

Tunnel Interfaces

If you are looking to establish an SSH tunnel between two networks and treat the tunnel as an interface, this may help.

- First I would recommend enabling `rc.ip_forward` in `/etc/rc.d/` (on local and target machines)

```
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
```

- Almost forgot you will need to enable tunneling in `/etc/ssh/sshd_conf`. Find this part `#PermitTunnel no` uncomment and change to `yes` (on the target machine) you can use `vi` to edit or try the below code:

```
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
```

- Next I'm using `autossh` found at slackbuilds.org. The below script requires it (there is an alternate method mentioned at the bottom. Replace line 20.)
- You will also want to have non-interactive ssh connections set up. Meaning [authorized_keys setup with public/private keys](#).
- Also assuming you have `sudo` privileges and user on the remote machine (using the `NOPASSWD: ALL` option in `/etc/sudoers`)
Example: as root type "visudo" and add your user like this

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

- Next I'm using a simple bash script which I will explain after peeking at it.

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"
```

- To direct traffic over the tunnel try:

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

- To substitute normal ssh rather than autossh substitute this line below for the one with autossh (line 20)

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

To recap some of the options: “-fw 0:0” forks the ssh process to the background / opens a tunnel and “0:0” picks the local and remote interfaces (ie tun0 and tun0). The “-o” options are used to specify parameters for openSSH.

Results

Now you can access the remote computer using “192.168.5.2” and your communications will go through the ssh tunnel. In the above script you may wish to modify the network and mask to fit your needs. I chose “192.168.5” but your configuration will likely be different.

I'm not going to be able to elaborate on the iptables rules still a little green there, but they work.

Here are a couple usage examples. Directing traffic to the tunnel interface. first example 10.10.132.0-255 second just 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](#)

[howtos](#), [tunnel](#), [tun0](#), author [ricky cardo](#)

From:
<https://docs.slackware.com/> - **SlackDocs**

Permanent link:
https://docs.slackware.com/howtos:network_services:tunnel_interfaces

Last update: **2020/06/25 13:41 (UTC)**

