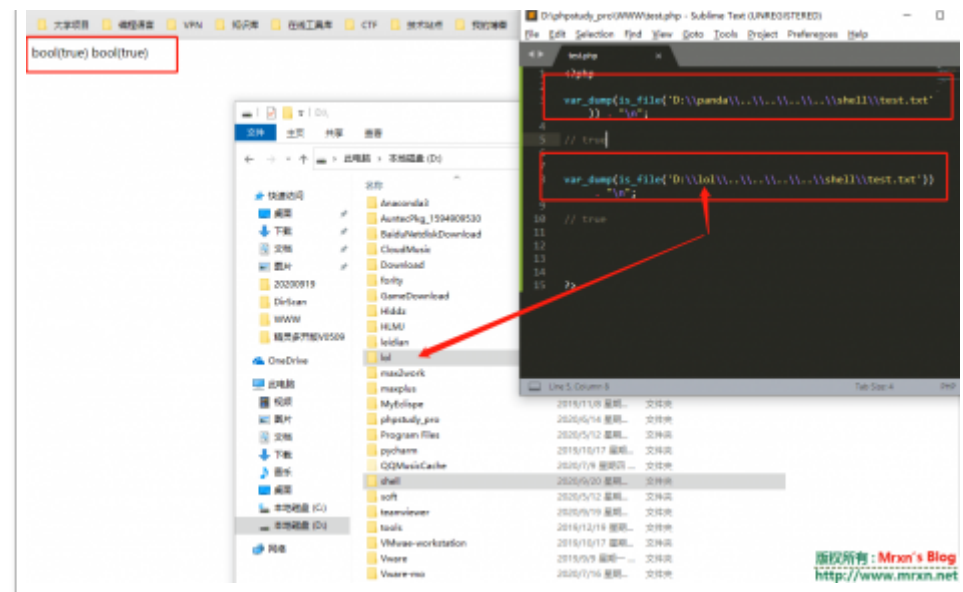
继续调用栈可以看下其实这个 fetch() 函数调用的是内置模板引擎的 fetch 方法，这个方法实际上就是将要输出的页面内容赋值给一个变量，为了方便，thinkphp 在对模板渲染的过程中，添加了 php 标签功能，使得其可以解析 php 代码。

总之一句话，这个漏洞其实就是由于对传入变量过滤不严导致的模板引擎注入漏洞，只要控制了传入模板的文件，就可以利用模板本身的渲染功能，实现包含漏洞 [getshell](#)

另外需要注意的是，当验证传入的模板是否是文件时，使用的 is\_file() 函数，这个函数在 Linux 下和 windows 下的判断会有所不同，具体如下：

1、在 linux 下利用 is\_file() 来判断类似于 /\*\*\*\*/../../../etc/passwd 文件时，如果 \*\*\*\* 是不存在的目录，则会返回 false，在 windows 下，这个目录存在与否，均返回 true，如下图所示：

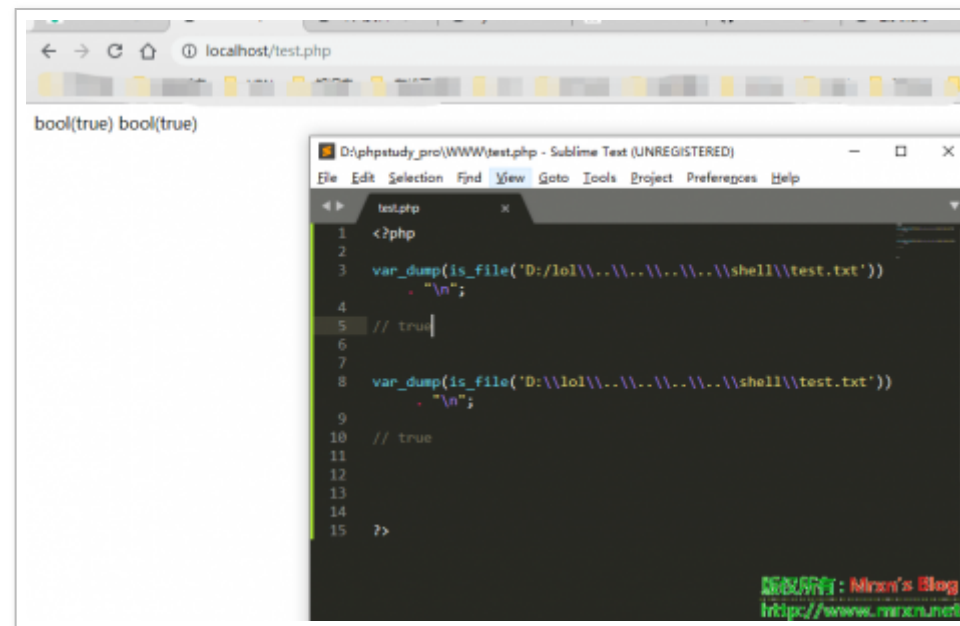




2、在 linux 下， `is_file()` 函数判可用于判断符号链接

3、在 linux 下， `is_file` 函数会受到权限的影响，当前用户权限不足或父目录没有设置 + x 权限时， `is_file()` 会返回 false

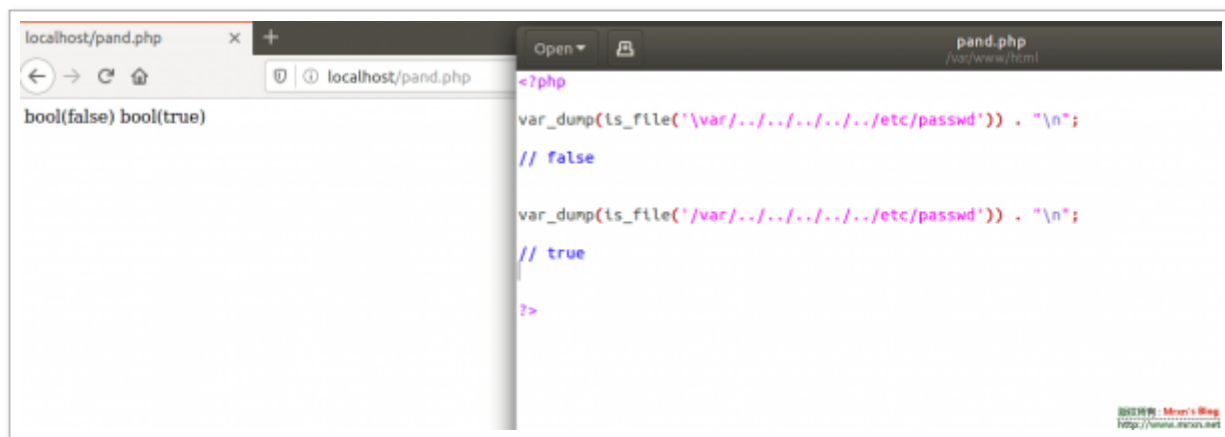
4、windows 系统里面 / 和 \ 都可以使用，但是在 linux 下只能使用 / 来分隔路径，因此这会导致 `is_file()` 在不同系统下的返回结果不一致



The screenshot shows a web browser window at `localhost/test.php` displaying the output `bool(true) bool(true)`. Overlaid on the browser is a Sublime Text editor window titled `D:\phpstudy_pro\WWW\test.php - Sublime Text (UNREGISTERED)`. The editor contains the following PHP code:

```
1 <?php
2
3 var_dump(is_file('/lol/../../../../../../../../shell/test.txt'))
4     . "\n";
5 // true
6
7 var_dump(is_file('D:\\lol\\../../../../../../../../shell/test.txt'))
8     . "\n";
9
10 // true
11
12
13
14
15 ?>
```

At the bottom right of the code editor, there is a signature: `DEVELOPER: Miran's Blog` and a URL: `http://www.miran.me/`.



5、is\_file() 判断文件时，如果文件大小超过  $2^{32}$  时，会判断失败

### 0x3. 漏洞验证

通过前文可知，这个漏洞的利用点在 `_empty()` 函数，需要注意的是，在官方文档中通常 `_empty()` 方法是用来判断一个方法是否存在，如果不存在，则进入该函数。而这里是开发者自定义的方法，因此直接传入 `_empty` 方法，调用 `name` 参数即可。

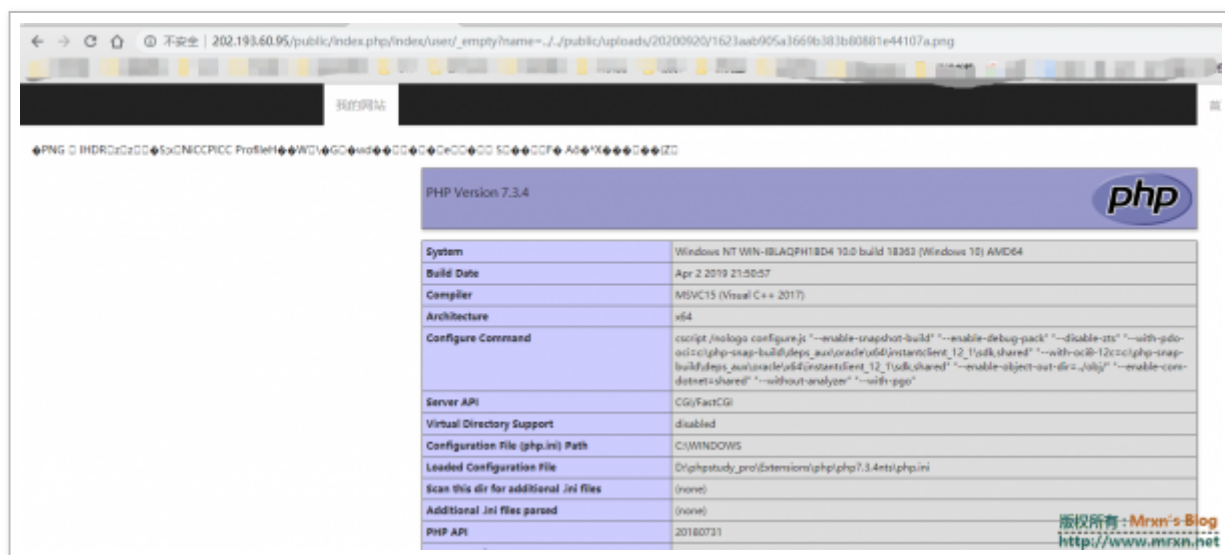
利用过程如下：

在前台的会员中心，个人资料处，上传修改头像：





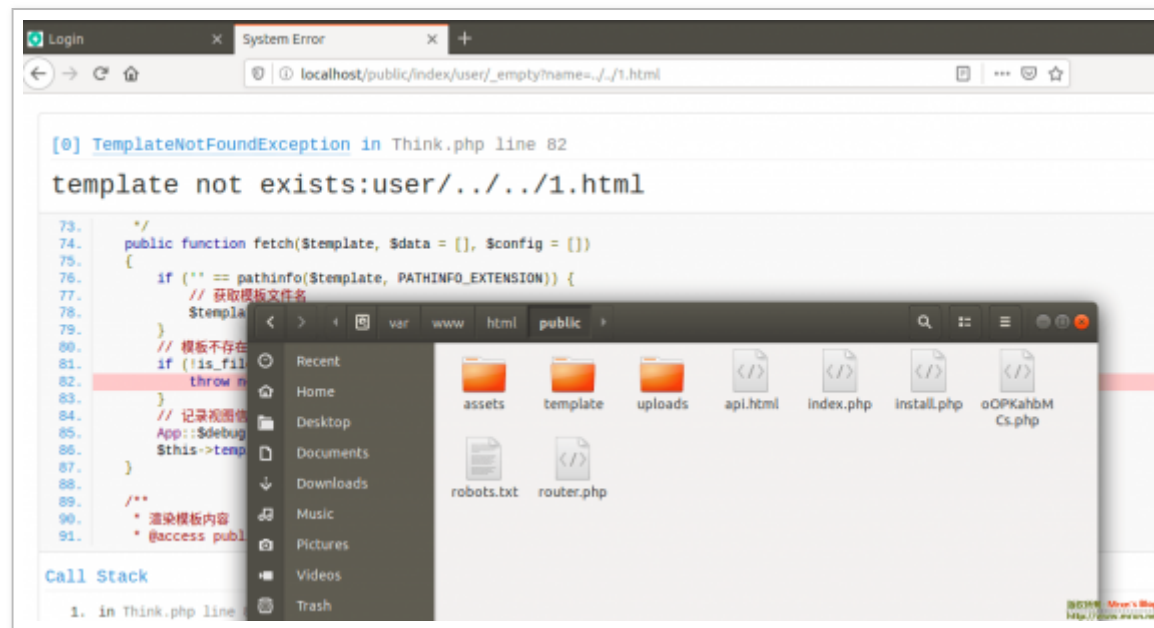
记录下路径后，成功 getshell



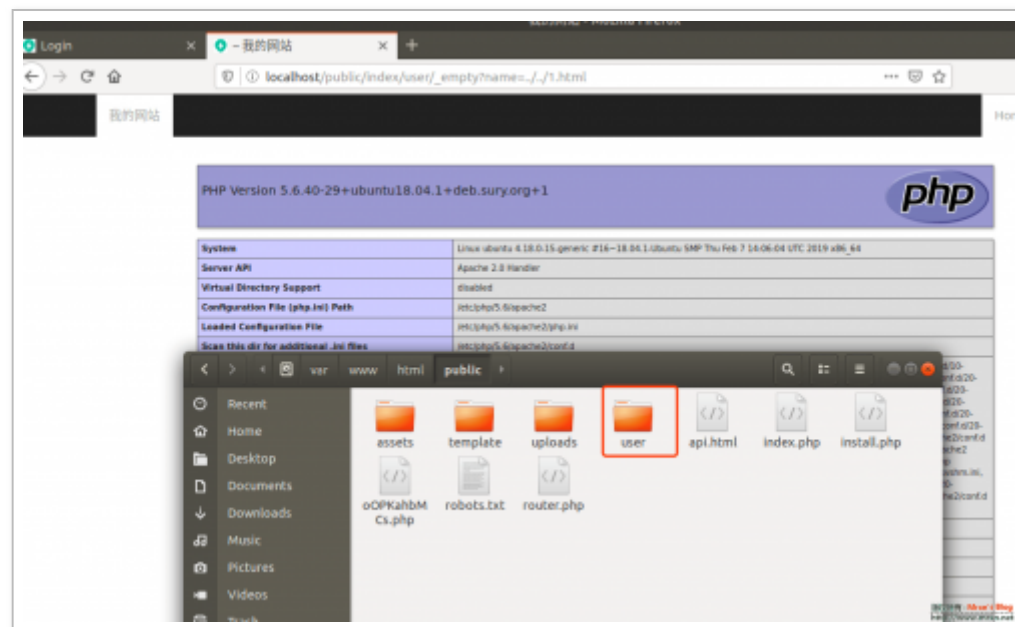
在 Linux 下 通过这种方法会失效 因为在 /public 路径下不存在 user 目录 由前文中的知识



点可以知道，当不存在这个目录的时候，无论怎么跳转目录，`is_file()` 函数返回的结果始终未 `false`，因此无法利用该 漏洞，如下图所示：



当我们在 /public 目录下创建文件夹 /user，在利用，即可成功：



最后感谢 @ joseph 师傅提供的 漏洞 点，又学习了一波

文章出自：<https://forum.90sec.com/t/topic/1294>

作者：<https://forum.90sec.com/u/panda>

大家可以投稿 90 注册啊！给你们投票！哈哈哈