The following details the procedure for establishing a site-to-site bridged VPN between two Linksys WRT54GL routers. Other routers should work just as easily, but other routers have not been tested by the author. To understand more about bridged VPNs, you can read Ethernet Bridging on www.openvpn.net.

Also, this document utilizes public key authentication rather than static key authentication. Static key authentication should work just as easily (possibly easier), but the author has not tried to establish a site-to-site VPN using static key authentication.

Contents

Install DD-WRT onto Your Routers

1. Download the latest stable release of the OpenVPN version of DD-WRT from DD-WRT. At the time this was written, the file you need to download is dd-wrt.v23_vpn_generic.bin. --Teccs 08:05, 4 Jun 2007 (CEST) For WRT54GL v1.1 router, I use dd-wrt.v23_vpn_wrt54g.bin. This is sp3.
2. For each router, follow the instructions for flashing the routers with the .bin file you downloaded. Both routers need to be flashed with the same file.

Generate Keys for Authentication

1. Download OpenVPN from HERE onto the computer that you are going to use to communicate with (i.e., configure) the routers.
2. Follow these instructions to install OpenVPN onto your computer.
3. Follow these instructions to generate keys for authentication. (Note: when generating the client key, create only one key and name it client, rather than client1.)

Configure the Server Router

Basic Setup

1. Log onto the first router, which will be our server router, using the GUI.
2. Go to Setup > Basic Setup.
3. Set the Local IP Address of the first router to 192.168.1.1 with a Subnet Mask of 255.255.255.0. (Of course, you are not required to use the network address 192.168.1.0. You can use any network address you like, but this tutorial will assume you are using this network.)

4. Make sure DHCP server is enabled.
5. Set the Start IP Address to 100.
6. Set Maximum DHCP Users to 50.
7. Set the Time Zone. (The time zone you select is not important to the success of this tutorial, but setting the **same** time zone on both routers **is** important.)
8. Click Save Settings.

Startup Script

1. Go to Administration > Commands
2. Paste the following into the Command Shell box:

```
cd /tmp
openvpn --mktun --dev tap0
brctl addif br0 tap0
ifconfig tap0 0.0.0.0 promisc up

echo "
# Tunnel options
mode server       # Set OpenVPN major mode
proto udp         # Setup the protocol (server)
port 1194         # TCP/UDP port number
dev tap0          # TUN/TAP virtual network device
keepalive 15 60   # Simplify the expression of --ping
daemon            # Become a daemon after all initialization
verb 3            # Set output verbosity to n
comp-lzo          # Use fast LZO compression

# OpenVPN server mode options
client-to-client  # tells OpenVPN to internally route client-to-client traffic
duplicate-cn      # Allow multiple clients with the same common name

# TLS Mode Options
tls-server        # Enable TLS and assume server role during TLS handshake
ca ca.crt         # Certificate authority (CA) file
dh dh1024.pem     # File containing Diffie Hellman parameters
cert server.crt   # Local peer's signed certificate
key server.key    # Local peer's private key
" > openvpn.conf

echo "
-----BEGIN CERTIFICATE-----
INSERT YOUR ca.crt HERE
-----END CERTIFICATE-----
" > ca.crt
echo "
-----BEGIN RSA PRIVATE KEY-----
INSERT YOUR server.key HERE
-----END RSA PRIVATE KEY-----
" > server.key
chmod 600 server.key
echo "
-----BEGIN CERTIFICATE-----
INSERT YOUR server.crt HERE
-----END CERTIFICATE-----
" > server.crt
echo "
-----BEGIN DH PARAMETERS-----
INSERT YOUR dh1024.pem HERE
-----END DH PARAMETERS-----
```

```
" > dh1024.pem

sleep 5
ln -s /usr/sbin/openvpn /tmp/myvpn
/tmp/myvpn --config openvpn.conf

route add -net 192.168.1.0/24 dev br0
```

1. Replace the "INSERT YOUR [FILE] HERE" text with the appropriate text from the .crt or .key files you generated during the Generate Keys for Authentication step.
2. Click Save Startup
3. Paste the following into the Command Shell box:

```
/usr/sbin/iptables -I INPUT -p udp --dport 1194 -j ACCEPT
```

1. Click Save Firewall
2. Reboot the router

Configure the Client Router

Basic Setup

1. Log onto the second router, which will be our client router, using the GUI.
2. Go to Setup > Basic Setup.
3. Set the Local IP Address of the second router to 192.168.1.254 with a Subnet Mask of 255.255.255.0. (Of course, if you used a different network address and subnet mask when setting up the server router, use that same address and mask for this router.)
4. Make sure DHCP server is enabled.
5. Set the Start IP Address to 50. NOTE: All the IP addresses of the client LAN and the server LAN must all be unique. You can't have 192.168.1.100 on the client LAN and 192.168.1.100 on the server LAN. In this case, all would be 192.168.1.x. Router IP addresses must also be different but must be 192.168.1.x as well. If you have multiple routers as simultaneous clients, all client's LAN IP addresses must be unique in the same way. Watch out that your DHCP assignments cannot create duplicate addresses.
6. Set Maximum DHCP Users to 50.
7. Set the Time Zone to the same time zone you set on the first (server) router.
8. Click Save Settings.

Startup Script

1. Go to Administration > Commands
2. Paste the following into the Command Shell box:

```
cd /tmp
ln -s /usr/sbin/openvpn /tmp/myvpn
./myvpn --mktun --dev tap0
brctl addif br0 tap0
ifconfig tap0 0.0.0.0 promisc up
sleep 5

echo "
client
daemon
```

```
dev tap0
proto udp
remote xxx.xxx.xxx.xxx 1194
resolv-retry infinite
nobind
persist-key
persist-tun
ca ca.crt
cert client.crt
key client.key
ns-cert-type server
comp-lzo
verb 3
" > /tmp/client.conf

echo "
-----BEGIN CERTIFICATE-----
INSERT YOUR ca.crt HERE
-----END CERTIFICATE-----
" > /tmp/ca.crt

echo "
-----BEGIN RSA PRIVATE KEY-----
INSERT YOUR client.key HERE
-----END RSA PRIVATE KEY-----
" > /tmp/client.key
chmod 600 /tmp/client.key

echo "
-----BEGIN CERTIFICATE-----
INSERT YOUR client.crt HERE
-----END CERTIFICATE-----
" > /tmp/client.crt

./myvpn --config client.conf

route add -net 192.168.1.0/24 dev br0
```

UPD from hryamzik: This script didn't work for me. I've composed the folowing:

```
cd /tmp
ln -s /usr/sbin/openvpn /tmp/myvpn
./myvpn --mktun --dev tap0
brctl addif br0 tap0
ifconfig tap0 promisc

cat <<EOF> /tmp/up.sh
/sbin/ifconfig tap0 0.0.0.0
EOF

chmod +x /tmp/up.sh

echo "
daemon            # Become a daemon after all initialization
client
dev tap0
proto udp
remote domain.comt 1194
resolv-retry infinite
nobind
persist-key
```

```
persist-tun
ca ca.crt
cert client.crt
key client.key
comp-lzo
verb 3
mssfix 1200
up \"/tmp/up.sh\"
" > client.conf

echo "
-----BEGIN CERTIFICATE-----

-----END CERTIFICATE-----
" > ca.crt
echo "
-----BEGIN RSA PRIVATE KEY-----

-----END RSA PRIVATE KEY-----
" > /tmp/client.key
chmod 600 /tmp/client.key

echo "
-----BEGIN CERTIFICATE-----

-----END CERTIFICATE-----
" > /tmp/client.crt

./myvpn --config client.conf

route add -net 192.168.1.0/24 dev br0
```

Be carefull with "route add", I had some problems with it when used wrong values. I did **not** updated iptables.

End of upd by hryamzik.

Start of upd by strfr

The original script above the hryamzik's one works for me well on Asus WL-500GP v1.0 with mega build 10949M NEWD Eko, I have not tried the hryamzik's one.

End of upd by strfr

1. Replace the "INSERT YOUR [FILE] HERE" text with the appropriate text from the .crt or .key files you generated during the Generate Keys for Authentication step.
2. Replace the xxx.xxx.xxx.xxx text with the public IP address (or DNS name) of your first (server) router. (Note: if your Internet Service Provider gives your routers dynamic IP addresses (and they probably do), you might want to look into Dynamic DNS services such as DynDNS or No-IP.com. You can start to learn more about dynamic DNS by looking at the help from Setup > DDNS in the DD-WRT GUI.)
3. Click Save Startup
4. Reboot the router (i.e., In the GUI, click Administration > Management and click Reboot Router at the bottom)

Storing config and key files on /jffs by robertut

If you hate the long startup scripts, or want to make sure your router won't brick due to large nvram text, you can store everything on the /jffs partition of your router, if you enable it. The capacity of this can be several MB, while you only need 6-10KB for all the openvpn config files.

All you need to do is to create a simple startup file, which will also be on the /jffs.

Create a file called vpn.ipup on your PC, with similar content (change to suit your needs):

```
#!/bin/sh
echo "$(date) - wan restart detected, restarting vpn service" >>/jffs/wanrestart.log
killall yourtapclient
rm /jffs/yourtapclient
ifconfig tap0 down
brctl delif br0 tap0
openvpn --rmtun --dev tap0
sleep 3
/usr/sbin/openvpn --mktun --dev tap0
brctl addif br0 tap0
ifconfig tap0 0.0.0.0 promisc up
ln -s /usr/sbin/openvpn /jffs/yourtapclient
/jffs/yourtapclient --daemon --config /jffs/config.ovpn
```

In the web interface:

1. Go to Administration > Management > JFFS2 Support and enable JFFS2 and Clean JFFS2, hit Apply
2. Go to Services > NAS > ProFTPD and enable it, select /jffs, enable Allow Write, in the User Password List type "admin admin" (without quotes), hit Apply
3. Use your favourite FTP client program, and connect to the router port 21, using admin/admin as user and pass. Copy over your config.ovpn config file, the ca, crt, key files. Make folder /etc/config, and drop vpn.ipup inthere.
4. Log in via telnet or ssh into your router, type "chmod 755 /jffs/etc/config/vpn.ipup" to make your startup script executable.

The vpn.ipup startup file, located in /jffs/etc/config folder will be executed by dd-wrt every time the WAN or PPP interface goes up, after the firewall. This ensures, besides that it starts only when internet access is on after boot, that if your PPP connection drops and re-establishes, openvpn is also restarted. The script's first line generates a log in /jffs/wanrestart.log, to have a clue when, what and why happened. You may want to delete this line, to prevent filling up /jffs, if you have often reboots or disconnections.

For security reasons, you may want to disable the ProFTPD service after you finish, as it was only needed to transfer the configuration files to the router.

Blocking DHCP requests through the VPN

Since there will be multiple DHCP servers on one bridged network, this may result in clients receiving IP addresses from another site's DHCP server. In most cases this will be an undesirable side effect, as the client's internet traffic will be routed through the VPN instead of its own local gateway.

This script will load the necessary ebtable modules that will allow the router to filter the DHCP broadcast from being sent and received through the VPN tunnel. See this thread for more information about the script. I've placed this in all VPN end points to prevent any DHCP broadcasts from entering the tunnel.

Startup Script

1. Go to Administration > Commands
2. Click edit under Startup
3. Paste the following at the end of your current Startup script:

```
echo "begin-base64 644 -" > /tmp/ebt_ip.o.gz.u64
echo "H4sIADwAAAACA5VWz28bVRD+dteON20KG9eq3BJUB21UVy1m2+RQRCO5cX5w" >> /tmp/ebt_ip.o.gz.u64
echo "yMGHHpAQcr32Eq9wHMveIFAPNSkHDq7wIb1H5R+p0gpx7J8QtaH8EBfuSGbm" >> /tmp/ebt_ip.o.gz.u64
echo "vbdh8whEHelp9ps3883Mm/ecPFhZXzUMA7EYsPEPAv5Ikc1BdUGgIt7CWSQl" >> /tmp/ebt_ip.o.gz.u64
echo "i5ejFJY8B2VntjrAecIFPByyLUM2B4cjA43RGDveN6g4HJMD7CxeH9nvo5KT" >> /tmp/ebt_ip.o.gz.u64
echo "XK9HNva9KsUxPo/fiKvyyMZ7N5hrmrgmVUxHcV0hrmm8OrK3FBfbJmC6FnyK" >> /tmp/ebt_ip.o.gz.u64
echo "M92P4At7AUuPBsRZwHMPWBq+TXUOHBN5DJzZsg8T5uM8DnfHqHi4ZMHMmpjL" >> /tmp/ebt_ip.o.gz.u64
echo "+7hWaCKNQW62CFzCy91rTgUW1VSg2gykXdZpwgae3FpDWeRap9oc6qeCPYH5" >> /tmp/ebt_ip.o.gz.u64
echo "+6LqDZQnd0qeCyqPqWI4D5+vQzXHua6rXBvqTIsqlzzffS9LfiZpORc+y8Oh" >> /tmp/ebt_ip.o.gz.u64
echo "oWYD/Dycgj90nWXqaUXkuE887B/bc1gRfGzL8YwdiBmniIf5TMU1hV+GjohZ" >> /tmp/ebt_ip.o.gz.u64
echo "QwbLgsvGgcX+sd1K2A2aCe/xtzybH76fIdsNmtPv47Rr49muTfh9NbcU9kXd" >> /tmp/ebt_ip.o.gz.u64
echo "V1W/WboLT8lvjIee6wR0ZtLOtbnOipgN4x8pBfN58L0zYl47Xh73HLbpdyR7" >> /tmp/ebt_ip.o.gz.u64
echo "dEd21N1l31e78bwpXtTmqP14RjnyUXMS+2yXve9x5beBi7Q+vGKjQL1+655s" >> /tmp/ebt_ip.o.gz.u64
echo "Oy5hF6fKx+a/bX9l8ARvIO/SuvwfXMn9gvX/+wun7LfDRtDpB4tr1XV8EfQ6" >> /tmp/ebt_ip.o.gz.u64
echo "Qbv2ZdDrh1udxZulhdK8x86l/tebUd0nHfWkbsVfvaBdioKvIvnVrEd1lPx+" >> /tmp/ebt_ip.o.gz.u64
echo "n+FG2Pl8S9q7zR5Km83A394o1f1w/iahrabYfwO5LN4aMKFwmVY10Z96wHiH" >> /tmp/ebt_ip.o.gz.u64
echo "1mQi7hMb4LFNqfiUuOvAVYVjipElOXW+OY3vJwrOKz4rwfeB0jHfC+3sY74F" >> /tmp/ebt_ip.o.gz.u64
echo "2UPXTPjlE37Mx/iOqi+WX8lvoPmxLGr1Hdj8VmR9mUR9qxpfK3W8vnjvrvo2" >> /tmp/ebt_ip.o.gz.u64
echo "E37uCXmnE72y0B8XfHYCX5KLZYaKOiB9TtWdUmczqfHRm8GnJ/DpYhyr7c9x" >> /tmp/ebt_ip.o.gz.u64
echo "cs/SoixYGk5reFLDZzQ8oeGM8OB3mBO/fwb99sh3F+M57Q7Ma3hVvyOBH9XC" >> /tmp/ebt_ip.o.gz.u64
echo "bqmBsBNGNXop2+0AjXZQ72x3Y1irRa2wHyOO4BfXj4JebbMeNVrCtN3RjOLu" >> /tmp/ebt_ip.o.gz.u64
echo "pUx5j1jPGFJ/p/QLpfdMqelHGGm6SEXSE1OyzzQtjzGtW4xpkLcZn5N98/6y" >> /tmp/ebt_ip.o.gz.u64
echo "2i/y7InnutKc31T/qjAuKF1W+p7SfwMbOFhxAAkAAA==" >> /tmp/ebt_ip.o.gz.u64
echo "====" >> /tmp/ebt_ip.o.gz.u64
uudecode /tmp/ebt_ip.o.gz.u64 | gunzip -cd > /tmp/ebt_ip.o

sleep 15 ; insmod ebtables ; insmod ebtable_filter ; insmod /tmp/ebt_ip.o &
sleep 25 ; ebtables -I INPUT -i tap0 -p IPv4 --ip-protocol udp --ip-destination-port 67:68 -j DRO
sleep 5 ; ebtables -I OUTPUT -o tap0 -p IPv4 --ip-protocol udp --ip-destination-port 67:68 -j DRO
```

1. Click Save Startup
2. Reboot the router

After rebooting you can test to see if the script successfully loaded. Remember to allow time for script to run as it has several sleep commands. These are necessary for the script to run properly, and will take a minute or two to fully load. Run the following command through ssh or telnet:

```
ebtables -L
```

If the Startup script ran successfully, it should return the following:

```
Bridge table: filter

Bridge chain: INPUT, entries: 1, policy: ACCEPT
-p IPv4 -i tap0 --ip-proto udp --ip-dport 67:68 -j DROP

Bridge chain: FORWARD, entries: 0, policy: ACCEPT

Bridge chain: OUTPUT, entries: 1, policy: ACCEPT
-p IPv4 -o tap0 --ip-proto udp --ip-dport 67:68 -j DROP
```

Your Site-To-Site VPN bridge should now only distribute IP addresses locally, and route all your client's internet traffic through their local gateway.

note 1: This has been tested using the wrt54gl. Some modification may be necessary depending on your router, as bootup sequences and timing differ. Mainly the sleep commands seem to be the key here, and may have to be adjusted to make things work.

note 2: It appears in the forums that x86 and 2.6 builds do not work with this method, and require alternate methods (possibly easier) to enable ebtables.