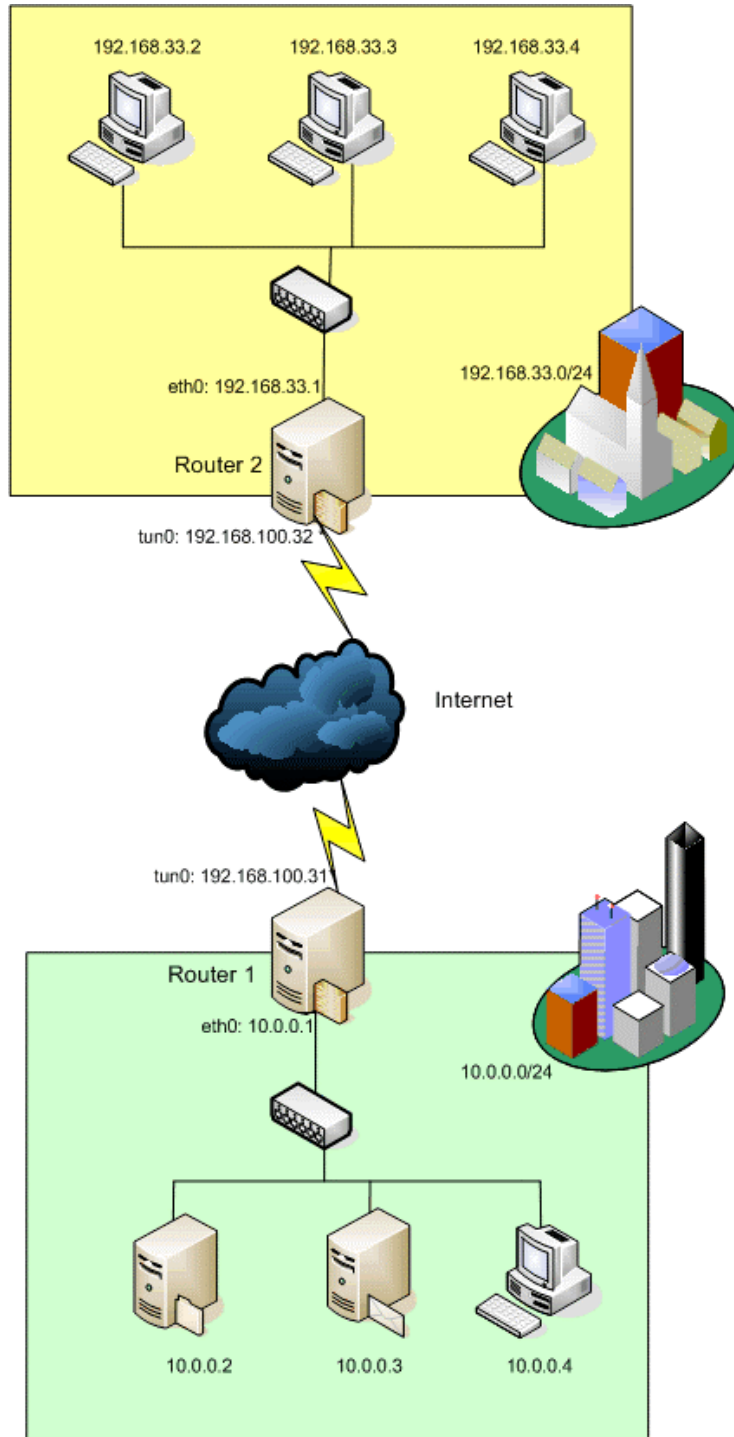


Set Up A SSH-Based Point To Point Connection (Linux-Tip.net)

OpenSSH version 4.3 introduced a new feature: the ability to create on-the-fly "Virtual Private Networks" via the tunnel driver (the so-called "tun" driver). This allows you to create a network interface that bridges two physically disparate network segments in different locations. This article explains how to use SSH to set up SSH-based point to point connections with OpenSuse 11.0, which can then be used to create routes that create virtual private networks.



What does this picture show? We have two OpenSuse 11.0 Routers, Router 1 and Router2 in different locations. Router 1 is connected to the "green network" 10.0.0.0/24 and Router 2 is

Set Up A SSH-Based Point To Point Connection (Linux-Tip.net)

connected to the “yellow network” 192.168.33.0/24 via Ethernet. Both routers providing NAT functionality and make Internet connection available for the connected clients.

In our test environment both routers are using the tun interfaces (192.168.100.31/32) to establish a SSH-based point to point connection. Important! Please change this IP to routable addresses in reality!

Once the tunnel is set up, the machines in the “yellow network” will be able to directly access the “green network” and vice versa using a secure VPN link.

Please keep in mind that you have to secure both routers using IP filter or firewall functionality in your real environment.

Step 1: Preparing the OpenSuse Routers

Install a basic system for both routers. Please keep in mind that you do not need desktop environments like KDE or Gnome. In the “Desktop Selection” part, use Other – Minimal Server Selection (Text Mode) (see pictures below).

Configure the network interface (eth0) in Router 2 using the IP addresses 192.168.33.1 and Router 1 -> 10.0.0.1. The net mask will be 255.255.255.0 on both sites. For further information and help, please use this link.

That’s it; we’re later able to configure the router by using the yast command. For test purposes, please disable the built in firewall.



Set Up A SSH-Based Point To Point Connection (Linux-Tip.net)

Step 2: Preparing Router 1

Open `/etc/ssh/sshd_config` using your favorite editor and enable the following lines:

```
PermitRootLogin yes ## Line 41
PermitTunnel yes ## line 111
RSAAuthentication yes ## line 45
PubkeyAuthentication yes ## line 46
```

Restart ssh by using the following command:
`/etc/init.d/sshd restart`

Generating the key :
`ssh-keygen -t rsa`

store the key in:
`/root/.ssh/id_rsa-vpn ## no passphrase`

Copy the public Key to Router2:
`scp /root/.ssh/id_rsa-vpn.pub 192.168.100.32:/root/.ssh/`

Remark: Please create the directory `/root/.ssh/` on router 2 first.

Step 3: Preparing Router 2

Start to configure sshd like you have done with Router1.

Add the key you have received from router1 to the file `authorized_keys`
`cat /root/.ssh/id_rsa-vpn.pub > /root/.ssh/authorized_keys`

Check the connection from router 1 to router 2. It should work without a password authentication.
`ssh -i /root/.ssh/id_rsa-vpn 192.168.100.32`

Step 4: Establishing the point to point connection and creating the routes

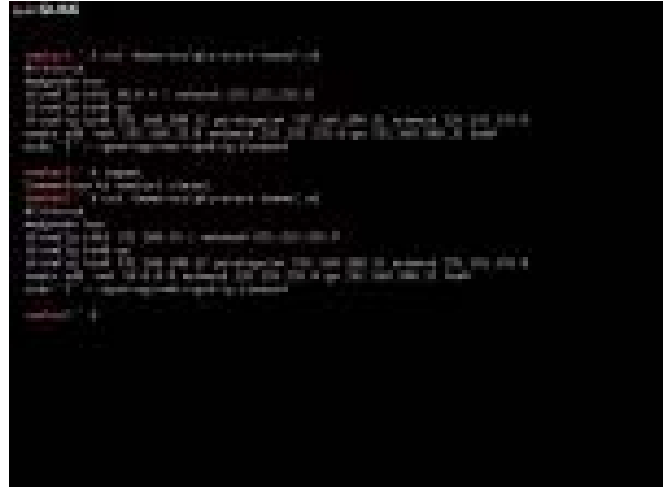
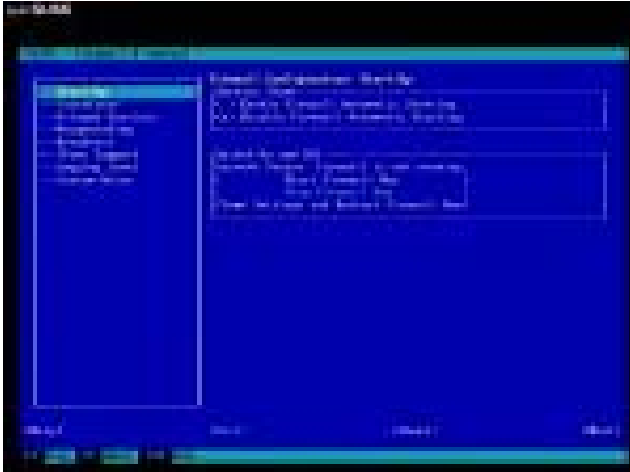
Creating a TUN device on both systems:
`ssh -w 0:0 -i /root/.ssh/id_rsa-vpn router2`

Configure the network devices and add default routes on both systems like this:

```
Router 1:
modprobe tun
ifconfig tun0 up
ifconfig tun0 192.168.100.31 pointopoint 192.168.100.32 netmask 255.255.255.0
route add -net 192.168.33.0 netmask 255.255.255.0 gw 192.168.100.32 tun0
echo "1" > /proc/sys/net/ipv4/ip_forward
```

```
Router 2:
modprobe tun
ifconfig tun0 up
ifconfig tun0 192.168.100.32 pointopoint 192.168.100.31 netmask 255.255.255.0
route add -net 10.0.0.0 netmask 255.255.255.0 gw 192.168.100.31 tun0
echo "1" > /proc/sys/net/ipv4/ip_forward
```

Set Up A SSH-Based Point To Point Connection (Linux-Tip.net)



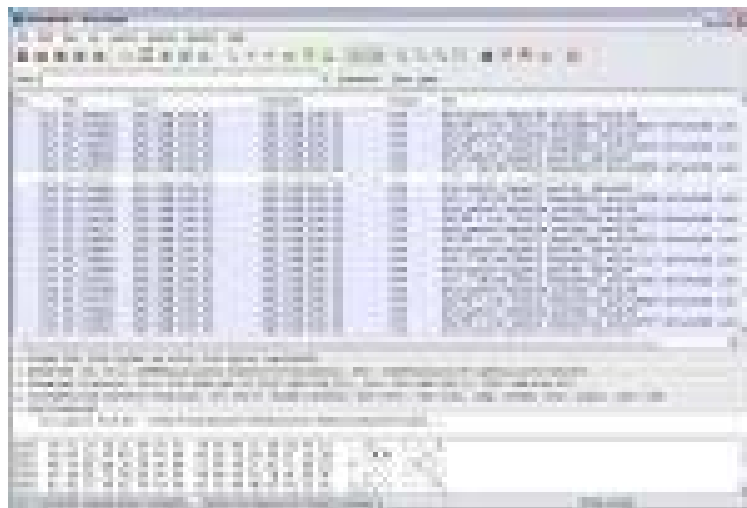
Step 5: Test the connection

SSH is such a wonderfully flexible and versatile program, and it has built-in support for creating a secure VPN to do just that. The idea is to make it so that all traffic from the “green network” to the “yellow network” is routed through the remote server using a secure VPN link. We will use Wireshark to test this.

Test 1: Ping the desktop with IP 192.168.33.2 (Yellow network) from the Mail server with IP 10.0.0.3 (green network).

Test2: Try to access a webpage running on the server with IP 10.0.0.2 (green network) from desktop with IP 192.168.33.2 (yellow network).

In both test we should just see encrypted packages like displayed in the picture below.



Step 5: Create your own script to start the tunnel automatically

There are other ways to set up SSH-based point to point connections and to automate the process, but we will leave these as an exercise for the reader. If you haven't a clue how to start, download for example these files and try to adjust it for your favorite distribution.