Thursday, 16 January 2014

Raspberry Pi as a VPN Wireless Access Point

The following post explains how you can turn a Raspberry Pi (RPI) into a wireless router that connects to the Internet over a VPN. By plugging this into your existing wireless router, you give yourself a second wireless network, and any devices connected to this network will access the Internet over the VPN.

This is especially useful for:

- Sharing a single VPN connection between several devices.
- Using the VPN connection with devices that don't support VPN or proxy settings.

As an added bonus, the use of NAT and a couple of firewall rules provides a good level of security for any connected device.

I've found this setup works very nicely, and is fine even for streaming media.

Sources

These instructions draw heavily from the following extremely useful articles:

- RPI Wireless Hotspot
- IPTables HowTo
- 10 iptables rules to help secure your Linux box
- Setting up a DNS for the local network on the Ubuntu 12.04 (Precise Pangolin) server

You'll Need

- A Rasberry Pi with Raspbian installed.
- A wireless USB adapter with a chipset that supports Access Point or Master mode. I used a Panda PAU03, and it worked perfectly and has a good signal. See the RPI Wireless Hotspot article for other options.
- A wired Ethernet connection between the RPI and your router.
- A VPN service you can connect to that supports tunneled connections and OpenVPN. It's entirely possible you could get this to work with a tap VPN connection, but I can only vouch for the tunneled variety. OpenVPN support is a must: watch out as not all providers support it.

Instructions

The following instructions assume a basic knowledge of Linux, the command prompt, and the ability to edit files with an editor such as Vi or Nano.

I've reconstructed these from my command-line history and the above articles, but haven't done a clean run-through, but I think they should work. Please let me know if you find any mistakes.

Basic Security

Your going to be connecting your RPI to the rest of the Internet via a VPN, which means it won't enjoy the protection of your router's firewall: the VPN tunnel will punch right through and expose your RPI to any machine on the Internet. We'll lock down the VPN connection later on, but before you start, make sure you've changed the default user password using the passwd command.

Initial Setup

Before you start, your RPI will need to be connected to your router via the Ethernet port and able to access the Internet, and your wirless USB adapter will need to be plugged-in.

Install Software

Install the access point server (hostapd), DHCP server (udhcpd), OpenVPN and DNS proxy server (bind9):

```
sudo apt-get install hostapd udhcpd bind9 openvpn
```

Configure and Secure the VPN

Your VPN service provider should be provide an OpenVPN configuration file you can use to connect to their VPN server. Copy this file into `/etc/openvpn`, and rename it `openvpn.conf`:

```
cp <your config file> /etc/openvpn/openvpn.conf
```

Start the OpenVPN service:

```
sudo service start openvpn
```

You can check the connection is open with:

```
ifconfig
```

You should see a network interface listed called `tun0`, assuming your VPN provider uses a tunnel (rather than a tap) interface.

You can test the VPN tunnel using the following command:

```
curl --interface tun0 freegeoip.net/json/
```

This uses an IP geolocation service to look up the geographic details of the IP address your tunnel connection is using (you might need to give the connection a few seconds to come up). The IP address and other details should be different if you stop the VPN service:

```
sudo service openvpn stop
curl freegeoip.net/json/
```

You now need to lock down that VPN tunnel using iptables. The following changes will prevent any unsolicited connections from other machines on the Internet, and make your Pi much less visible on the network:

```
sudo iptables -A INPUT -i tun0 -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A INPUT -i tun0 -j DROP
```

Now save the iptables rules:

```
sudo sh -c "iptables-save > /etc/iptables.nat.vpn.secure"
```

To ensure the rules are re-applied at reboot, edit the file `/etc/network/interfaces` and add the following line at the end of the file:

```
up iptables-restore < /etc/iptables.nat.vpn.secure
```

Before we go any further, restart the VPN connection:

```
sudo service openvpn restart
```

Configure Wireless Network and Access Point

Now we have a VPN, we can set up our wireless network and access point.
First, check your wireless adapter is working:

```
ifconfig
```

You should see an interface listed called `wlan0`.

Edit the file `/etc/udhcpd.conf` as follows:

```
start 192.168.0.2
end 192.168.0.254
```

```
interface wlan0
remaining yes
opt dns 192.168.0.1
option subnet 255.255.255.0
opt router 192.168.0.1
option lease 864000 # 10 days
```

This will give your new wireless network the IP range 192.168.0.1 - 192.168.0.254, and assign the address 192.168.0.1 to the wireless connection of your RPI. You might need to change the IP addresses if they clash with your existing network (check using ifconfig and looking for the IP address of the `eth0` interface). The above configuration also tells any connected devices to use the RPI for their DNS server: we'll get to that later.

Edit the file `/etc/default/udhcpd` and un-comment the following line by removing the # from the front:

```
#DHCPD_ENABLED="yes"
```

becomes:

```
DHCPD_ENABLED="yes"
```

Set your Pi's IP address:

```
sudo ifconfig wlan0 192.168.0.1
```

and to keep the change at reboot, edit the file `/etc/network/interfaces` and replace the line:

```
iface wlan0 inet dhcp
```

with:

```
iface wlan0 inet static
  address 192.168.0.1
  netmask 255.255.255.0
```

Note those are tabs in front of the indented lines.

In the same file, comment out the following lines by adding a hash at the start:

```
allow-hotplug wlan0
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet manual
```

becomes:

```
#allow-hotplug wlan0
#wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
#iface default inet manual
```

Now configure your wireless connection by editing the file `/etc/hostapd/hostapd.conf` as follows (you'll need to create it if it doesn't exist already):

```
interface=wlan0
driver=nl80211
ssid=YOUR_SSID
hw_mode=g
channel=6
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
```

```
wpa=2
wpa_passphrase=YOUR_PASSWORD
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Change YOUR_SSID and YOUR_PASSWORD to be network name and password respectively for your new wireless network. It's also worth checking the channel your existing router is using and making sure this one doesn't clash.

Now edit the file `/etc/default/hostapd` and change the line:

```
#DAEMON_CONF=""
```

to:

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

Now start up the wireless network:

```
sudo service hostapd start
sudo service udhcpd start
```

and make sure the services start at reboot:

```
sudo update-rc.d hostapd enable
sudo update-rc.d udhcpd enable
```

Configure DNS

Now we'll set up a local caching DNS server on the RPI which will be used by the connected devices.

Edit the file `/etc/bind/named.conf.options` and add a forwarders section as follows:

```
forwarders {
8.8.8.8;
8.8.4.4;
};
```

The above IP addresses will use Googles public DNS server, but obviously you can choose an alternative if you prefer. Just don't try to use the DNS of your existing router, which won't be accessible over the VPN.

Now restart the DNS server:

```
sudo service bind9 restart
```

and make sure it starts again at reboot:

```
sudo update-rc.d bind9 enable
```

Set Up NAT for the VPN Connection

Finally, we just need to set up NAT for the VPN connection, which will allow us to share the connection with any connected devices.

Enable NAT:

```
sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
```

To set this at reboot, edit the file `/etc/sysctl.conf` and add the following line at the end:

```
net.ipv4.ip_forward=1
```

Now set up NAT for the VPN connection:

```
sudo iptables -t nat -A POSTROUTING -o tun0 -j MASQUERADE
```

and to save this change so it's re-applied at reboot:

```
sudo sh -c "iptables-save > /etc/iptables.nat.vpn.secure"
```

Interestingly the RPI Wireless Hotspot article I based a lot of this on suggested adding a couple of other iptables rules to link the wireless and wired network adapters, but I found they weren't necessary. If you find the above alone isn't working, try the following:

```
sudo iptables -A FORWARD -i tun0 -o wlan0 -m state --state RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -i wlan0 -o tun0 -j ACCEPT
```

Testing

OK, you're done. To test the setup, connect a device to the new wireless network using the password you configured. Now open a browser and navigate to the IP geolocation service we used to test the VPN connection earlier:

http://freegeoip.net/

With any luck you should see the same details you saw when you accessed the service over the VPN on the command line. To make absolutely sure things have gone well, try changing back to your other wireless network and refreshing the page: you should see the details change.

Now if you're feeling really confident you can reboot your RPI and check everything comes back up OK. :)

Labels: access point, bind9, hostapd, iptables, openvpn, raspberry pi, router, rpi, udhcpd, vpn, wifi, wireless