Tunnel Interfaces

Se você deseja estabelecer um túnel SSH entre duas redes e tratar o túnel como uma interface, isso pode ajudar.

- Primeiro, eu recomendaria ativar o rc.ip_forward no /etc/rc.d/ (nas máquinas local e de destino)

```
root@darkstar:~# chmod +x /etc/rc.d/rc.ip_forward
```

or for a temporary (lose at reboot) way use

```
root@darkstar:~# echo "1" > /proc/sys/net/ipv4/ip_forward
```

- Quase esqueci que você precisará ativar o tunelamento em /etc/ssh/sshd_conf. Encontre esta parte *#PermitTunnelno* remova o comentário e mude para yes (na máquina de destino), você pode usar o vi para editar ou tentar o código abaixo:

```
root@darkstar:~# /usr/bin/sudo /bin/sed -e "s/#PermitTunnel\
no/PermitTunnel\ yes\ #changed\ `date '+%Y%m%d' \
`\ by\ `/bin/whoami`/" -i.stock_slackware-`/bin/awk '{print $2}'
/etc/slackware-version` /etc/ssh/sshd_config
```

- Em seguida, estou usando o autossh encontrado em slackbuilds.org . O script abaixo exige isso (há um método alternativo mencionado na parte inferior. Substitua a linha 20.)
- Você também desejará ter conexões ssh não interativas configuradas. Significado authorized_keys configuração com chaves públicas/privadas.
- Supondo também que você tenha privilégios de sudo e usuário na máquina remota (usando a opção `NOPASSWD: ALL` em `/etc/sudoers`))
  Exemplo: como tipo de raiz “`visudo`” e adicione o seu usuário assim

```
rich ALL=(ALL) NOPASSWD: ALL
```

- Em seguida, estou usando um script bash simples que explicarei depois de espreitar.

rc.tunnel

```
#!/bin/bash

############################################################
#
#Enter the ip of the target you wish to make a tunnel with.
#By ip address or hostname
#
target=74.79.121.210
port=22
#
############################################################
# suggestions contact rich at lehcar.duckdns.org
```

```
# with thanks to Billy T (for idea and assistance)
############################################################
#load module
/usr/bin/sudo /sbin/modprobe tun
#load remote module
/usr/bin/ssh -p $port $target "/usr/bin/sudo /sbin/modprobe tun"
/bin/sleep 1

/usr/bin/sudo /usr/bin/autossh -p $port -M 0 -o "ServerAliveInterval
60" -o "ServerAliveCountMax 3" -fw 0:0 $target /bin/true
/bin/sleep 4
/usr/bin/ssh -p $port $target "/usr/bin/sudo /sbin/ifconfig tun0
192.168.5.2 pointopoint 192.168.5.1 netmask 255.255.255.252 broadcast
192.168.5.3"
/usr/bin/sudo /sbin/ifconfig tun0 192.168.5.1 pointopoint 192.168.5.2
netmask 255.255.255.252 broadcast 192.168.5.3
/usr/bin/sudo /usr/sbin/iptables -t nat -A POSTROUTING -o eth0 -j
MASQUERADE
/usr/bin/sudo /usr/sbin/iptables -A FORWARD -i eth0 -o tun0 -m state --
state RELATED,ESTABLISHED -j ACCEPT
/usr/bin/sudo /usr/sbin/iptables -A FORWARD -i tun0 -o eth0 -j ACCEPT

/usr/bin/ssh -p $port $target "/usr/bin/sudo /usr/sbin/iptables -t nat
-A POSTROUTING -o eth0 -j MASQUERADE"
/usr/bin/ssh -p $port $target "/usr/bin/sudo /usr/sbin/iptables -A
FORWARD -i eth0 -o tun0 -m state --state RELATED,ESTABLISHED -j ACCEPT"
/usr/bin/ssh -p $port $target "/usr/bin/sudo /usr/sbin/iptables -A
FORWARD -i tun0 -o eth0 -j ACCEPT"
```

- Para direcionar o tráfego pelo túnel, tente:

```
user@darkstar:~$ sudo /sbin/route add -net 74.125.131.0 netmask
255.255.255.0 dev tun0
```

- Para substituir o ssh normal em vez de autossh, substitua esta linha abaixo pela linha com autossh (linha 20)

```
/usr/bin/sudo /usr/bin/ssh -p $port -o "ServerAliveInterval 60" -o
"ServerAliveCountMax 3" -fw 0:0 $target /bin/true
```

Para recapitular algumas das opções: "`-fw 0: 0`" bifurca o processo ssh em segundo plano/abre um túnel e "`0: 0`" escolhe as interfaces local e remota (ou seja, tun0 e tun0) As opções "`-o`" são usadas para especificar parâmetros para o openSSH.

Resultados

Agora você pode acessar o computador remoto usando "`192.168.5.2`" e suas comunicações passarão pelo túnel ssh. No script acima, você pode modificar a rede e a máscara para atender às

suas necessidades. Eu escolhi “192.168.5”, mas sua configuração provavelmente será diferente.

Não vou conseguir elaborar as regras do iptables sou um newbie nesta questão, mas elas funcionam.

Aqui estão alguns exemplos de uso. Direcionando o tráfego para a interface do túnel. primeiro exemplo 10.10.132.0-255 segundo 10.10.182.15

```
/sbin/route add -net 10.10.132.0 netmask 255.255.255.0 dev tun0
/sbin/route add -net 10.10.182.15 netmask 255.255.255.255 dev tun0
```

Sources

- Originally written by ricky_cardo

- — *slackjeff 2020/06/23 18:31 (BRT)*

howtos, tunnel, tun0, author ricky cardo