

{Network & Hardware layout}

First I'll describe my network setup and platform.

We have 9 private networks on the 192.168.x.x network, connected via a core router and 1 Mandrake Linux 9.2 server with 2 interfaces, a private 192.168.x.x address and a public address protected via IPTables, using FWBuilder 1.11 as a front end.

{Port setup}

I want to support up to 10 tunnels, so I opened up UDP Ports 5000 – 5010.

{Deciding on a network}

I decided to use the 192.168.200.x network for the VPN.

I asked the Network administrator to make a entry into the core router that points all requests for the 192.168.200.x network to the Linux box. All network routes are setup via office.up script on the Linux box and the office.up.bat batch file on the Windows (Road Warriors) side.

{Problems encountered}

My problem was not understanding that each tunnel had to be on it's own network. I took subnet as; I could assign each tunnel to an address like

192.168.200.1 (tap0 server) 192.168.200.10 (tap0 client)
192.168.200.2 (tap1 server) 192.168.200.11 (tap1 client)

As soon as tap1 was tried, it would time out until tap0 was terminated.

I finally figured out that the other taps had to be on completely different network. This was a problem with the Network administrator.

He suggested instead using a subnet mask of 255.255.255.240, this way he only had to make 1 entry into the router.

It worked.

{Doing the math}

If I've figured correctly, using the 240-subnet mask will allow the 192.168.240.x network to support 14 tunnels. I believe the table should be listed first, since it will make the config files understandable.

Network table will follow, remember each server/client end will need an IP address, I listed my choices to the right of each network listed. Also, this numbering scheme should work with any private address space, not just 192.168.240.x:

Tunnel	(01) 192.168.200.17 - 192.168.200.30	(200.17 – 200.18) (Server Side – Client Side)
	192.168.200.31 (Broadcast)	
	192.168.200.32 (NA)	
	(02) 192.168.200.33 - 192.168.200.46	(200.33 – 200.34)
	192.168.200.47 (Broadcast)	
	192.168.200.48 (NA)	
	(03) 192.168.200.49 – 192.168.200.62	(200.49 – 200.50)
	192.168.200.63 (Broadcast)	
	192.168.200.64 (NA)	
	(04) 192.168.200.65 – 192.168.200.78	(200.65 – 200.66)
	192.168.200.79 (Broadcast)	
	192.168.200.80 (NA)	
	(05) 192.168.200.81 – 192.168.200.94	(200.81 – 200.82)
	192.168.200.95 (Broadcast)	
	192.168.200.96 (NA)	
	(06) 192.168.200.97 – 192.168.200.110	(200.97 – 200.98)
	192.168.200.111 (Broadcast)	
	192.168.200.112 (NA)	
	(07) 192.168.200.113 – 192.168.200.126	(200.113 – 200.114)
	192.168.200.127 (Broadcast)	
	192.168.200.128 (NA)	
	(08) 192.168.200.129 – 192.168.200.142	(200.129 – 200.130)
	192.168.200.143 (Broadcast)	
	192.168.200.144 (NA)	
	(09) 192.168.200.145 – 192.168.200.158	(200.145 – 200.146)
	192.168.200.159 (Broadcast)	
	192.168.200.160 (NA)	

- (10) 192.168.200.161 – 192.168.200.174 (200.161 – 200.162)
192.168.200.175 (Broadcast)
192.168.200.176 (NA)
- (11) 192.168.200.177 – 192.168.200.190 (200.177 – 200.178)
192.168.200.191 (Broadcast)
192.168.200.192 (NA)
- (12) 192.168.200.193 – 192.168.200.206 (200.193 – 200.194)
192.168.200.207 (Broadcast)
192.168.200.208 (NA)
- (13) 192.168.200.209 – 192.168.200.222 (200.209 – 200.210)
192.168.200.223 (Broadcast)
192.168.200.224 (NA)
- (14) 192.168.200.225 – 192.168.200.238 (200.225 – 200.226)
192.168.200.239 (Broadcast)
192.168.200.240 (NA)

When using the 240-subnet mask, it effectively breaks the network segment into smaller networks, with their own broadcast channels. Normally, broadcast is handled on .255.

{ Server & Client configs }

I will list the first 3 config files of my setup. Each tunnel needs to have IP pairs. I use the 1st two IP addresses of any network (Others can be used. I used the 1st number for the server and the 2nd number for the client.)

[Server configs (tap0.conf)]

```
# Device type  
dev tap
```

```
# Server adapter Vitural IP  
ifconfig 192.168.200.17 255.255.255.240
```

```
# SSL Key  
secret keys/static.key
```

```
# Tunnel UDP Port(1 port per tunnel)  
port 5001
```

```
# Restart Control
```

persist-key
persist-tun
ping-timer-rem
ping-restart 60
ping 10

Compression
comp-lzo

UID
user nobody
#group nobody

Additional Windows settings
tun-mtu-extra 32
tun-mtu 1500

Log detail level (Up to 10)
verb 5

Keeps repeated entries to a minimum
mute 10

[Client configs (tap0.opvn)]

Linux server.
remote LinuxIpAddress

port number than the default of 5000.
port 5001

Enable 'dev tap' or 'dev tun' but not both!
dev tap

Only define this option for 'dev tap'.
ifconfig 192.168.200.18 255.255.255.240

SSL Key
secret static.key

ping-restart 60
ping-timer-rem
up-delay 5
up office.up.bat
persist-tun

persist-key
resolv-retry 86400

keep-alive ping
ping 10
tun-mtu-extra 32
tun-mtu 1500

enable LZO compression
comp-lzo

moderate verbosity
verb 5
mute 10

[Server configs (tap1.conf)]

Device type
dev tap

Server adapter Virtual IP
ifconfig 192.168.200.33 255.255.255.240

SSL Key
secret keys/static.key

Tunnel UDP Port(1 port per tunnel)
port 5002

Restart Control
persist-key
persist-tun
ping-timer-rem
ping-restart 60
ping 10

Compression
comp-lzo

UID
user nobody
#group nobody

Additional Windows settings
tun-mtu-extra 32
tun-mtu 1500

```
# Log detail level (Up to 10)
verb 5
```

```
# Keeps repeated entries to a minimum
mute 10
```

[Client configs (tap1.opvn)]

```
# Linux server.
remote LinuxIpAddress
```

```
# port number than the default of 5000.
port 5002
```

```
# Enable 'dev tap' or 'dev tun' but not both!
dev tap
```

```
# Only define this option for 'dev tap'.
ifconfig 192.168.200.34 255.255.255.240
```

```
# SSL Key
secret static.key
```

```
ping-restart 60
ping-timer-rem
up-delay 5
up office.up.bat
persist-tun
persist-key
resolv-retry 86400
```

```
# keep-alive ping
ping 10
tun-mtu-extra 32
tun-mtu 1500
```

```
# enable LZO compression
comp-lzo
```

```
# moderate verbosity
verb 5
mute 10
```

[Server configs (tap2.conf)]

Device type
dev tap

Server adapter Virtual IP
ifconfig 192.168.200.49 255.255.255.240

SSL Key
secret keys/static.key

Tunnel UDP Port(1 port per tunnel)
port 5003

Restart Control
persist-key
persist-tun
ping-timer-rem
ping-restart 60
ping 10

Compression
comp-lzo

UID
user nobody
#group nobody

Additional Windows settings
tun-mtu-extra 32
tun-mtu 1500

Log detail level (Up to 10)
verb 5

Keeps repeated entries to a minimum
mute 10

[Client configs (tap2.opvn)]

Linux server.
remote LinuxIpAddress

port number than the default of 5000.

port 5003

Enable 'dev tap' or 'dev tun' but not both!
dev tap

Only define this option for 'dev tap'.
ifconfig 192.168.200.50 255.255.255.240

SSL Key
secret static.key

ping-restart 60
ping-timer-rem
up-delay 5
up office.up.bat
persist-tun
persist-key
resolv-retry 86400

keep-alive ping
ping 10
tun-mtu-extra 32
tun-mtu 1500

enable LZO compression
comp-lzo

moderate verbosity
verb 5
mute 10

{Linux Routes}

My Linux server's private address is 192.168.104.14, the following is from my office.up script file:

```
#!/bin/sh
```

```
/sbin/route add -net 192.168.100.0 netmask 255.255.255.0 gw 192.168.104.14  
/sbin/route add -net 192.168.101.0 netmask 255.255.255.0 gw 192.168.104.14  
/sbin/route add -net 192.168.102.0 netmask 255.255.255.0 gw 192.168.104.14  
/sbin/route add -net 192.168.103.0 netmask 255.255.255.0 gw 192.168.104.14  
/sbin/route add -net 192.168.105.0 netmask 255.255.255.0 gw 192.168.104.14  
/sbin/route add -net 192.168.106.0 netmask 255.255.255.0 gw 192.168.104.14  
/sbin/route add -net 192.168.112.0 netmask 255.255.255.0 gw 192.168.104.14  
/sbin/route add -net 192.168.115.0 netmask 255.255.255.0 gw 192.168.104.14
```

```
/sbin/route add -net 192.168.117.0 netmask 255.255.255.0 gw 192.168.104.14
```

{ Windows Routes }

This is from my office.up.bat on the first tunnel:

```
route add 192.168.100.0 mask 255.255.255.0 192.168.200.17
route add 192.168.101.0 mask 255.255.255.0 192.168.200.17
route add 192.168.102.0 mask 255.255.255.0 192.168.200.17
route add 192.168.103.0 mask 255.255.255.0 192.168.200.17
route add 192.168.104.0 mask 255.255.255.0 192.168.200.17
route add 192.168.105.0 mask 255.255.255.0 192.168.200.17
route add 192.168.106.0 mask 255.255.255.0 192.168.200.17
route add 192.168.112.0 mask 255.255.255.0 192.168.200.17
route add 192.168.115.0 mask 255.255.255.0 192.168.200.17
route add 192.168.117.0 mask 255.255.255.0 192.168.200.17
```

Please note, the default route for each client will be different. On my tap0, it's 17, but on my tap1, it would be 33 and my tap2 would be 49. Check the network table pairs listed against the configuration files I've shown.

{ Starting the server }

I created the following scripts to start/stop/restart the tunnels. I just modified the scripts available.

[start.sh]

```
#!/bin/bash
```

```
/bin/echo `date` ' Starting OpenVPN as a daemon' >>/etc/openvpn/vpn.log
```

```
# A sample OpenVPN startup script
# for Linux.
```

```
# openvpn config file directory
dir=/etc/openvpn
```

```
# load TUN/TAP kernel module
modprobe tun
```

```
# enable IP forwarding
/bin/echo 1 > /proc/sys/net/ipv4/ip_forward
```

```
/usr/local/sbin/openvpn --cd $dir --daemon --config /etc/openvpn/tap1.conf
```

```
/usr/local/sbin/openvpn --cd $dir --daemon --config /etc/openvpn/tap2.conf
/usr/local/sbin/openvpn --cd $dir --daemon --config /etc/openvpn/tap3.conf
/usr/local/sbin/openvpn --cd $dir --daemon --config /etc/openvpn/tap4.conf
/usr/local/sbin/openvpn --cd $dir --daemon --config /etc/openvpn/tap5.conf
/usr/local/sbin/openvpn --cd $dir --daemon --config /etc/openvpn/tap6.conf
/usr/local/sbin/openvpn --cd $dir --daemon --config /etc/openvpn/tap7.conf
/usr/local/sbin/openvpn --cd $dir --daemon --config /etc/openvpn/tap8.conf
/usr/local/sbin/openvpn --cd $dir --daemon --config /etc/openvpn/tap9.conf
/usr/local/sbin/openvpn --cd $dir --daemon --config /etc/openvpn/tap10.conf
```

[stop.sh]

```
#!/bin/bash
```

```
# stop all openvpn processes
```

```
/bin/echo `date` ' Terminating OpenVPN daemon' >>/etc/openvpn/vpn.log
```

```
killall -TERM openvpn
```

```
/bin/echo `date` ' Removing hard coded routes' >>/etc/openvpn/vpn.log
```

```
/sbin/route del -net 192.168.100.0 netmask 255.255.255.0 dev eth1
```

```
/sbin/route del -net 192.168.101.0 netmask 255.255.255.0 dev eth1
```

```
/sbin/route del -net 192.168.102.0 netmask 255.255.255.0 dev eth1
```

```
/sbin/route del -net 192.168.103.0 netmask 255.255.255.0 dev eth1
```

```
/sbin/route del -net 192.168.105.0 netmask 255.255.255.0 dev eth1
```

```
/sbin/route del -net 192.168.106.0 netmask 255.255.255.0 dev eth1
```

```
/sbin/route del -net 192.168.112.0 netmask 255.255.255.0 dev eth1
```

```
/sbin/route del -net 192.168.115.0 netmask 255.255.255.0 dev eth1
```

```
/sbin/route del -net 192.168.117.0 netmask 255.255.255.0 dev eth1
```

[restart.sh]

```
#!/bin/bash
```

```
cd /etc/openvpn
```

```
# stop all openvpn processes and remove hard coded routing (If this isn't done, OpenVPN will not restart)
```

```
# This is only necessary on the EPI side of the VPN. Current routes are hard coded.
```

```
# Normally, the routing would be handled by OpenVPN.
```

```
/bin/echo `date` ' Stopping OpenVPN' >>/etc/openvpn/vpn.log
```

```
/bin/echo `date` ' Issuing terminate signal' >>/etc/openvpn/vpn.log
```

```
killall -TERM openvpn
```

```
/bin/echo `date` ' Deleting all hard coded routes' >>/etc/openvpn/vpn.log
```

```
/sbin/route del -net 192.168.100.0 netmask 255.255.255.0 dev eth1  
/sbin/route del -net 192.168.101.0 netmask 255.255.255.0 dev eth1  
/sbin/route del -net 192.168.102.0 netmask 255.255.255.0 dev eth1  
/sbin/route del -net 192.168.103.0 netmask 255.255.255.0 dev eth1  
/sbin/route del -net 192.168.105.0 netmask 255.255.255.0 dev eth1  
/sbin/route del -net 192.168.106.0 netmask 255.255.255.0 dev eth1  
/sbin/route del -net 192.168.112.0 netmask 255.255.255.0 dev eth1  
/sbin/route del -net 192.168.115.0 netmask 255.255.255.0 dev eth1  
/sbin/route del -net 192.168.117.0 netmask 255.255.255.0 dev eth1
```

```
# initiate the start.sh
```

```
/bin/echo `date` ' Restarting OpenVPN' >>/etc/openvpn/vpn.log
```

```
/etc/openvpn/start.sh
```