VPN Tutorial for Newbies:
Installing BolehVPN on the ASUS RT-N66U Router with Merlin Firmware

By Don DeGracia, Jan 01, 2014
dondeg@compuserve.com
www.dondeg.com

Introduction

Imagine walking around naked: everybody can see everything!! This is effectively what you are doing if you do not try to secure yourself on the internet. The hardware and software is now so sophisticated that any tom, dick or harry can easily snoop on your internet activity and see everything you do.

I am breaking with the format of this blog to post this tutorial because, well, without freedom of speech, it doesn't really matter what else I talk about. So, this post is about protecting your privacy on the internet, and thereby protecting your freedom of speech.

Since the ISPs, NSA, and everybody else are spying on internet traffic, I, like so many others, have begun using a VPN (virtual private network) service. VPNs encrypt all of your internet traffic so that anyone tapping your internet connection will not be able to see what you are doing. It is kind of like tinting your car windows so people cannot see in., or like having curtains over your windows, again, so people cannot see in.

However, all of this is very confusing and there is a steep learning curve. The purpose of this post is to provide some simple explanations of things, and to provide step by step instructions for people just getting started.

First order of business: there are two main ways to set up VPNs. (1) As a server, where you make and use the VPN, and (2) As a client, where someone else (a company) has the VPN, and you pay them to log in as a client and use their VPN service to cloak yourself. This post is about the 2nd option, where you purchase VPN service from someone else. I don't do **not** discuss here setting up a local VPN server on your home network. There are many tutorials out there if this is what you want to do.

My main starting point for choosing a VPN company was this article. The general consensus is to NOT use free VPN services because, well, you get what you pay for. VPN companies you pay for either allow a short free trial or a very inexpensive trial period of a few days or a week, so you can try it out and see if you are happy with it.

Most VPN companies provide a small application that runs on one's computer and routes all the internet traffic through the VPN's servers. Installing and running such apps are easy and all the companies have straight-forward tutorials for getting going. If you are only going to run the VPN on one or two computers, just install their application and you are done, and you can stop reading.

Once I started to learn about this stuff, it was apparent one can set up their VPN service to run on their home wireless router. Then, everything connected to the router is protected by the VPN. However, this is harder to do, and I had to search through myriad message boards to get things to work. So, the point of this post is to give a step-by-step tutorial of how I set up my system. Be aware, it is highly specific to both my router and the VPN service I am using. However, even if you don't have the same set up I have, at least it can show you all the steps from A to Z and give you a guide-post to figure out how to do the same things on your system.

Ingredients

Like any recipe, you need a list of ingredients:

[1] **Internet connection**. Obviously! I have cable internet.
[2] **Router** – Asus RT-N66U (The "dark knight" – sheesh – marketing!). What I am describing here only works with the newest routers, which can be programmed. I actually went and just bought this router (again as of Dec. 2013) just for the purpose of being able to program it to run the VPN service.
[3] **Router firmware**. This is the web interface to control your router. For the RT-N66U, I installed the Merlin version of the Asus firmware (version 3.0.0.4.374.35_4).
[4] **VPN Service**. I decided to go with BolehVPN, because I heard they had good customer service (which subsequently I discovered is true – they are very responsive to customer support) and for other reasons in the article above. You need to research companies and go with one that meets your needs and that you think is trustworthy. The best type of VPN service uses the open source OpenVPN which is a form of the SSL/TLS protocol. The company you choose should allow an option for OpenVPN. Since it is open source, and everybody can see the code, there is no change for hidden backdoors put in by the NSA or whoever. For some background, here is a short list of VPN protocols. Here is a more extensive explanation of VPNs in general.

Steps Overview

Like any recipe, you need to follow the steps.

[1] Install router.
[2] Install firmware.
[3] Program the OpenVPN client in the firmware to connect to your VPN.
[4] (optional) Program the router to control whether specific devices go through the VPN or through your ISP.

STEP 1: INSTALL ROUTER.

This is straight-forward: just follow the instructions that came with your router.

STEP 2: UPGRADE FIRMWARE.

A. DD-WRT. I went through a whole thing with the open source firmware DD-WRT. The short of it is, this didn't work for me, but I will briefly give some background. DD-WRT is a firmware that allows one to have much greater control over their router. It is an incredible piece of software, but for many people, including me, it is probably WAY too much control. The reason dd-wrt enters the picture is because it contains an OpenVPN client, and many VPN companies have instructions for setting up their VPN service on dd-wrt. So there is A LOT of stuff on message boards about setting up VPN services on dd-wrt. It may be the case that, for your particular router and your particular VPN company, you need to install dd-wrt on your router. Then, the VPN company will give specific instructions to install access to their VPN servers on dd-wrt. If this is the case for you, just follow instructions.

B. Merlin/Tomato firmware. In my case, it turned out that all I needed to do was install the Merlin firmware on my router. Merlin has an OpenVPN client based on the Tomato firmware (Tomato, like dd-wrt, is a non-proprietary firmware that expands your router's capabilities). It is easy to install Merlin and the instructions provided are easy to follow.

Note: there is a whole thing out there about the 32 kB or 64 kB nvram which is particularly confusing in the dd-wrt forums. If you install the latest version of Merlin on theh RT-N66U, it will make your router 64 kB nvram, and you don't need to do anything fancy. Just follow the install instructions in the Merlin download. The steps to install Merlin on your router are:

A. Download the version of Merlin for your ASUS router.
B. Unzip the file.
C. Log on to your router by opening a web page and typing 192.168.1.1 in your web browser (input your user name and password. You DO have a strong password, right? You can check here)
D. Click the "Administration" tab on the left panel.
E. Click the "Firmware Upgrade" tab.
F. Browse for the Merlin *.trx file.
G. Click the "Upload" button.
H. Your router will upgrade the firmware…just wait until it is done.
I. As it says in the Merlin instructions, you generally **don't** need to reset your router to defaults, but make sure to read the install instructions!

That's it! You now have an OpenVPN programmable ASUS RT-N66U.

STEP 3: PROGRAM THE OPENVPN CLIENT.

Okay, so this is where you really need step by step instructions. Whether you have Merlin, dd-wrt or Tomato, the first thing to do is determine if your VPN company has step by step instructions for programming the VPN client in your firmware. I saw several examples of companies that provide this information. BolehVPN just released, as of Dec. 2013, instructions for programming Merlin.

To install OpenVPN instructions on your router you will need the OpenVPN certificates and key files and *.opvn files that contain connection instructions. These should be supplied by your VPN company and there should be someplace you can download these from your VPN company. You will need these to program the router. REMEMBER, this tutorial assumes you have chosen a VPN company that allows OpenVPN connections.

For BolehVPN, the instructions to program the router are:

A. From the BolehVPN web site, download the zip file containing the keys/certificates and *.ovpn connection files. Unzip to a folder on your desktop.
B. Log into the Merlin interface at 192.168.1.1.
C. Go to Advanced Settings on the left, and click on "VPN".
D. Click on the "OpenVPN Client" tab.
E. You can set up TWO different VPN log on set ups by choosing either "Client 1" or "Client 2" in "Select Client Instance". Set either Client 1 or Client 2.
F. In the "Import ovpn file", hit "Browse" and select the *.ovpn file that has the connection you wish to use.
G. Click the "Upload" button.
H. Many of the subsequent boxes will automatically fill once you upload the *.opvn file. But to be safe, check that the settings are the following:
    a. "Start with WAN" = no
    b. "Interface Type" = TUN
    c. "Protocol" = UDP
    d. "Server Address and Port" = should display IP address to which you will connect, and the port should be 443 (which is specific for the VPN company I use, yours might be different).
    e. "Firewall" = automatic
    f. "Authorization Mode" = TSL
    g. "Username/Password Authentication" = No
    h. "Extra HMAC Authorization" = Outgoing (1)
    i. "Create NAT on tunnel" = Yes
I. In the "Authorization Mode" (which is set to TSL) click on "Content modification of Keys & Certificates". This will open a new floating window with three boxes. You upload keys and certificates into these boxes as follows:
    a. Certificate Authority is the file named **ca.crt**
    b. Client Certificate is the file named '**your---username.crt**' (replace with your actual username)
    c. Client Key is the file named **'your---username.key'** (replace with your actual username)

IMPORTANT: For each of the files named above, you will need to open the file with a text editor AND COPY/PASTE ONLY the part that starts with

-----Begin xxx ----
to
-----End xxx ----

J. After copying and pasting all 3, click "Save" and return to previous screen.
K. In "Authentication Mode" drop box choose "Static Key"
L. Again, click on "Content modification of Keys & Certificates".
M. This now opens another floating window with just one box. COPY/PASTE ta.key as per the same format above.
N. Click "Save" and return to previous screen.
O. Go back to the "Authentication Mode" drop box and reset it to "TSL".
P. Under "Advanced Settings" make sure they are set as:
    a. "Poll Interval" = 0
    b. "Redirect Internet traffic" = No
    c. "Accept DNS Configuration" = Exclusive
    d. "Encryption cipher" = AES-128-CBC
    e. "Compression" = None
    f. "TLS renegotiation Time" = -1
    g. "Connection Retry" = -1
    h. "Verify Server Certificate" = No
Q. In the "Custom Configuration" box, do NOT make any changes.
R. Click "Apply" at the bottom of the screen to save changes.
S. At the top of the screen click "ON" to start the VPN.

A couple notes about the above:

[1] You can program two separate *.opvn set ups in Client 1 and Client 2 (step E above). You will need to repeat all the steps above if you program into Client 2.

[2] Under Step Hd above, you can manually change the IP address to connect to a different IP. If all the settings in two *.ovpn files are the same, then all you need to do is change the IP address in the "Server Address and Port" box, and hit "Apply". If you do this, you should turn the VPN client OFF first.

BINGO! You are now connected to BolehVPN through your router! All the connections on your local area network (LAN), including wireless connections, will be going through the VPN at this point.

STEP 4: PROGRAM WHICH DEVICES GO THROUGH THE VPN

However, having all the connections go through the VPN creates problems for some services. When you go through the VPN, it slows your internet connection for a variety of reasons. For most purposes, you won't even notice. For example, I can still watch high def Youtube video and see no effect of the VPN running. However, if you are, say, playing

online games where every ounce of bandwidth is used, you will notice a significant speed decrease.

So, it would be nice to be able to tell the router which device to send through the VPN and which to allow to just normally through your ISP connection. This last step explains how to do this. This was the hardest step of all because it relies on programming your router using scripts and it requires logging into your router using a program other than the web interface. But, I am not a network programmer at all, and I managed to piece together how to do this. Because I had to piece together so many sources, I think it will be helpful to somebody to just list the steps here.

NOTE: THIS IS TOTALLY SPECIFIC FOR THE ASUS ROUTERS RUNNING MERLIN! For other routers and other firmware, I wish you the best of luck!!

Steps overview:

A. Set up JFFS partition on the RT-N66U.
B. Obtain and customize script for controlling which devices will go through the VPN.
C. You will need to know the IP addresses of devices you want to EXCLUDE from the VPN connection.
D. Copy script to JFFS partition and run it.

STEP A. SET UP JFFS PARTITION.

(This is taken from a post by wizin that is here)

[1] Assuming you have VPN Account and have it already working with OpenVPN in your Asus-Merlin Router (test manually if VPN works first)
[2] So make sure its **ON** and **Start with WAN** option
[3] Goto to **Administration** > **System**
[4] **Enable JFFS partition** = YES
[5] **Format JFFS partition at next boot** = YES
[6] **REBOOT ROUTER**

You now have space on your router for storing the script that will control which devices connect to either the VPN or ISP.

STEP B. THE SCRIPT.

The following script is taken from message #9 by Grdnkln that is posted here. What this script does is BY DEFAULT routes all devices through the VPN. The way the script works is you list in the very last lines of the script the IP addresses of the devices you do NOT want to pass through the VPN. What you need to do is:

[1] Open a text file in notepad.
[2] Copy and paste everything between the START SCRIPT and END SCRIPT markers into the text file and save the text file.

[3] What to do with the script is described below, after the script.

Note: the entire script is also attached at the end of this file

------START SCRIPT----------

```
#!/bin/sh

sleep 2

touch /tmp/000wanstarted


# This code goes in the WAN UP section of the Tomato GUI.
# This code based on the contributions from this thread:
#              http://www.linksysinfo.org/index.php?threads/route-only-specific-ports-through-vpn-
openvpn.37240/
#
# And from material in these articles:
#  http://linux-ip.net/html/adv-multi-internet.html
#  http://fedorasolved.org/Members/kanarip/iptables-howto
#
# This script configures "selective" VPN routing. Normally Tomato will route ALL traffic out
# the OpenVPN tunnel. These changes to iptables allow some outbound traffic to use the VPN, and
some
# traffic to bypass the VPN and use the regular Internet instead.
#


#  To list the current rules on the router, issue the command:
#      iptables -t mangle -L PREROUTING
#
#  Flush/reset all the rules to default by issuing the command:
#      iptables -t mangle -F PREROUTING
#

#
# First it is necessary to disable Reverse Path Filtering on all
# current and future network interfaces:
#



for i in /proc/sys/net/ipv4/conf/*/rp_filter ; do
  echo 0 > $i
done

#
# Delete and table 100 and flush any existing rules if they exist.
#
ip route flush table 100
ip route del default table 100
ip rule del fwmark 1 table 100
ip route flush cache
iptables -t mangle -F PREROUTING

#
# Copy all non-default and non-VPN related routes from the main table into table 100.
# Then configure table 100 to route all traffic out the WAN gateway and assign it mark "1"
#
# NOTE: Here I assume the OpenVPN tunnel is named "tun11".
#
#
ip route show table main | grep -Ev ^default | grep -Ev tun11 \
  | while read ROUTE ; do
      ip route add table 100 $ROUTE
done
ip route add default table 100 via $(nvram get wan_gateway)
ip rule add fwmark 1 table 100
ip route flush cache
```

```
#
# Define the routing policies for the traffic. The rules will be applied in the order that they
# are listed. In the end, packets with MARK set to "0" will pass through the VPN. If MARK is set
# to "1" it will bypass the VPN.
#
# EXAMPLES:
#
#  All LAN traffic will bypass the VPN (Useful to put this rule first, so all traffic bypasses
the VPN and you can configure exceptions afterwards)
#    iptables -t mangle -A PREROUTING -i br0 -j MARK --set-mark 1
#  Ports 80 and 443 will bypass the VPN
#    iptables -t mangle -A PREROUTING -i br0 -p tcp -m multiport --dport 80,443 -j MARK --set-
mark 1
#  All traffic from a particular computer on the LAN will use the VPN
#    iptables -t mangle -A PREROUTING -i br0 -m iprange --src-range 192.168.1.2 -j MARK --set-
mark 0
#  All traffic to a specific Internet IP address will use the VPN
#    iptables -t mangle -A PREROUTING -i br0 -m iprange --dst-range 216.146.38.70 -j MARK --set-
mark 0
#  All UDP and ICMP traffic will bypass the VPN
#    iptables -t mangle -A PREROUTING -i br0 -p udp -j MARK --set-mark 1
#    iptables -t mangle -A PREROUTING -i br0 -p icmp -j MARK --set-mark 1


# By default all traffic goes through the VPN
iptables -t mangle -A PREROUTING -i br0 -j MARK --set-mark 0

# Spotify explicitly by passes the VPN
#  All traffic from a particular computer on the LAN will use the VPN
    iptables -t mangle -A PREROUTING -i br0 -m iprange --src-range 192.168.1.xxx -j MARK --set-
mark 1
    iptables -t mangle -A PREROUTING -i br0 -m iprange --src-range 192.168.1.xxx -j MARK --set-
mark 1
```

------END SCRIPT----------

STEP C. CUSTOMIZING THE SCRIPT.

You next need to customize the script by telling it which devices will NOT pass through the VPN. Customizing the script is easy. You just need to copy and paste the final line of the script for each device you want to exclude from the VPN:

```
iptables -t mangle -A PREROUTING -i br0 -m iprange --src-range 192.168.1.xxx -j MARK --set-mark 1
```

All you need to do is replace “192.168.1.xxx” with the actual IP address of the device you want to exclude from the VPN.

To get the IP address of the device:

[1] Open Merlin in your web browser.
[2] On the left select “Network Map”
[3] Click on the ball that says “Clients”
[4] This opens a window called “Client Status” that lists all the devices and their IP addresses. Make note of the IP addresses of the devices you want to exclude.

Important side note: Generally the router will assign IP addresses on the fly and the IP addresses listed above can potentially change. If you want to ensure this does not happen then:

[1] Under "Advanced Settings" on the left bar click "LAN".
[2] Click the "DHCP Server" tab.
[3] Under "Manually Assigned IP around the DHCP list (Max Limit : 128)" you can assign fixed IP addresses to the devices you want to exclude.
[4] Click the red arrow next to the empty window labeled "MAC Address" and you should see a list of all your connected devices.
[5] Choose the device you want to exclude from the VPN. Click the plus arrow.
[6] Repeat until you have selected all the devices you want to exclude from the VPN.
[7] Now these devices will have permanent IP addresses, and you do not have to worry that they may change and mess up your script in the future.

Finally, for each device you want to exclude, at the end of the script in your text file, put in a separate line for each, with the correct IP address filled in:

```
iptables -t mangle -A PREROUTING -i br0 -m iprange --src-range 192.168.1.xxx -j MARK --set-mark 1
```

At this point, you should have the script in a text file in notepad, and the final lines of the script should have the IP addresses of all the devices you want to exclude from the VPN. There should be one of the above lines for each IP address. Save your file as a text file on your desktop or wherever. Then make a COPY of the text file and rename it "openvpn-event" WITH NO EXTENSION!!!

Step D: Copy Script to Router and Run Script

This last step is kind of the trickiest of all because you have to log on to your router using a different program and use this program to copy the script and run it. This information is also taken from the post by wizin that is here.

A prelude to this is you need to have SSH enabled on your router. To enable SSH on Merlin do this:

[1] Under "Advanced Settings" on the left bar click "Administration".
[2] Under "Miscellaneous" make sure these are set as follows:
   a. Enable SSH = Yes
   b. Allow SSH Port Forwarding = No
   c. SSH service port = 22
   d. Allow SSH access from WAN = No
   e. Allow SSH password login = Yes
   f. Enable SSH Brute Force Protection = No
[3] If you changed any of the above, click "APPLY" when all done.

To copy the script you need to:

[1] Download a Software like WinSCP.
[2] Install WinSCP then start it up.
[3] To make a new connection fill in the fields as follows:
    a. File Protocol – SCP (**NOTE MAKE SURE IT IS SCP!!!)**
    b. Hostname: 192.168.1.1
    c. Username/Password: Whatever you use to login to the router
    d. Port 22
[4] Save and then hit Login
[5] This will log you into your router. There will be two folder trees. On the right is your router folder tree.
[6] In the router folder tree, you heed to go upto the root folder where you see jffs folder under the root.
[7] Go Inside Folder then Go Inside Scripts Folder
[8] Place the openvpn-event file you created above in this folder (/jffs/scripts)
[9] Once the file is copied there, then right click > Properties > Change Octal to 0777
[10] Thats it
[11] Close WinSCP
[12] Reboot Router

At this point, the script should take effect and the devices you excluded should NOT be going through the VPN and everything else on your LAN should be going through the VPN.

If you want to add or subtract devices from the script, just modify the openvpn-event file. You can easily do this by logging back into the router, right clicking the file and chose "EDIT".

CONCLUSION

Whew! The price we pay for privacy. Oh well, now that it's well known that everybody is spying on everybody else over the internet, it's worth the effort to go through all the above to shut yourself off from prying eyes.

Ok, well, I hope I have saved somebody out there some time. It took me several weeks of searching and reading on the internet to figure out all of the above. Hopefully this step by step primer, as complicated as it may seem, saves somebody all the time and trouble I went through. And special thanks to all the people I linked to above who made all of this possible.

```
#!/bin/sh

sleep 2

touch /tmp/000wanstarted


# This code goes in the WAN UP section of the Tomato GUI.
# This code based on the contributions from this thread:
#  http://www.linksysinfo.org/index.php?threads/route-only-specific-ports-through-vpn-openvpn.37240/
#
# And from material in these articles:
#  http://linux-ip.net/html/adv-multi-internet.html
#  http://fedorasolved.org/Members/kanarip/iptables-howto
#
# This script configures "selective" VPN routing. Normally Tomato will route ALL traffic out
# the OpenVPN tunnel. These changes to iptables allow some outbound traffic to use the VPN, and some
# traffic to bypass the VPN and use the regular Internet instead.
#


#  To list the current rules on the router, issue the command:
#      iptables -t mangle -L PREROUTING
#
#  Flush/reset all the rules to default by issuing the command:
#      iptables -t mangle -F PREROUTING
#

#
# First it is necessary to disable Reverse Path Filtering on all
# current and future network interfaces:
#



for i in /proc/sys/net/ipv4/conf/*/rp_filter ; do
  echo 0 > $i
done

#
# Delete and table 100 and flush any existing rules if they exist.
#
ip route flush table 100
ip route del default table 100
ip rule del fwmark 1 table 100
ip route flush cache
iptables -t mangle -F PREROUTING

#
# Copy all non-default and non-VPN related routes from the main table into table 100.
# Then configure table 100 to route all traffic out the WAN gateway and assign it mark "1"
#
# NOTE: Here I assume the OpenVPN tunnel is named "tun11".
```

```
#
#
ip route show table main | grep -Ev ^default | grep -Ev tun11 \
  | while read ROUTE ; do
     ip route add table 100 $ROUTE
done
ip route add default table 100 via $(nvram get wan_gateway)
ip rule add fwmark 1 table 100
ip route flush cache


#
# Define the routing policies for the traffic. The rules will be applied in the order that they
# are listed. In the end, packets with MARK set to "0" will pass through the VPN. If MARK is set
# to "1" it will bypass the VPN.
#
# EXAMPLES:
#
#  All LAN traffic will bypass the VPN (Useful to put this rule first, so all traffic bypasses the VPN and you can
configure exceptions afterwards)
#    iptables -t mangle -A PREROUTING -i br0 -j MARK --set-mark 1
#  Ports 80 and 443 will bypass the VPN
#    iptables -t mangle -A PREROUTING -i br0 -p tcp -m multiport --dport 80,443 -j MARK --set-mark 1
#  All traffic from a particular computer on the LAN will use the VPN
#    iptables -t mangle -A PREROUTING -i br0 -m iprange --src-range 192.168.1.2 -j MARK --set-mark 0
#  All traffic to a specific Internet IP address will use the VPN
#    iptables -t mangle -A PREROUTING -i br0 -m iprange --dst-range 216.146.38.70 -j MARK --set-mark 0
#  All UDP and ICMP traffic will bypass the VPN
#    iptables -t mangle -A PREROUTING -i br0 -p udp -j MARK --set-mark 1
#    iptables -t mangle -A PREROUTING -i br0 -p icmp -j MARK --set-mark 1


# By default all traffic goes through the VPN
iptables -t mangle -A PREROUTING -i br0 -j MARK --set-mark 0

# Spotify explicitly by passes the VPN
#  All traffic from a particular computer on the LAN will use the VPN
    iptables -t mangle -A PREROUTING -i br0 -m iprange --src-range 192.168.1.xxx -j MARK --set-mark 1
    iptables -t mangle -A PREROUTING -i br0 -m iprange --src-range 192.168.1.xxx -j MARK --set-mark 1
```