PHPCMS_V9.2 任意文件上传 getshell 漏洞分析

介绍 & 预备知识

介绍：PHPCMS 是一款网站管理软件。该软件采用模块化开发, 支持多种分类方式。

预备知识

PHPCMS 是采用 MVC 设计模式开发, 基于模块和操作的方式进行访问，采用单一入口模式进行项目部署和访问，无论访问任何一个模块或者功能，只有一个统一的入口。

参数名称	描述	位置	备注
m	模型 / 模块名称	`phpcms/modules`中模块目录名称	必须
c	控制器名称	`phpcms/modules/模块/*.php` 文件名称	必须
a	事件名称	`phpcms/modules/模块/*.php` 中方法名称	

模块访问方法 [示例]: `http://www.xxx.com/index.php?m=content&c=index&a=show&id=1`

其中 `m = content` 为模型 / 模块名称 位于 phpcms/modules/content

`c = index` 为控制器名称 位于 **phpcms/modules/content/index.php**

`a = show` 为时间名称 位于 **phpcms/modules/content/index.php** 中 `show()` 方法 id = 1 为其他参数 与正常 get 传递参数形式相同

还有一点就是访问 `http://www.xxx.com/index.php`

phpcms 默认路由会定位到 **content** 模块的 **index** 控制器中的 `init` 操作，因为系统在没有指定模块和控制器的时候，会执行默认的模块和操作.

所以跟访问 `http://www.xxx.com/index.php?m=content&c=index&a=init` 是一样的

> 参考来源：http://www.sjzphp.com/webdis/router_url_907.html

环境搭建 & 所需工具

- phpstudy2018
    - `php-5.4.45-nts + Apache`
- PHPCMS_V9.2
- Burpsuite2.1，2021 年最新那个 burp 编码有问题（可能我没调好），数据乱码，导致上传错误

测试站点网址：`www.phpcms92.com`

访问 `/install/install.php` 文件进行安装，下一步

下一步，配置相关信息

安装完成！！！

漏洞复现

访问首页 `index.php`

注册一个账户（这里我以 Tao 这个普通用户进行演示）

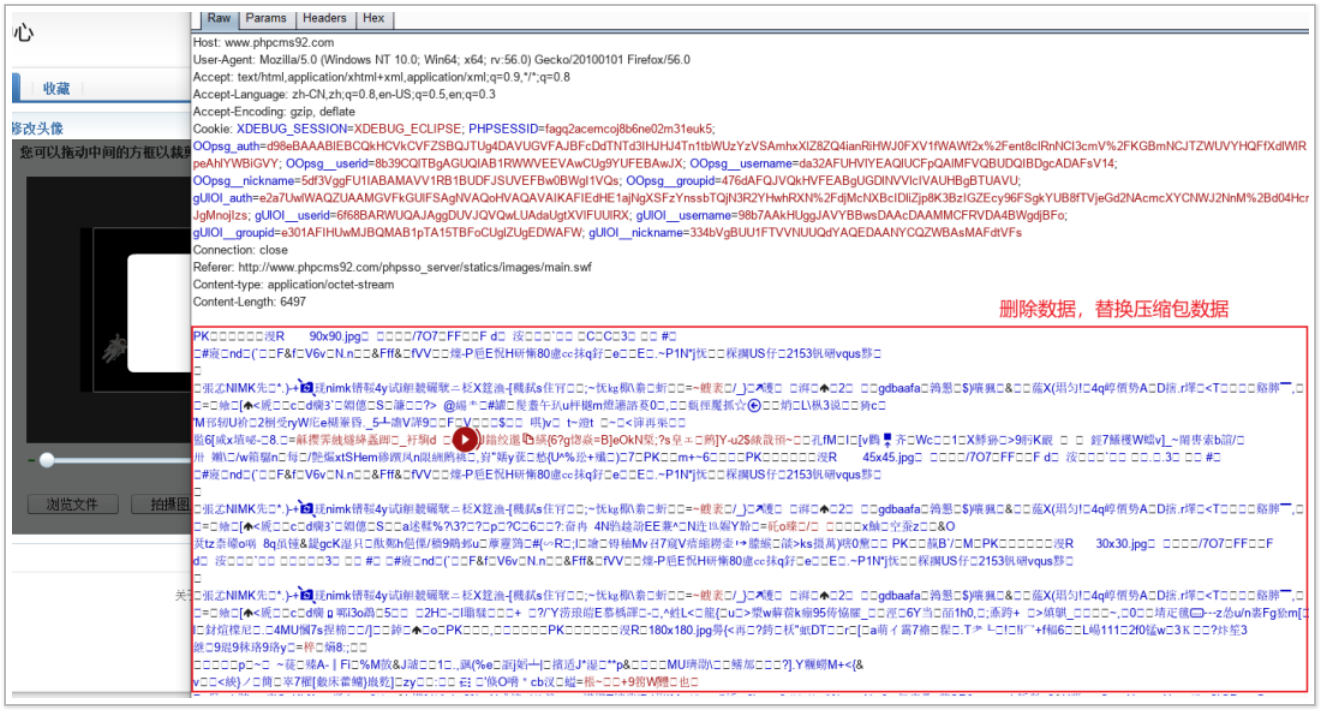
到个人主页修改头像处，上传头像

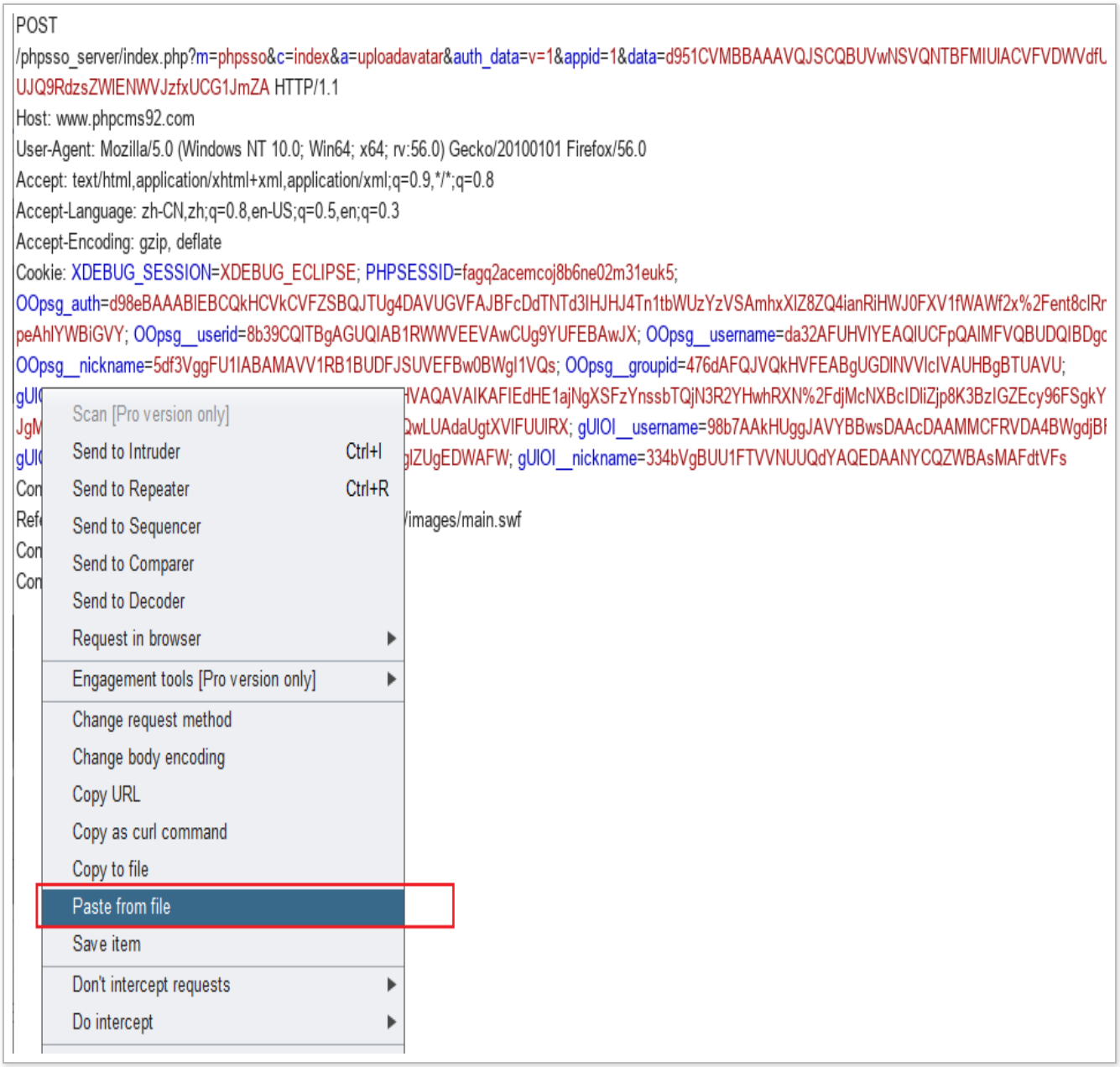
在此之前，还要准备一个后缀为 `zip` 的压缩包，具体内容如下：

php 文件需要放在二层目录下然后再进行压缩

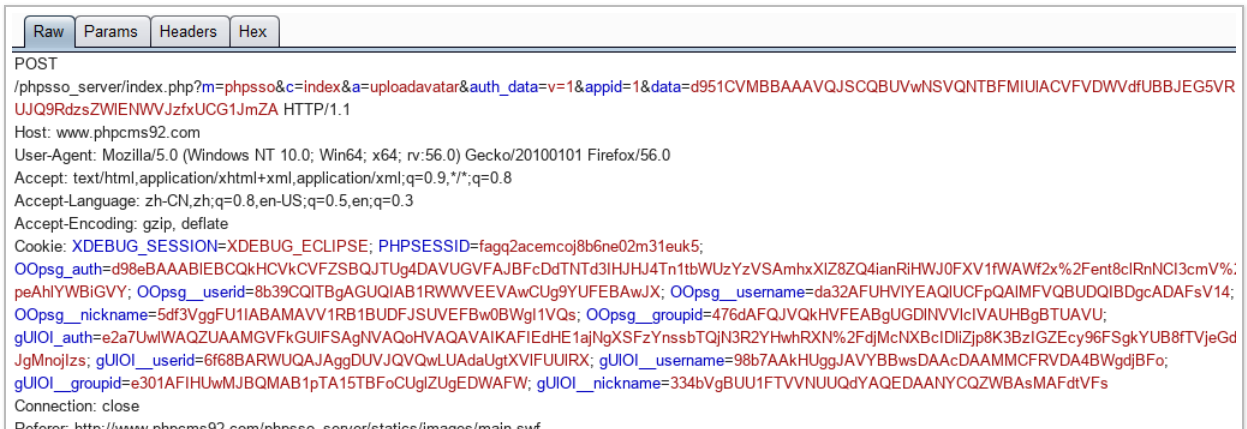
上传头像照片（Burp 抓包）–> 保存图片

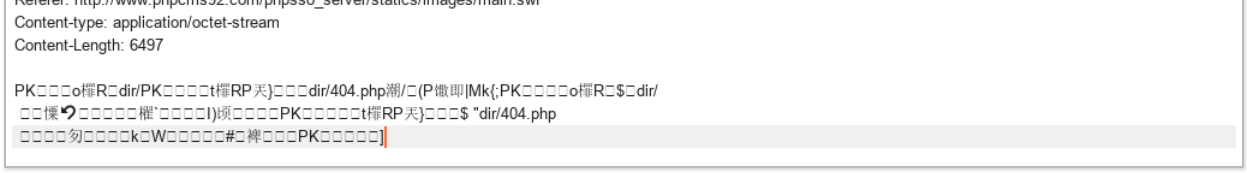
将之前的图片数据删除



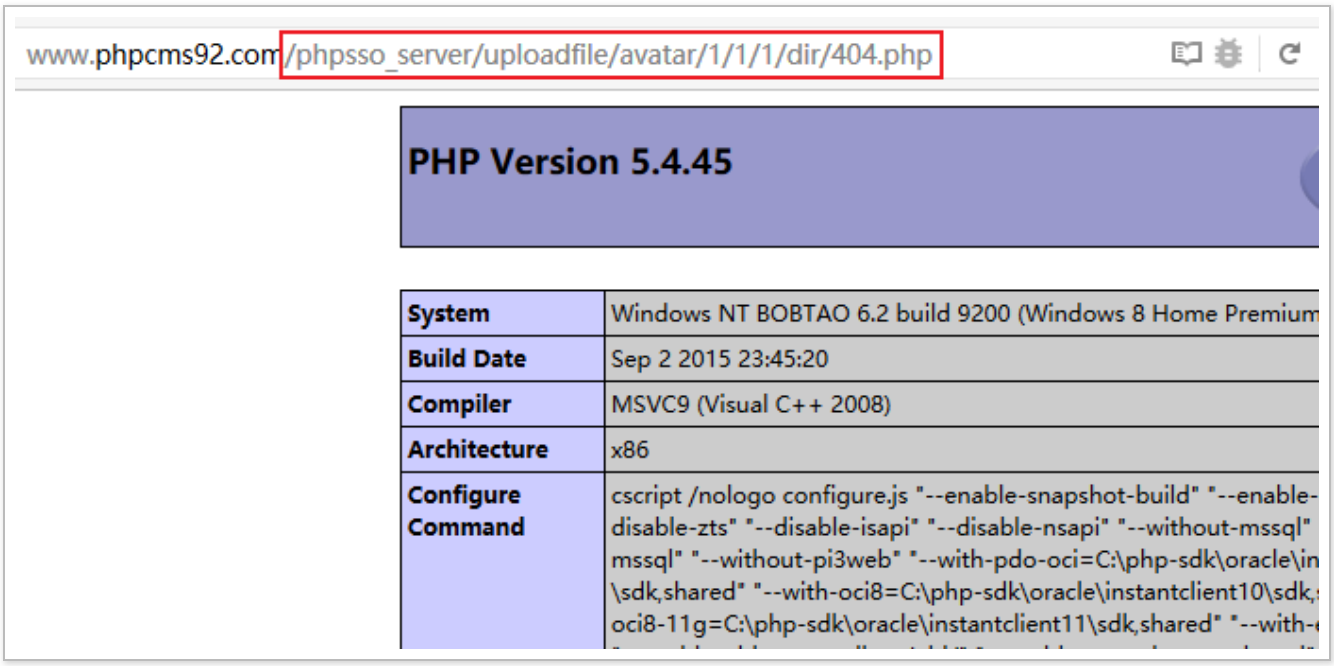


将 `Tao.zip` 中数据，按照上图的操作添加至请求中，最终效果如下图。然后放行





访问 `phpsso_server/uploadfile/avatar/1/1/1/dir/404.php` （这里的 `1` 是注册后用户的 id）



漏洞分析

在分析之前，我们先说一下漏洞存在处的功能，执行流程，以及漏洞产生的原因。

在编辑头像处，我们上传头像，前端会将我们上传的图片进行分割成三张 (三个尺寸大小)。然后前端打包压缩成 zip 数据，当我们保存图片时，我们的压缩包数据会上传到服务器，通过 `uploadavatar` 函数进行处理 (函数在文件 `phpsso_server/phpcms/modules/phpsso/index.php`)；而这个函数的执行流程就是：

1. 在保存上传头像文件夹处，创建一个跟用户 id 对应的文件夹
2. 将前端打包的压缩包通过 post 传来的数据进行保存，保存名为用户 id 的 zip 文件
3. 解压数据包
4. 判断未在数组内文件名命名的文件，不是则通过 `unlink` 函数遍历删除

上面流程存在问题的地方有，1. 未对压缩包内容进行处理，2. 解压遍历删除使用的是 `unlink` 函数，这个函数只能删除文件，不能删除文件夹。因为这一原因，我们只需将压缩包文件里带一个目录，目录里带恶意文件，即可绕过。

unlink

(PHP 4, PHP 5, PHP 7)
unlink — 删除文件

说明

```
unlink ( string  $filename  [, resource  $context  ] ) : bool
```

删除 **filename** 。和 Unix C 的 unlink() 函数相似。 发生错误时会产生一个 **E_WARNING** 级别的错误。

参数

filename
文件的路径。

context

Note: 在 PHP 5.0.0 中增加了对上下文（Context）的支持。有关上下文 *(Context)* 的说明参见 Streams。

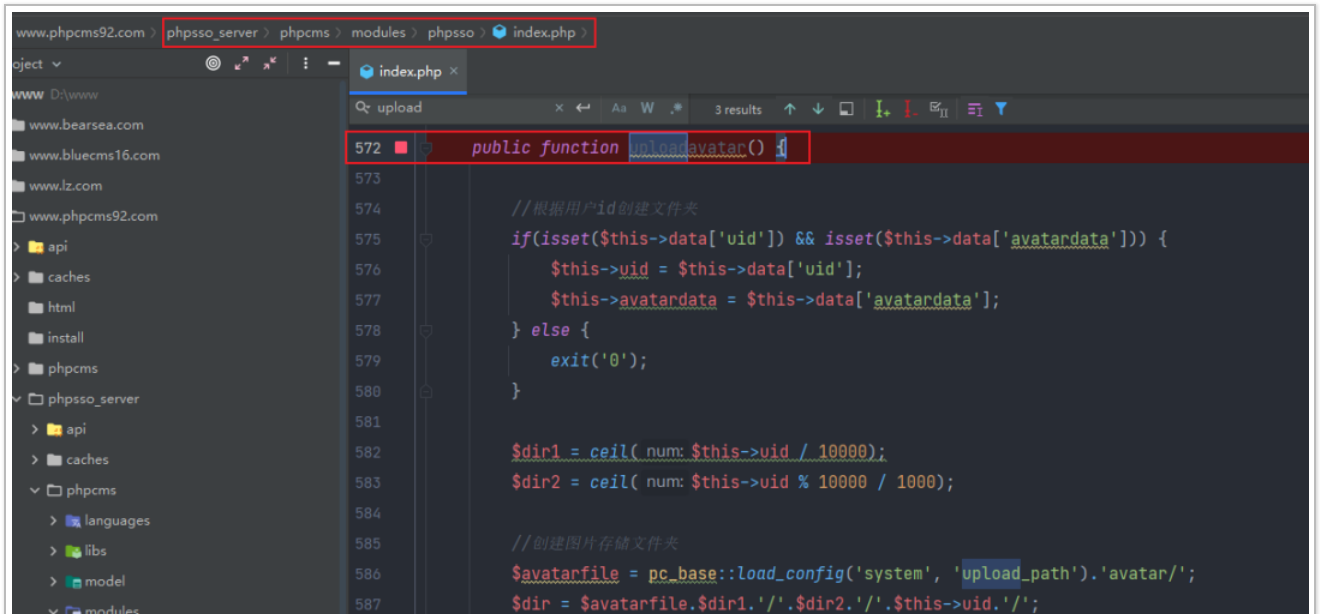
返回值

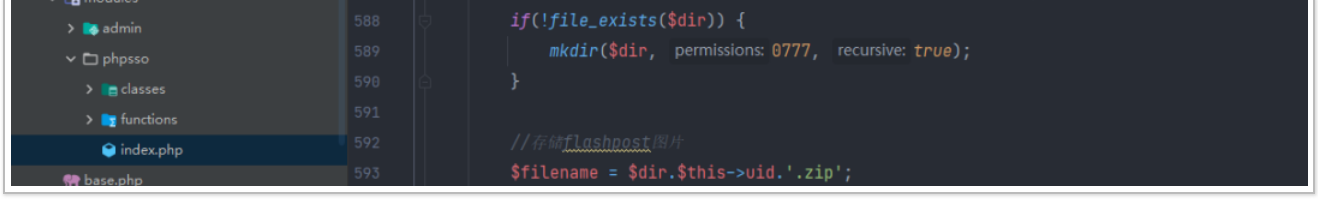
成功时返回 **TRUE** ， 或者在失败时返回 **FALSE** 。

图片处理请求为 `/phpsso_server/index.php?m=phpsso&c=index&a=uploadavatar`

定位文件 `phpsso_server/phpcms/modules/phpsso/index.php` 572 行

> 为什么定位到这，开头介绍有说

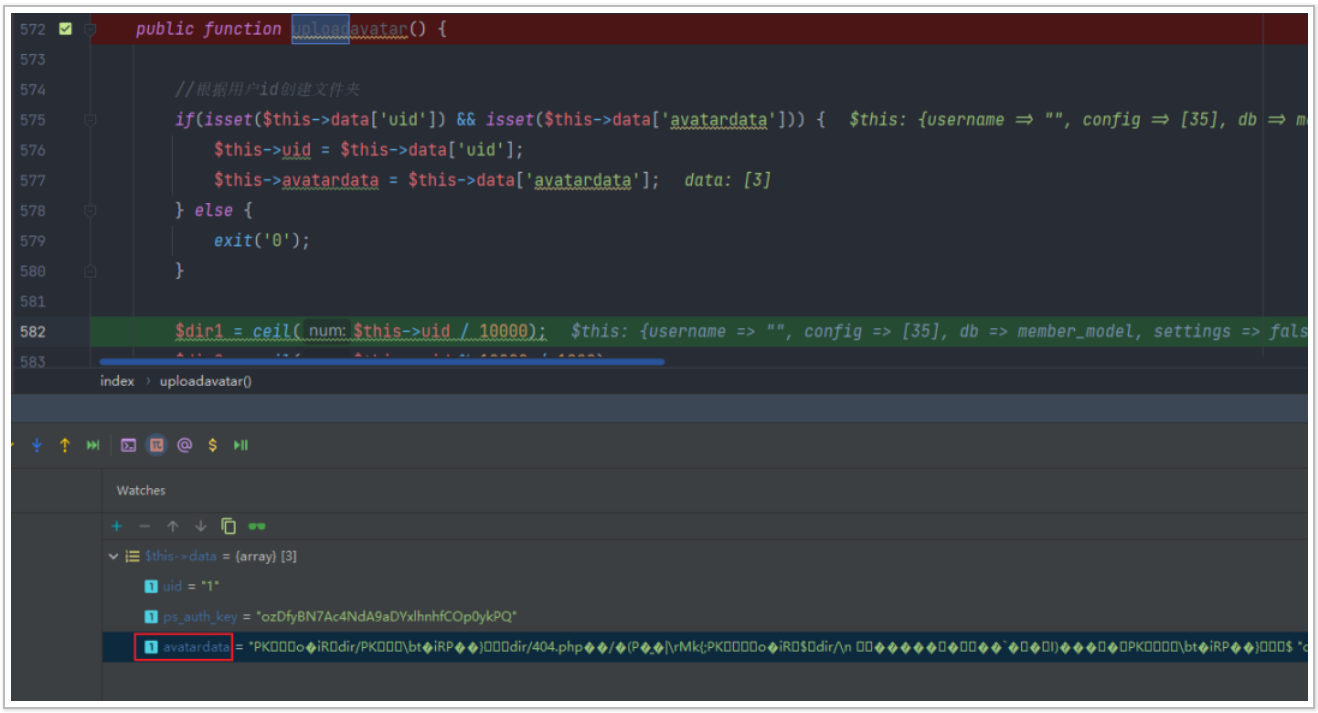




调试，向下执行

```
public function uploadavatar() {

//根据用户id创建文件夹
if(isset($this->data['uid']) && isset($this->data['avatardata'])) {
$this->uid = $this->data['uid'];
$this->avatardata = $this->data['avatardata'];
} else {
exit('0');
}
```

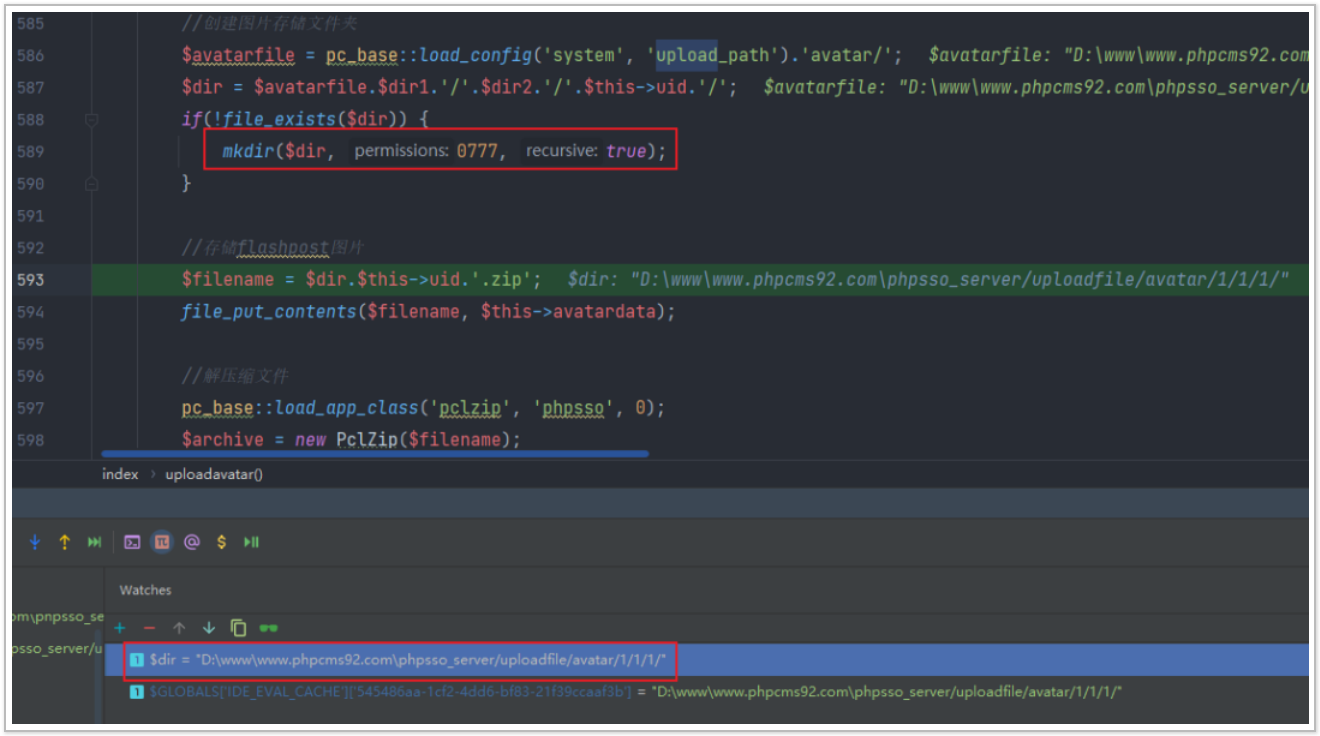


可以发现 `$this->data['avatardata']` 变量存储着我们上传修改的数据（恶意）

而 `$this->data['avatardata']` 是通过伪协议获取的（文件为 phpsso_server/phpcms/modules/phpsso/classes/phpsso.class.php），具体代码如下：

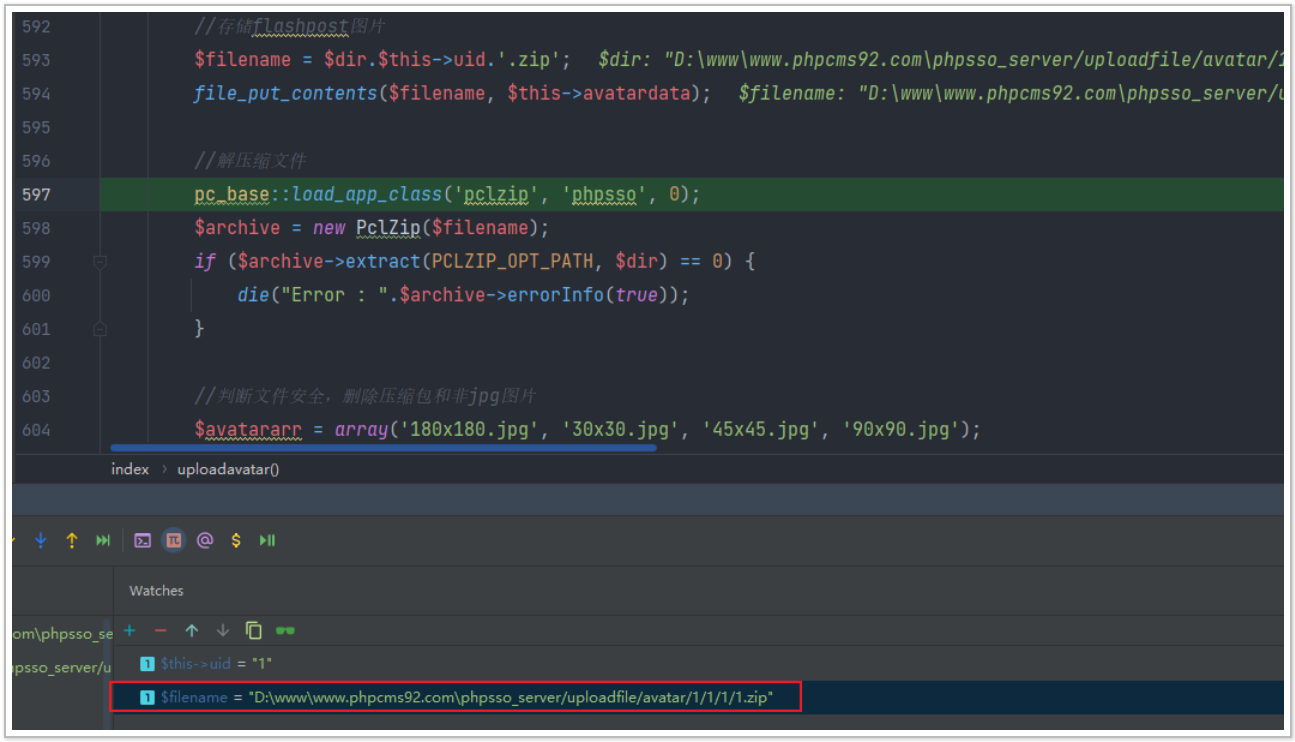
```
$postStr = file_get_contents("php://input");
if($postStr) {
$this->data['avatardata'] = $postStr;
}
```

继续向下走，新建存放图片目录

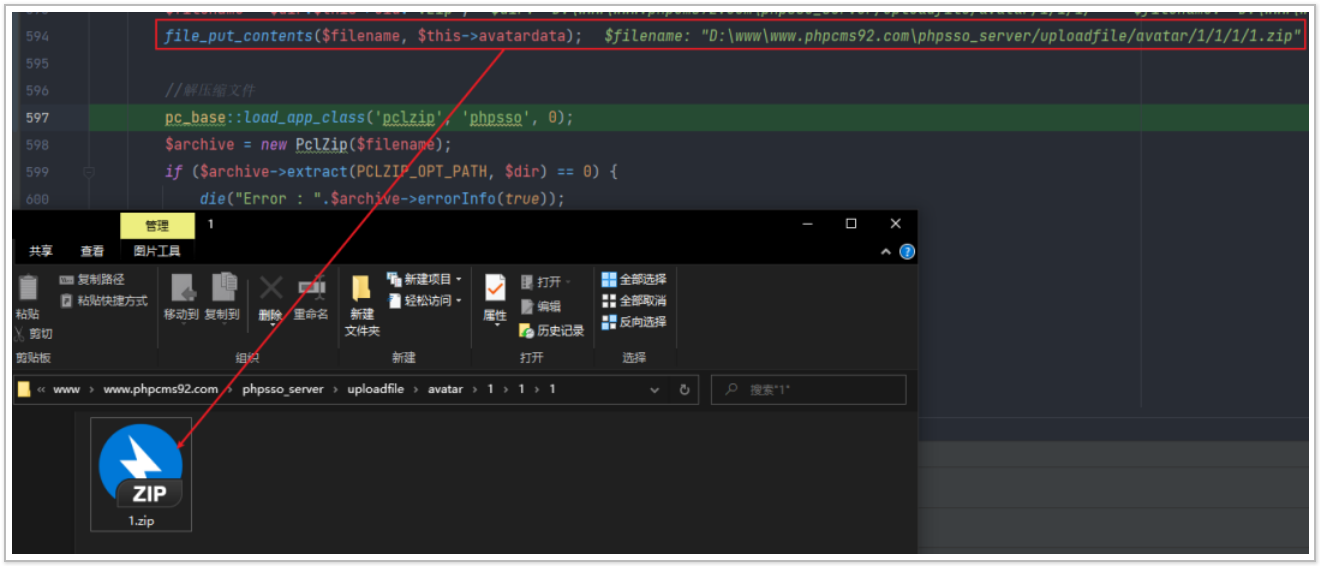



```
//创建图片存储文件夹
$avatarfile = pc_base::load_config('system', 'upload_path').'avatar/';
$dir = $avatarfile.$dir1.'/'.$dir2.'/'.$this->uid.'/';
if(!file_exists($dir)) {
    mkdir($dir, 0777, true);
}
$filename = $dir.$this->uid.'.zip';
file_put_contents($filename, $this->avatardata);
```

上面代码第五行创建目录。之后进行新命名压缩包，名为用户 id 值。然后将我们上面通过伪协议获取的数据进行写入

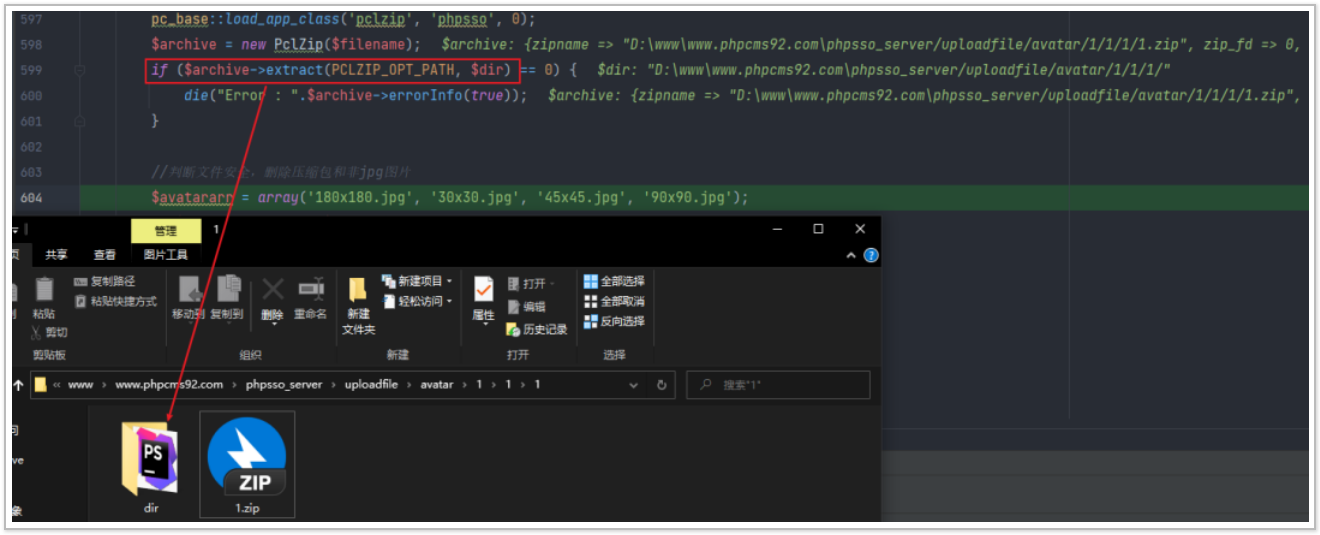


如下图，可以发现，新建了 `1.zip`



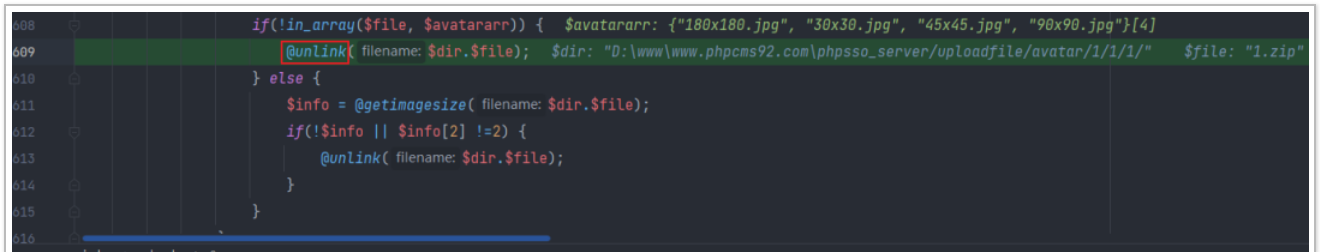
压缩包内容如下，就是我们修改上传的数据

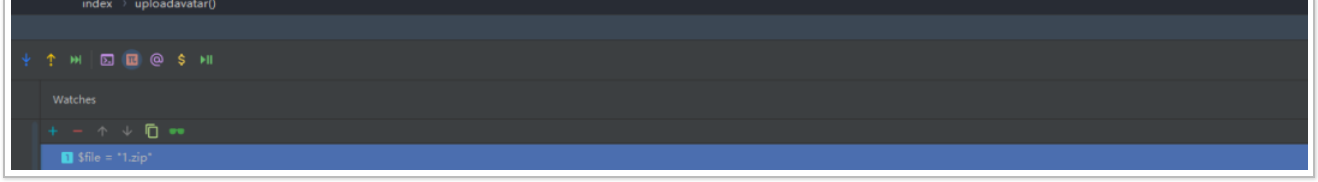
之后解压缩。。。



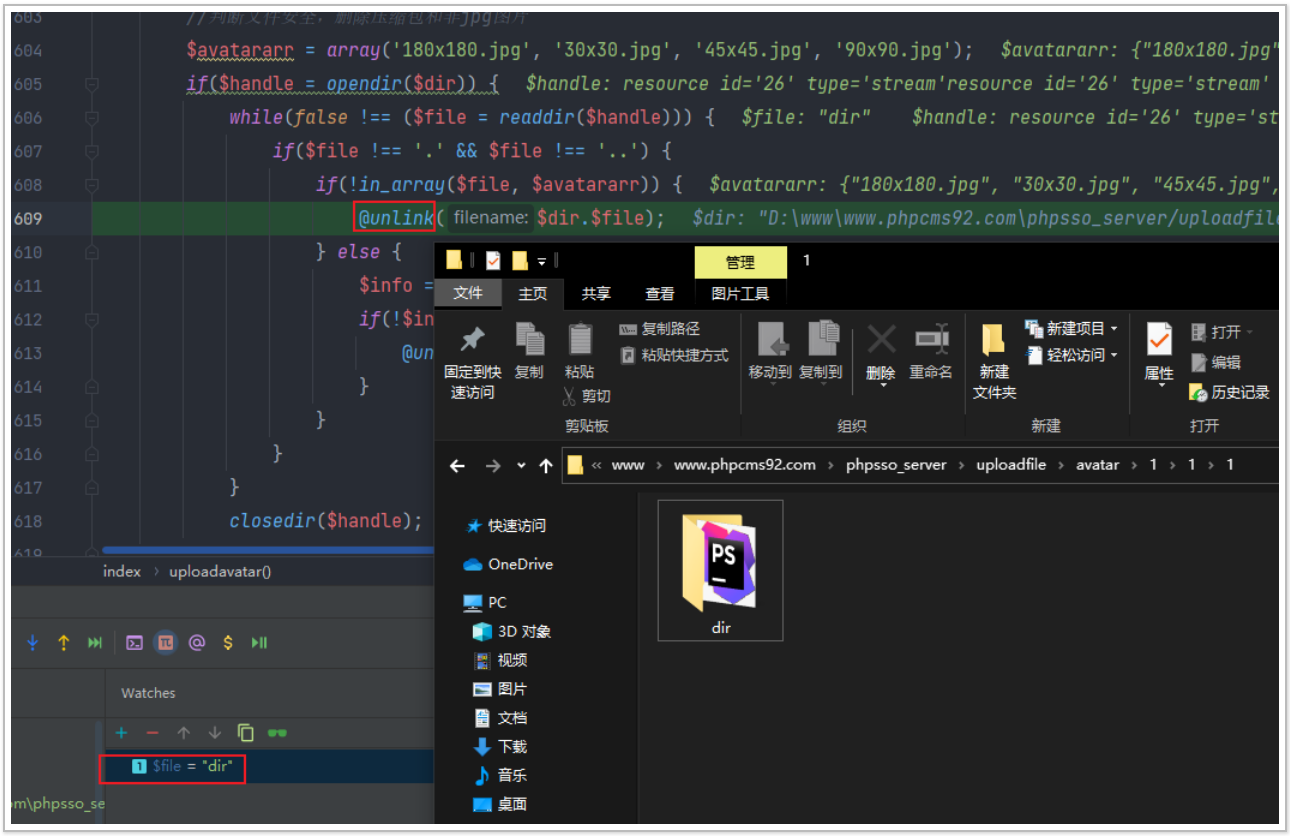
走到遍历白名单判断文件，排除 `.` （当前目录） `..` （上级目录）

下图删除了压缩包文件

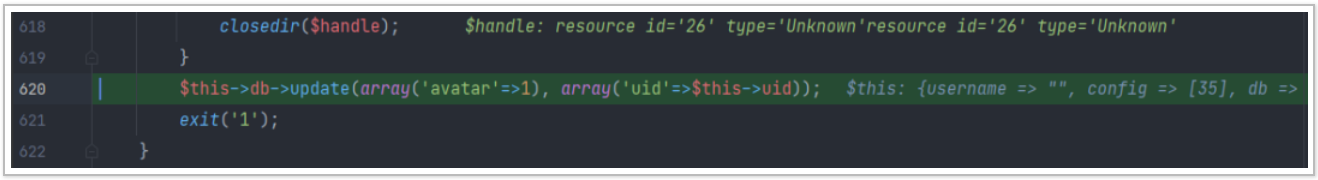




继续执行，当判断到 `dir` 目录时，因为 `dir` 目录不属于数组里（白名单），然后执行 `unlink(dir目录)` 。由于 `unlink` 函数只能删除文件，无法删除文件夹，所以就留下了恶意代码文件。



接着跳出了 if 语句，继续执行，将信息更新至数据库



所以，漏洞产生的原因就是 `unlink` 函数

```
if(!in_array($file, $avatararr)) {
@unlink($dir.$file);// 漏洞产生的原因
```

因为 `unlink` 无法删除文件夹，这就是为什么上面利用的压缩包里的恶意代码文件需要放在目录下

漏洞修复

- 不使用 zip 压缩包处理图片文件
- 使用最新版的 phpcms

文章中有什么不足和错误的地方还望师傅们指正。