

KNOPPIX HACKS™

Includes
Knoppix on
CD-ROM

*100 Industrial-Strength
Tips & Tools*



O'REILLY®

Kyle Rankin

HACK
#40

Create an Emergency Router

Turn Knoppix into a router or firewall.

Avoid thinking that Knoppix can be used only for demonstration purposes or is fit only for light desktop use. Knoppix is a full-fledged portable installation of Linux, which means it can do most anything an installed version of Linux can do. For instance, Knoppix comes ready to use as a fully functional router or firewall with all of the normal utilities, such as *route* and *iptables*, that you use on any other Linux distribution. These tools make Knoppix particularly handy if you need an emergency Network Address Translation (NAT) router or a bridge. When the router goes down, you can take your Knoppix “demonstration” CD, boot it on a spare machine with two NICs, and demonstrate how to save the day. With just a few commands, you can route across any of the network connections Knoppix supports from DSL to dial-up to wireless. This hack walks you through turning a machine into a bridge and then a NAT router.

Configure the Network

The machine you are using as the emergency router must have two different network connections that already work independently of each other. This can be satisfied with two network cards, a network card and a modem, a network card and a wireless card, or any two network connections that Knoppix supports. Configuring network connections under Knoppix is covered in “Connect to the Internet” [Hack #17].

After both networks are working, you can link the two either with a bridge or with NAT. Generally, you want to use a bridge to connect two local networks so that machines on either network can communicate directly with any machine on the other network. Use NAT when you need to share a single Internet or network connection across a local network with the NAT machine acting as a sort of firewall. Machines on the other side of the NAT are not able to communicate directly with local machines unless you set up firewall rules on the NAT machine to forward ports.

To create either of these routers, you must enable IP forwarding in the Linux kernel. Most firewall and routing HOWTOs instruct you to do this by running the following command as root:

```
root@tty0[root]# echo 1 > /proc/sys/net/ipv4/ip_forward
```

However, under Knoppix, you must change that command so that it works under the *sudo* environment by typing:

```
knoppix@tty0[knoppix]$ sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
```

Now that IP forwarding is enabled, you can configure your bridge or NAT router.



If you are dropping this Knoppix machine in the place of a broken router, save a lot of trouble by giving Knoppix the same IPs as the previous router. In the case of a bridge, once you provide Knoppix with the same IPs and enable IP forwarding, the bridge is ready to go.

For the purposes of these examples, assume that the Knoppix computer is connected to two networks—192.168.0.* on eth0 and 192.168.1.* on eth1. Run *ifconfig*, and you should get the following output:

```
knoppix@tty1[knoppix]$ /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:DE:AD:BE:EF:00
          inet addr:192.168.0.5  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6918 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4678 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:675976 (660.1 KiB)  TX bytes:447963 (437.4 KiB)
          Interrupt:9 Base address:0xb800

eth1      Link encap:Ethernet  HWaddr 00:C0:FF:EE:00:00
          inet addr:192.168.1.5  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4933 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4988 errors:1 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:496574 (484.9 KiB)  TX bytes:749568 (732.0 KiB)
          Interrupt:3 Base address:0x100

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:33 errors:0 dropped:0 overruns:0 frame:0
          TX packets:33 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3016 (2.9 KiB)  TX bytes:3016 (2.9 KiB)
```

These networks already have a default route set up for each of these interfaces, which you can see by running the route command:

```
knoppix@tty1[knoppix]$ route
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use
Iface
192.168.0.0      *               255.255.255.0   U        0      0      0 eth0
192.168.1.0      *               255.255.255.0   U        0      0      0 eth1
default          192.168.1.1     0.0.0.0         UG       0      0      0 eth0
default          192.168.0.1     0.0.0.0         UG       0      0      0 eth1
```

Build a Bridge

Creating a bridge with *route* is pretty straightforward once you see the commands involved. In fact, if both networks are already configured to use this machine as the gateway, and you have already enabled IP forwarding, then congratulations—you are finished! Otherwise, read the following instructions to learn how to configure the routing for your bridge.

So far I haven't had to change anything in the networking. In my example, I set up static IPs (“[Connect to the Internet](#)” [\[Hack #17\]](#)), but if you had DHCP running on either or both sides of the network with different default gateways, the bridge would have worked fine too. At this point, the Knoppix machine should be able to ping machines on both the 192.168.0.* and the 192.168.1.* networks, but machines on 192.168.0.* shouldn't be able to ping 192.168.1.* and vice versa.

I want to make the Knoppix machine the link between my two networks. For this to happen, the machines on either network must use the Knoppix machine as the bridge to the other network. If one of the two networks is already configured to use this Knoppix machine as its default gateway, then all packets going outside of the subnet route through it by default, and you don't have to bother with any extra routing for that network. If both networks are already set to use this machine as the default gateway, then you are finished. Either of these scenarios might be the case if you drop in Knoppix to replace a bridge and assign it the same IP addresses as the previous bridge.

If a network does not use the Knoppix machine as its gateway, you must add a route to the actual gateway on that subnet. This route tells the gateway to route any traffic going to the other subnet, through the Knoppix bridge. To route through the Knoppix bridge requires root access to the network's default gateway, to add the new route. In our example, the default gateways are 192.168.0.1 and 192.168.1.1, respectively, so on 192.168.0.1, run the following command as root:

```
root@tty0[root]# route add -net 192.168.1.0 netmask 255.255.255.0  
gw 192.168.0.5
```

On 192.168.1.1, run:

```
root@tty0[root]# route add -net 192.168.0.0 netmask 255.255.255.0  
gw 192.168.1.5
```

Once you set up these new routes, machines on either side of the bridge can ping each other, and your bridge is complete.

Network with NAT

Performing IP masquerading or NAT with Knoppix is as simple as configuring it as a bridge, if not simpler. NAT is commonly used to share a single public IP address (like you might get from a DSL or cable provider) with a local network behind the NAT router.

For NAT to work, all of the machines on the local network must be configured to use the Knoppix machine as the default gateway. In our example, the 192.168.1.* network is behind this NAT “firewall” to access the 192.168.0.* network, so each of the machines on 192.168.1.* is using 192.168.1.5 (the IP address we assigned the NIC connected to the local network) as their default gateway.

The NAT works by taking all of the packets coming from 192.168.1.* (the local network) and going to 192.168.0.* (the external network) and making them appear as though they are from 192.168.0.5—the IP address we assigned the NIC connected to the external network. When a machine on the external network responds, it responds directly to 192.168.0.5. Then the Knoppix machine translates the address to refer to the 192.168.1.* machine that originally sent the packet. Then Knoppix forwards it to the local network. For all intents and purposes, the 192.168.0.* network doesn’t know that the 192.168.1.* network exists.

To set up Knoppix as a NAT router, you really only need to type in a single *iptables* command. To create a NAT for our example network, type:

```
knoppix@tty0[knoppix]$ sudo iptables -t nat -A POSTROUTING -s  
192.168.1.0/255.255.255.0 -o eth0 -j SNAT --to-source 192.168.0.5
```

This *iptables* command creates a rule to take all packets coming from the 192.168.1.* network and going from eth0 and makes them appear as though they are from 192.168.0.5. If you want to use IP masquerading instead of NAT (useful for forwarding over a dial-up connection that might drop while the computer is booted, which results in a different IP), type the following command instead:

```
knoppix@tty0[knoppix]$ sudo iptables -t nat -A POSTROUTING -o eth0 -j  
MASQUERADE
```

Substitute ppp0 for eth0 if you are forwarding over a dial-up connection. At this point, you should be able access the outside 192.168.0.* network from any of the machines on the 192.168.1.* network.

The *iptables* command creates a NAT rule, but doesn’t actually create a proper firewall. NAT does prevent people from easily accessing any local IPs behind the NAT router. However, if you are interested in setting up Knoppix with firewall rules suitable for your network, you can reference one of

the many great HOWTOs and tutorials on using stateful packet filtering under Linux with *iptables*.

See Also

- The official netfilter page at <http://www.netfilter.org/documentation> (in particular, the packet-filtering HOWTO).
- The Advanced Routing HOWTO at http://www.ibiblio.org/pub/Linux/docs/HOWTO/other-formats/html_single/Adv-Routing-HOWTO.html.