VPN over SSH

There are several ways to set up a Virtual Private Network through SSH. Note that, while this may be useful from time to time, it may not be a full replacement for a regular VPN. See for example **[1] (http://sites.inka.de/bigred/devel/tcp-tcp.html)**.

Contents

OpenSSH's built in tunneling

OpenSSH has built-in TUN/TAP support using -w<local-tun-number>:<remote-tun-number>. Here, a layer 3/point-to-point/ TUN tunnel is described. It is also possible to create a layer 2/ethernet/TAP tunnel.

Create tun interfaces using netcfg

Create tun interfaces:

```
$ cat /etc/network.d/vpn
INTERFACE='tun5'
CONNECTION='tuntap'
MODE='tun'
USER='vpn'
GROUP='network'

IP='static'
SKIPNOCARRIER='yes'
ADDR='<IP>'
IPCFG=('ip route add <REMOTE-NETWORK/MASK> via <REMOTE-SIDE-IP>')
```

Then do 'netcfg -u vpn' or add it into /etc/conf.d/netcfg.

Create tun interfaces using systemd-networkd

```
$ cat /etc/systemd/network/vpn.netdev
[NetDev]
Name=tun5
Kind=tun
```

```
[Tun]
User=vpn
Group=network

$ cat /etc/systemd/network/vpn.network
[Match]
Name=tun5

[Address]
Address=192.168.200.2/24
```

Once these files are created, enable them by **restarting** `systemd-networkd.service`.

Also you may manage tun interfaces with 'ip tunnel' command.

Creating interfaces in SSH command

SSH can create both interfaces automatically, but you should configure IP and routing after the connection is established.

```
ssh \
  -o PermitLocalCommand=yes \
  -o LocalCommand="sudo ifconfig tun5 192.168.244.2 pointopoint 192.168.244.1 netmask 255.255.255.0" \
  -o ServerAliveInterval=60 \
  -w 5:5 vpn@example.com \
  'sudo ifconfig tun5 192.168.244.1 pointopoint 192.168.244.2 netmask 255.255.255.0; echo tun0 ready'
```

Start SSH

```
ssh -f -w5:5 vpn@example.com -i ~/.ssh/key "sleep 1000000000"
```

or you may add keep-alive options if you are behind a NAT.

```
ssh -f -w5:5 vpn@example.com \
       -o ServerAliveInterval=30 \
       -o ServerAliveCountMax=5 \
       -o TCPKeepAlive=yes \
       -i ~/.ssh/key "sleep 1000000000"
```

Troubleshooting

- ssh should have access rights to tun interface or permissions to create it. Check owner of tun interface and/or /dev/net/tun.
- Obviously if you want to access a network rather than a single machine you should properly set up IP packet forwarding, routing and maybe a netfilter on both sides.

Using PPP over SSH

pppd can easily be used to create a tunnel through an SSH server:

```
/usr/sbin/pppd updetach noauth silent nodeflate pty "/usr/bin/ssh
root@remote-gw /usr/sbin/pppd nodetach notty noauth" ipparam vpn
10.0.8.1:10.0.8.2
```

When the VPN is established, you can route traffic through it. To get access to an internal network:

```
ip route add 192.168.0.0/16 via 10.0.8.2
```

To route all Internet traffic through the tunnel, for example to protect your communication on an unencrypted network, first add a route to the SSH server through your regular gateway:

```
ip route add <remote-gw> via <current default gateway>
```

Next, replace the default route with the tunnel

```
ip route replace default via 10.0.8.2
```

Helper script

pvpn (https://github.com/halhen/pvpn) (available in **AUR** package **pvpn (https://aur.archlinux.org/packages/pvpn/)**[AUR]) is a wrapper script around `pppd` over SSH.

See also

- **Configuring Network**
- **Router**
- **Ssh**
- **sshuttle-git (https://aur.archlinux.org/packages/sshuttle-git/)**[AUR], a **python** tunnel

Retrieved from "https://wiki.archlinux.org/index.php?title=VPN_over_SSH&oldid=507920"