

其他

文件解压引发的Getshell

安全狗safedog 2020-04-06 10:00:39 528624

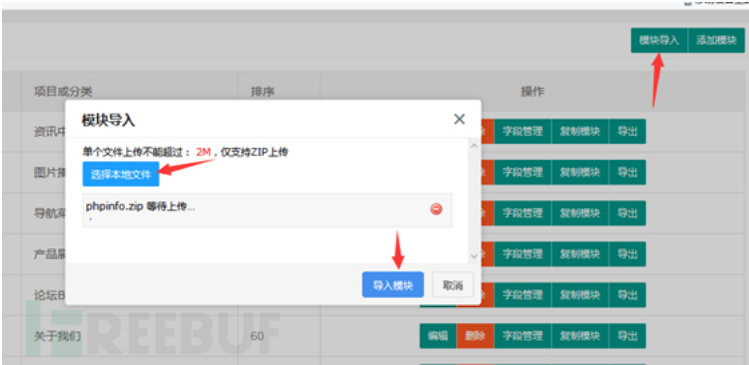
声明

本文仅供学习和研究，由于传播、利用此文所提供的信息而造成的任何直接或者间接的后果及损失，均由使用者本人负责，海青实验室及文章作者不承担任何责任。
安全狗海青实验室拥有此文章的修改和解释权，如欲转载或传播，必须保证此文的完整性，包括版权声明在内的全部内容，未经海青实验室同意，不得任意修改或者增减此文章内容，不得以任何方式将其用于商业目的。

攻击者可以快速的从文件上传功能点获得一个网站服务器的权限，所以一直是红蓝对抗中的“兵家必争之地”。而随着对抗不断升级，开发人员的安全意识的不断提高，多多少少也学会了使用黑名单或者白名单等一些手段进行防御，但刁钻的攻击者，依然发现了疏忽之处。本文以两个实例，简单介绍文件解压功能导致的getshell问题。

PHPOK CMS后台任意文件上传

首先，准备一个zip文件里，里面包含了一个PHP文件。然后，在导入模块中将zip文件上传。



在尝试的时候发现，导入模块失败了，但是查看文件夹内容，可以发现文件已经成功被写入了。



安全狗safedog
安全狗官方账号

关注

406

文章数

信创环境下高级威胁：信息化负责人该如何增
2023-11-17
教育案例分享 | 安全校提升立体化纵深防
2023-11-16
聚焦云原生安全|如何业互联网应用筑牢安
2023-09-28

浏览更多

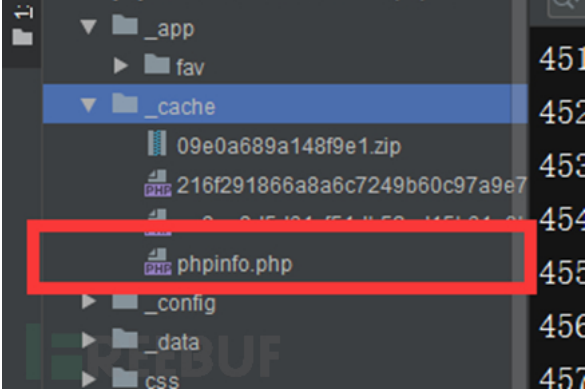
文章目录

- 声明
- PHPOK CMS后台
- 流程分析
- Jspxcms后台的zip
- 流程分析
- 总结

其他

26

6



流程分析

从代码层面来看看整个流程是怎样的？

状态	方法	地址	请求头	类型
成功	POST	127.0.0.1	admin.php?c=upload&f=zip&PHPSESSID=756d88tgmvg6jap187r9m4b	html
成功	GET	127.0.0.1	admin.php?c=module&f=import&zipfile=_cache/6b03c2be125afad.zip&157060294...	json

当我们点击导入模块时，浏览器会发送两条请求。先看第一条POST请求。根据PHPOK CMS路由分发的规则，可以定位到upload控制器中的zip方法。

```
38  /**
39   * 接收ZIP包上传，主要用于更新及数据导入，上传的表单id固定用upfile
40   */
41   public function zip_f()
42   {
43     $rs = $this->lib('upload')->zipfile('upfile');
44     if($rs['status'] != 'ok'){
45       $this->json($rs['error']);
46     }
47     $this->json($rs['filename'], true);
48   }
49
```

而在43行处，引入了libs\upload.php文件并且调用zipfile()函数。

```
123  /**
124   * public function zipfile($input, $folder='')
125   {
126     if(!$input){
127       //如果未指定存储文件夹，则使用
128     }
129     if(!$folder){
130       $folder = $this->dir_cache;
131     }
132     $this->cateid = 0;
133     $this->set_dir($folder);
134     $this->set_type('zip');
135     $this->cate = array('id'=>0, 'filemax'=>104857600, 'root'=>$folder, 'folder'=>"/", 'filetypes'=>'zip');
136     if(isset($FILES[$input])){
137       $rs = $this->save($input);
138     }
139     if($rs['status'] != 'ok'){
140       return $rs;
141     }
142     $rs['cate'] = $this->cate;
143     return $rs;
144   }
145
```

在第135行处，设定了文件后缀类型为zip,由于137行处的if条件不成立，流程进入140行，调用_save()方法。继续跟进_save()方法。



主站

公开课

商城

用户服务

行业服务

知识大陆

FVIP
资源搜索

其他

26

6

```

268     global $app;
269     $basename = substr(md5(strftime('%Y%m%d%H%M%S', time())), 0, 16);
270     $tmpname = $app->get('id', 'name');
271     $tmpname = $app->lib('class', 'string')->md5($tmpname);
272     $tmpname = $app->format($tmpname); //安全格式化数据
273     if (!$tmpname) {
274         return array('status' => 'error', 'error' => P_Lang('附件类型不符合要求'));
275     }
276     $ext = $this->file_ext($tmpname);
277     if (!$ext) {
278         return array('status' => 'error', 'error' => P_Lang('附件类型不符合要求'));
279     }
280     $chunk = $app->get('id', 'chunk', 'type', 'int');
281     $chunks = $app->get('id', 'chunks', 'type', 'int');
282     if (!$chunks) {
283         return array('status' => 'error', 'error' => P_Lang('无法打开输出流'));
284     }
285     $tmpid = 's_' . md5($tmpname);
286     $out_tmpfile = $this->dir_cache.$tmpid.'_'.$chunk;
287     if (!$out = @fopen($filename.$out_tmpfile.'.parttmp', 'mode' . $mode)) {
288         return array('status' => 'error', 'error' => P_Lang('无法打开输出流'));
289     }
290     if (!$in = @fopen($filename.'php://input', 'mode' . $mode)) {
291         return array('status' => 'error', 'error' => P_Lang('无法打开输入流'));
292     }

```

_save()方法在276行处使用file_ext()方法对传入的name参数也就是文件名进行后缀判断，而如果判断通过，则会在之后将压缩文件保存下来。跟进一步file_ext()方法。

```

177     private function file_ext($tmpname)
178     {
179         $ext = pathinfo($tmpname, options: PATHINFO_EXTENSION);
180         if (!$ext) {
181             return false;
182         }
183         $ext = strtolower($ext);
184         $filetypes = "jpg,gif,png";
185         if ($this->cate && $this->cate['filetypes']) {
186             $filetypes .= "," . $this->cate['filetypes'];
187         }
188         if ($this->file_type) {
189             $filetypes .= "," . $this->file_type;
190         }
191         $list = explode(' ', $filetypes);
192         $list = array_unique($list);
193         if (!in_array($ext, $list)) {
194             return false;
195         }
196         return $ext;
197     }

```

在该方法中，设置了一个白名单，文件名只能是jpg,gif,png,zip中的其中一种。为什么包含zip呢？不管是程序执行到第186或者189行处，在前面就已经给这两个变量设置值为zip了，所以在白名单中自然包含了zip。目前到此为止，整个上传流程没有什么问题，并且使用了白名单来限制后缀，所以只能上传zip压缩文件。接着再来看第二处请求

admin.php?c=module&f=import&zipfile=_cache%2Fa9414ae41044fc5a.zip&_=1570603538199

同样根据路由规则，可以定位到module控制器下的import方法

```

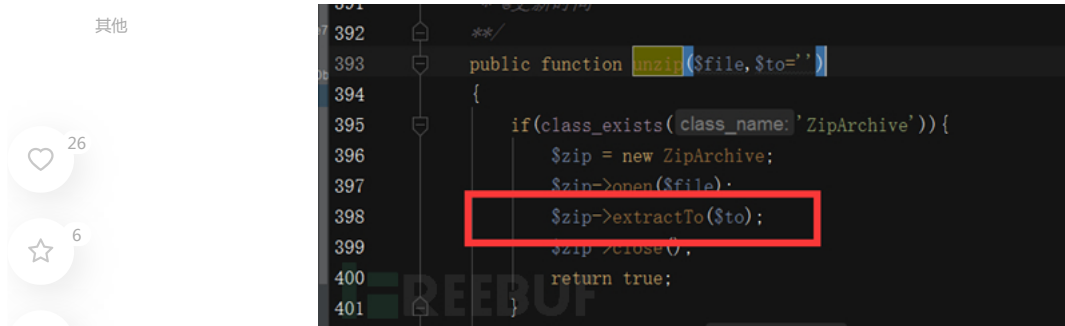
694     public function import_f()
695     {
696         $zipfile = $this->get('zipfile');
697         if (!$zipfile) {
698             $this->lib('form')->cssjs(array('form_type' => 'upload'));
699             $this->addjs('js/webuploader/admin.upload.js');
700             $this->view('module/import');
701         }
702         if (strpos($zipfile, '..') !== false) {
703             return array('status' => 'error', 'error' => P_Lang('导入模块失败，请检查解压是否成功'));
704         }
705         if (!file_exists($filename.$this->dir_root.$zipfile)) {
706             $this->lib('phpzip')->unzip($this->dir_root.$zipfile, $this->dir_cache);
707         }
708         if (!file_exists($filename.$this->dir_cache.'module.xml')) {
709             $this->lib('xml')->load($this->dir_cache.'module.xml', true);
710             if (!$rs) {
711                 $this->lib('xml')->load($this->dir_cache.'module.xml', true);
712             }
713             $tmp = $rs;
714             if (isset($tmp['_fields'])) {
715                 unset($tmp['_fields']);
716             }
717         }
718     }

```

文件解压引发的Getshell

安全狗S

函数，程序实例化了ZipArchive类进行解压，而目标路径\$to则是缓存文件夹。

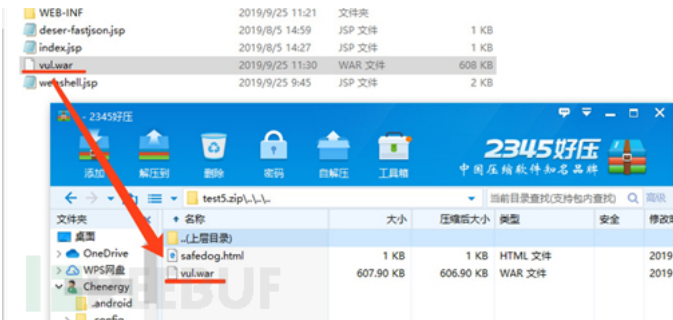


到此不难看出问题，程序将压缩包中的文件解压出来之后，并未进一步的进行判断，以至于里面包含的 PHP文件可以绕过上传文件的限制，使得原本建立的安全防御土崩瓦解。

Jspxcms后台的zip解压功能目录穿越漏洞导致getshell

由于这套CMS做了相关防御配置，压缩包里面像上文中直接加入JSP文件是无法执行的，会报403错误。因而想到使用目录穿越的方式，跳出JspxCM的根目录，并根据war包会自动解压的特点，从而getshell。

首先，建立一个恶意的war包,并且打成压缩包，如图所示。



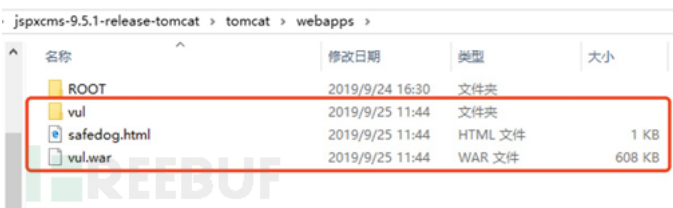
然后点击“上传文件”按钮，上传test5.zip，如图所示。



紧接的使用解压功能，将上传的压缩包解压出来。点击“ZIP解压按钮”，如图所示。



此时，在服务器端查看webapps目录的变化，可以发现safedog.html和vul.war文件被解压到了网站根目录“webapps/ROOT”之外，如图所示。



访问上传的webshell，效果如下。

<http://192.168.114.132:8080/vul/webshell.jsp?pwd=023&cmd=calc>
文件解压引发的Getshell

FreeBuf

主站

公开课

商城

用户服务

行业服务

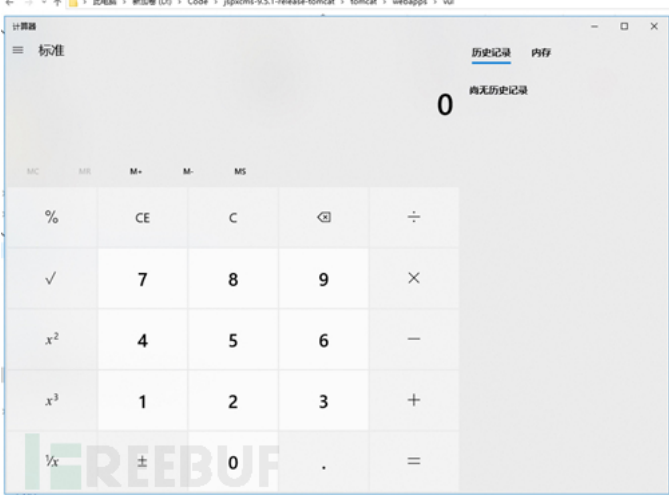
知识大陆

FVIP资源搜索

其他

26

6

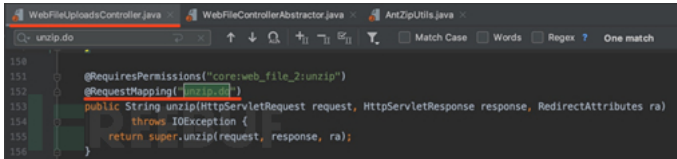


流程分析

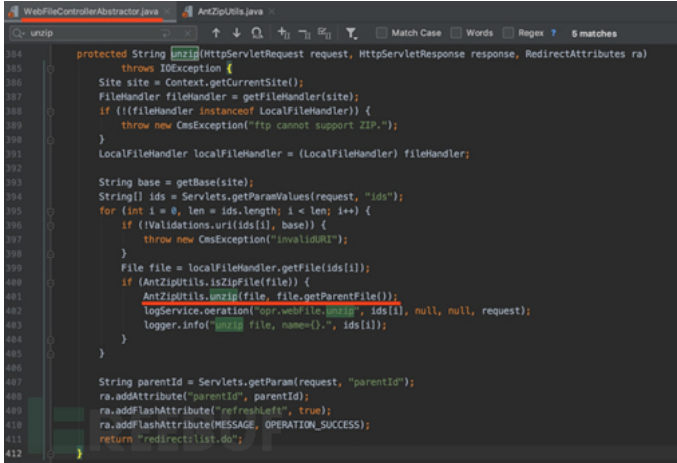
在后台登录之后，找到ZIP解压功能，通过BurpSuite进行抓包可以发现“解压文件”的接口调用情况，如图所示。



该接口对应了jspxcms-9.5.1-release-src/src/main/java/com/jspxcms/core/web/back/WebFileUploadsController.java的unzip方法，如图所示。



对unzip方法进行跟进，发现它的具体实现在jspxcms-9.5.1-release-src/src/main/java/com/jspxcms/core/web/back/WebFileControllerAbstractor.java中。可以发现，在对zip文件进行解压的时候，调用了AntZipUtil类的unzip方法，如图所示。



其他

26

6

```
AntZipUtils.java
122 public static void unzip(File zipFile, File destDir, String encoding) {
123     if (destDir.exists() && !destDir.isDirectory()) {
124         throw new IllegalArgumentException("destDir is not a directory!");
125     }
126     ZipFile zip = null;
127     InputStream is = null;
128     FileOutputStream fos = null;
129     File file;
130     String name;
131     byte[] buff = new byte[DEFAULT_BUFFER_SIZE];
132     int readed;
133     ZipEntry entry;
134     try {
135         try {
136             if (StringUtils.isNotBlank(encoding)) {
137                 zip = new ZipFile(zipFile, encoding);
138             } else {
139                 zip = new ZipFile(zipFile);
140             }
141             Enumeration<?> en = zip.getEntries();
142             while (en.hasMoreElements()) {
143                 entry = (ZipEntry) en.nextElement();
144                 name = entry.getName();
145                 name = name.replace('/', File.separatorChar);
146                 file = new File(destDir, name);
147                 if (entry.isDirectory()) {
148                     file.mkdirs();
149                 } else {
150                     // 创建父目录
151                     file.getParentFile().mkdirs();
152                     is = zip.getInputStream(entry);
153                     fos = new FileOutputStream(file);
154                     while ((readed = is.read(buff)) > 0) {
155                         fos.write(buff, 0, readed);
156                     }
157                     fos.close();
158                     is.close();
159                 }
160             }
161         } catch (IOException e) {
162             e.printStackTrace();
163         }
164     } catch (Exception e) {
165         e.printStackTrace();
166     }
167 }
```

总结

从上述两则实例来看，压缩包里包含着webshell颇有些特洛伊木马的味道。当程序对上传后缀限制的相当严格，例如上文提到的PHPOK CMS，已经使用白名单的机制只能上传四种后缀的文件，却依然因为对压缩文件的处理不当，导致整个防御机制的失效。在ZZZCMS中的实例，猜测开发者的本意是zip文件夹内的内容应该别可控的，但在目录穿越的加持下，超出了开发者的预期从而防御机制土崩瓦解。

那么如何防御呢？建议开发者在实现文件解压功能时考虑以下要点：

- 1) 限制文件的扩展名（如采用白名单的方式）；
- 2) 限制文件的名称（以较为严谨的黑名单约束文件名）；
- 3) 限制文件的大小（以免遭受压缩包**的DDoS攻击）。

因此，在开发的各个环节，都要将安全意识贯彻其中。千里之堤毁于蚁穴，也正是这种细微之处的小问题，才使得攻击者有机可乘。

*本文作者：安全狗safedog，转载请注明来自FreeBuf.COM

本文作者：安全狗safedog，转载请注明来自FreeBuf.COM

Getshell

实例分析

文件解压

被以下专辑收录，发现更多精彩内容

+ 收入我的专辑

+ 加入我的收藏

JAVA代码审计

实战

相关推荐

记一次任意文件下载到Getshell 原创