OpenVPN - Site-to-Site routed VPN between two routers

From DD-WRT Wiki

Info: Last edit by author was on Jan 2011.
The following details the procedure for establishing a site-to-site routed VPN between two DD-WRT/vpn image enabled routers. The author tried the config on two Linksys WRT54GL(v1.1)

Should you have any questions, please don't hesitate to contact the author on wzaatar at gmail dot com.

Contents

Procedure Summary

1. Router Preparation.
2. Install OpenVPN on your PC and generate your secret key.
3. Configure one router as the server.
4. Configure the second router as the client.
5. Test the VPN connection.
6. Advanced Configuration: Multiple routed networks.

Router Preparation

These VPN scripts have been tested starting v23 and have been confirmed to work in v24 of DD-WRT. Before proceeding, you need to download the VPN-flavoured version of DD-WRT from the DD-WRT Download Page (http://www.dd-wrt.com/dd-wrtv2/downloads.php) .

Due to the fact that most of us have DHCP-assigned dynamic IPs, you are also recommended to create a dynamic dns host for the server router. More information on this procedure is available here (http://www.dd-wrt.com/wiki/index.php/DDNS_-_How_to_setup_Custom_DDNS_settings_using_embedded_inadyn_-_HOWTO) .

Finally, make sure that your two routers are **not** distributing an overlapping IP subnet range. Usually, all routers come preconfigured with a 192.168.1.0 DHCP range distribution. Since you are doing **routed** configurations, you need to change the 192.168.1.0 subnet to another one. The easiest way is to adopt a sequential assignment:

Server side: 192.168.1.0 Client1 side: 192.168.2.0 Client2 side: 192.168.3.0 etc...

This way, when your internal networks communicate with each other, they don't overlap and you don't end up having miscommunication.

If you are looking for a bridged configuration, you'd better check this Wiki page instead (http://www.dd-wrt.com/wiki/index.php/OpenVPN_-_Site-to-Site_Bridged_VPN_Between_Two_Routers) .

Secret Key Generation

Prior to configuring your routers, you need to create a shared secret key. This key will be used to authenticate and encrypt your site to site communication.

Start by downloading the latest OpenVPN package from OpenVPN's main site (http://www.openvpn.net/download.html) . Install the package (Usually gets installed in C:\Program Files\OpenVPN if you are running

Windows). Now, get a command prompt and issue the following command from the OpenVPN directory:

```
openvpn --genkey --secret static.key
```

This will create a text file named 'static.key'. Opening it in Notepad, or any text editor will get you an output similar to the following one:

```
#
# 2048 bit OpenVPN static key
#
-----BEGIN OpenVPN Static key V1-----
aeb68165149e096d8f04252dd22fe67d
dd15d8c87e8a577c5c14ebd1ef0bf0b6
0e1d652f91fe66ed3774505e641936dd
458a6db60fb36b969d8bcd37803cf1d3
6d49383ec2daa1d2ae70e3ca49b950a4
bba985940e5e4a15fac702cbcf47f9d0
39f7939980bbb63d2964bb6216471162
0a519fe25d1e0d48044a1ad85dc94758
af6f7b7c52ccaaefa3d013fcbf621366
5ea18d9dc36c3b2a9ac277a9903998fe
45e10b0f79fd443727c3f30278981b3d
0fa525ad843645b4acc28969450bd601
4ce774aba0e830149489dc1592741580
fbd3cd24cc7baa68e06b3e3aedae2565
a36b8a3f687ddbb78411740d755249cf
45c0617c215b66eabc72f60f47b32c64
-----END OpenVPN Static key V1-----
```

Warning: Don't go lazy and copy the above, doing so will jeopardize your secure connection, recreate the file from scratch.

Server Configuration

Using Notepad or any text editor, create the following two configurations:

Config 1

```
# Move to writable directory and create scripts
cd /tmp
ln -s /usr/sbin/openvpn /tmp/myvpn

# Config for Site-to-Site SiteA-SiteB
echo "
proto udp
port 2000
dev tun0
secret /tmp/static.key
verb 3
comp-lzo
keepalive 15 60
daemon
" > SiteA-SiteB.conf

# Config for Static Key
echo "
-----BEGIN OpenVPN Static key V1-----
.............................................
...YOUR SECRET KEY TEXT SHOULD BE PASTED HERE...
```

```
.............................................
-----END OpenVPN Static key V1-----
" > static.key

# Create interfaces
/tmp/myvpn --mktun --dev tun0
ifconfig tun0 10.0.0.1 netmask 255.255.255.0 promisc up

# Create routes
route add -net OTHERSUBNET netmask 255.255.255.0 gw 10.0.0.2

# Initiate the tunnel
sleep 5
/tmp/myvpn --config SiteA-SiteB.conf
```

Warning: Watch out for the OTHERSUBNET chunk, you should replace it with your client network's subnet (for example: 192.168.2.0 or 192.168.3.0).

Also, do note that the static key that was created in the previous step should be pasted in the appropriate section, right after the '*echo* text.

Now, create a second configuration with the following text.

Config 2

```
# Open firewall holes
iptables -I INPUT 2 -p udp --dport 2000 -j ACCEPT
iptables -I FORWARD -i br0 -o tun0 -j ACCEPT
iptables -I FORWARD -i tun0 -o br0 -j ACCEPT
```

Now, go to your Router configuration interface, click on 'Administration' then 'Commands'. Paste your 'Config 1' in your 'Startup' section and you 'Config 2' in your 'Firewall' section.

You're done with the server configuration!

Client Configuration

The client configuration is very similar to the server configuration, with a few small modifications.

Again, you need to create two configs:

Config 1

```
# Move to writable directory and create scripts
cd /tmp
ln -s /usr/sbin/openvpn /tmp/myvpn

# Config for Site-to-Site SiteA-SiteB
echo "
remote REMOTEADDRESS
proto udp
port 2000
dev tun0
secret /tmp/static.key
verb 3
```

```
comp-lzo
keepalive 15 60
daemon
" > SiteA-SiteB.conf

# Config for Static Key
echo "
-----BEGIN OpenVPN Static key V1-----
.............................................
...YOUR SECRET KEY TEXT SHOULD BE PASTED HERE...
.............................................
-----END OpenVPN Static key V1-----
" > static.key

# Create interfaces
/tmp/myvpn --mktun --dev tun0
ifconfig tun0 10.0.0.2 netmask 255.255.255.0 promisc up

# Create routes
route add -net OTHERSUBNET netmask 255.255.255.0 gw 10.0.0.1

# Initiate the tunnel
sleep 5
/tmp/myvpn --config SiteA-SiteB.conf
```

Warning: Watch out for the OTHERSUBNET chunk, you should replace it with your server network's subnet (for example: 192.168.1.0).

Also, do note that the static key that was created in the previous step should be pasted in the appropriate section, right after the 'echo text.

In addition to the above, and since this is your client, you need to replace the REMOTEADDRESS with your server's IP address or the dynamic DNS address you created in the previous Router Preparation section.

Now, create a second configuration with the following text.

Config 2

```
# Open firewall holes
iptables -I INPUT 2 -p udp --dport 2000 -j ACCEPT
iptables -I FORWARD -i br0 -o tun0 -j ACCEPT
iptables -I FORWARD -i tun0 -o br0 -j ACCEPT
```

Now, go to your Router configuration interface, click on 'Administration' then 'Commands'. Paste your 'Config 1' in your 'Startup' section and you 'Config 2' in your 'Firewall' section.

You're done with the client configuration!

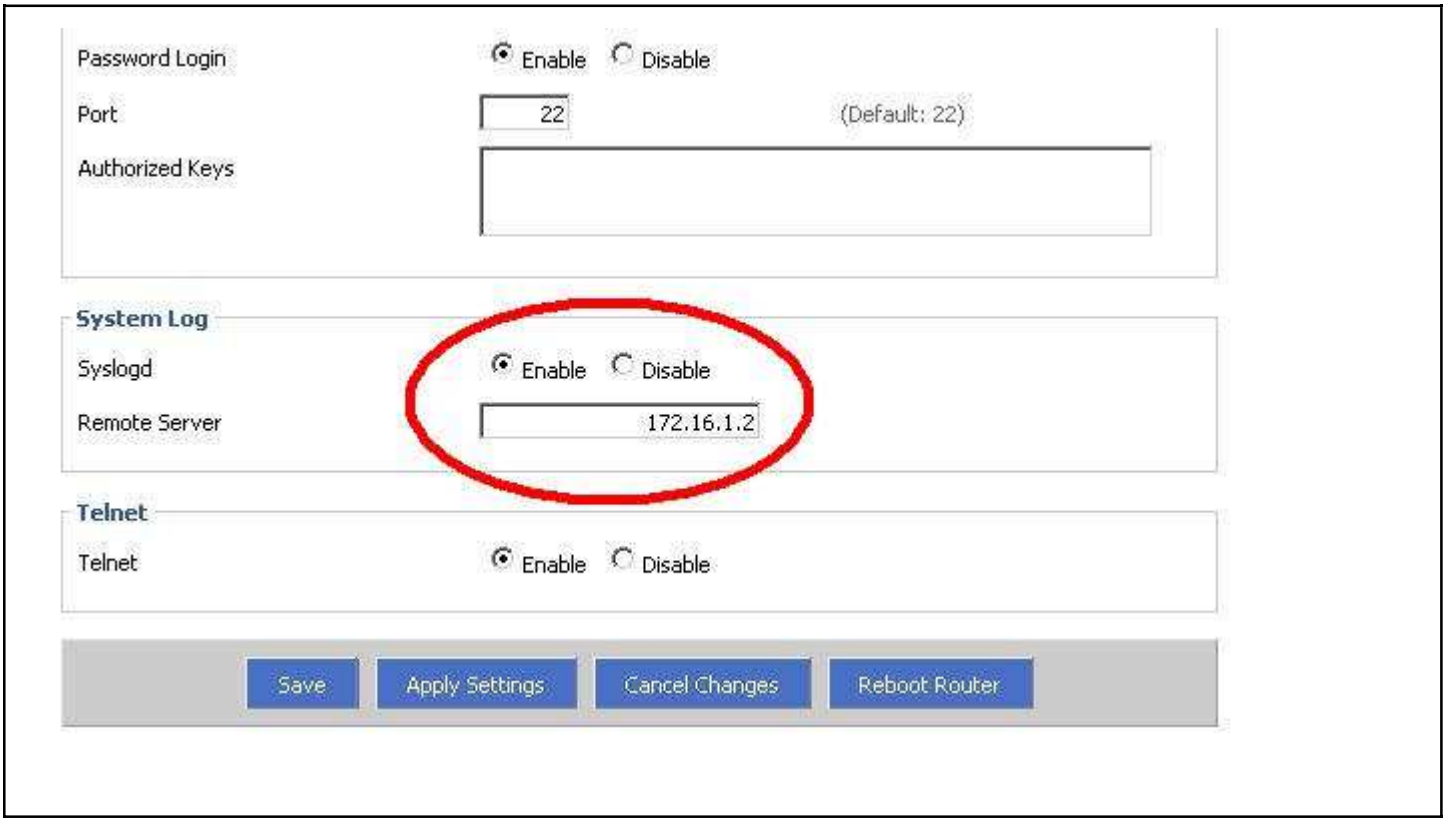
VPN Tests

I am getting many emails from people asking for this section, so I'll try to add some meat. Let me know what you think about it.

Due to the fact that our routed VPN configuration is not 'natively' supported by DD-WRT, but rather an ad-hoc one. There is no direct way to get information through the router's web interface. Instead, I propose the following suggestion:

Using Syslog

Syslogging is an excellent way to get all sorts of information on your routers. In addition to OpenVPN alerts and tunnel stats, you can get router access, DHCP usage, etc... Pretty much everything using Syslogging. *Unless a 'log' stanza is present in the openvpn config file, openvpn will dump all the logging to syslog by default.*
The recipe:
1. Open your browser and connect to your router's interface. Click on 'Services' and scroll all the way down (I am assuming DD-WRT RC4 and above has been flashed on your router).
2. Locate the 'System Log' section and click on 'Enable' next to **syslogd**. This will add one more space called 'Remote Server' right underneath 'syslogd', type in your computer's IP address or preferably any other workstation that can successfully ping the router.

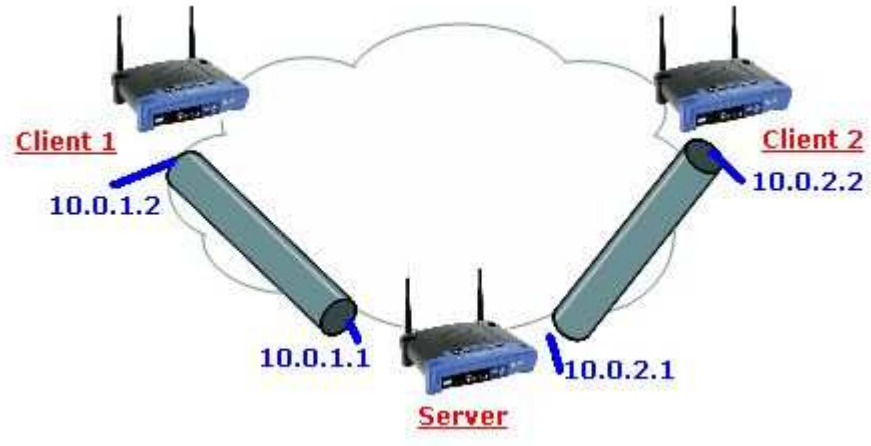


3. Download a Syslog Daemon and viewer, there is an excellent freeware version available at Kiwi Enterprises' website that I'm linking here (http://www.kiwisyslog.com/kiwi-syslog-daemon-overview/) .
4. Install the syslog daemon and **don't forget to enable it**, this can be done in the Kiwi Sylog Application's toolbar: Click on the 'Manage' menu then successively select 'Install the Service' then 'Start the Service'. You can verify that the syslog service is operational by pressing 'Ctrl+T', this should send a test message on the console.
5. To test it immediately, force your router to reboot and voila! You should see all messages coming from your router (including VPN initiation and communication messages) appearing on your Kiwi Syslog console page. If you are creating several VPN connections, you can configure all your routers to redirect their syslog to one PC, allowing you to easily monitor all your networks from one site! I'm currently managing 4 VPNed locations and this tool has proved to be **invaluable** to me.

Advanced Configuration: Multiple routed networks

Warning: This section is not for the faint-hearted people. Please read carefully and email me should you have any questions/comments/thoughts. Wiki is all about teamplay!

Let's assume we need to configure a 3-sites VPN connection as per the following figure:



Attention: I tried keeping this technique simple and didn't use Certificates/CAs. Should you be interested in more complex scenarios, I do consultancy work and would gladly assist you.

You need to first start by duplicating the above Client configuration on the two 'Client1' and 'Client2' routers. Pay extra attention to the IPs and IP ranges you are using and write down your configs. In essence, both clients will have pretty much the same configuration with one minor change. Since both will be connecting to the same server, you cannot use the same **port** number for both clients, so we will be giving port 1999 for the first client and 2000 for the second client.
Also, we need to tell Client1 how to reach Client2's subnet and vice-versa. This means including a second routing entry in our configuration. As such, our configurations will look pretty much like the following:

Client1 Configuration

Client1 -- Startup

```
# Move to writable directory and create scripts
cd /tmp
ln -s /usr/sbin/openvpn /tmp/myvpn

# Config for Site-to-Site Client1-Server
echo "
remote REMOTEADDRESS
proto udp
port 2000
dev tun0
secret /tmp/static.key
verb 3
comp-lzo
keepalive 15 60
daemon
" > Client1-Server.conf

# Config for Static Key
echo "
-----BEGIN OpenVPN Static key V1-----
.............................................
...YOUR SECRET KEY TEXT SHOULD BE PASTED HERE...
.............................................
-----END OpenVPN Static key V1-----
" > static.key
```

```
# Create interfaces
/tmp/myvpn --mktun --dev tun0
ifconfig tun0 10.0.1.2 netmask 255.255.255.0 promisc up

# Create routes
route add -net SERVERINTERNALSUBNET netmask 255.255.255.0 gw 10.0.1.1
route add -net CLIENT2INTERNALSUBNET netmask 255.255.255.0 gw 10.0.1.1

# Initiate the tunnel
sleep 5
/tmp/myvpn --config Client1-Server.conf
```

Client1 -- Firewall

```
# Open firewall holes
iptables -I INPUT 2 -p udp --dport 2000 -j ACCEPT
iptables -I FORWARD -i br0 -o tun0 -j ACCEPT
iptables -I FORWARD -i tun0 -o br0 -j ACCEPT
```

Client2 Configuration

Client2 -- Startup

```
# Move to writable directory and create scripts
cd /tmp
ln -s /usr/sbin/openvpn /tmp/myvpn

# Config for Site-to-Site Client2-Server
echo "
remote REMOTEADDRESS
proto udp
port 1999
dev tun0
secret /tmp/static.key
verb 3
comp-lzo
keepalive 15 60
daemon
" > Client2-Server.conf

# Config for Static Key
echo "
-----BEGIN OpenVPN Static key V1-----
............................................
...YOUR SECRET KEY TEXT SHOULD BE PASTED HERE...
............................................
-----END OpenVPN Static key V1-----
" > static.key

# Create interfaces
/tmp/myvpn --mktun --dev tun0
ifconfig tun0 10.0.2.2 netmask 255.255.255.0 promisc up

# Create routes
route add -net SERVERINTERNALSUBNET netmask 255.255.255.0 gw 10.0.2.1
route add -net CLIENT1INTERNALSUBNET netmask 255.255.255.0 gw 10.0.2.1

# Initiate the tunnel
sleep 5
/tmp/myvpn --config Client2-Server.conf
```

Client2 -- Firewall

```
# Open firewall holes
iptables -I INPUT 2 -p udp --dport 1999 -j ACCEPT
iptables -I FORWARD -i br0 -o tun0 -j ACCEPT
iptables -I FORWARD -i tun0 -o br0 -j ACCEPT
```

Server Configuration

As for the server, we need to perform three modifications:
1. Tell the server to listen to 2 connections, one on port 1999 and the other on port 2000. This can be done by running the openvpn daemon twice (As you will see in the coming configuration, we will be creating two TUN interface, called 'tun0' and 'tun1').
2. Make sure to add a route to the two clients.
3. Allow Client-to-Client connection in the Firewall configuration script.

Server -- Startup

```
# Move to writable directory and create scripts
cd /tmp
ln -s /usr/sbin/openvpn /tmp/myvpn

# Config for Site-to-Site Server-Client1
echo "
proto udp
port 2000
dev tun0
secret /tmp/static.key
verb 3
comp-lzo
keepalive 15 60
daemon
" > Server-Client1.conf

# Config for Site-to-Site Server-Client2
echo "
proto udp
port 1999
dev tun1
secret /tmp/static.key
verb 3
comp-lzo
keepalive 15 60
daemon
" > Server-Client2.conf

# Config for Static Key
echo "
-----BEGIN OpenVPN Static key V1-----
............................................
...YOUR SECRET KEY TEXT SHOULD BE PASTED HERE...
............................................
-----END OpenVPN Static key V1-----
" > static.key

# Create interfaces
/tmp/myvpn --mktun --dev tun0
/tmp/myvpn --mktun --dev tun1
ifconfig tun0 10.0.1.1 netmask 255.255.255.0 promisc up
ifconfig tun1 10.0.2.1 netmask 255.255.255.0 promisc up

# Create routes
```

```
route add -net CLIENT1INTERNALSUBNET netmask 255.255.255.0 gw 10.0.1.2
route add -net CLIENT2INTERNALSUBNET netmask 255.255.255.0 gw 10.0.2.2

# Initiate the tunnel
sleep 5
/tmp/myvpn --config Server-Client1.conf
/tmp/myvpn --config Server-Client2.conf
```

Server -- Firewall

```
# Open firewall holes for Client1
iptables -I INPUT 2 -p udp --dport 2000 -j ACCEPT
iptables -I FORWARD -i br0 -o tun0 -j ACCEPT
iptables -I FORWARD -i tun0 -o br0 -j ACCEPT

# Open firewall holes for Client2
iptables -I INPUT 2 -p udp --dport 1999 -j ACCEPT
iptables -I FORWARD -i br0 -o tun1 -j ACCEPT
iptables -I FORWARD -i tun1 -o br0 -j ACCEPT

# Allow Forwarding packets between Client1 and Client2
iptables -I FORWARD -i tun0 -o tun1 -j ACCEPT
iptables -I FORWARD -i tun1 -o tun0 -j ACCEPT
```

Passing DNS requests over your Routed VPN configuration

This section would not have been possible without the augmented work of Jean-Marc L.

A question that comes often once we get our routed network up is DNS resolution. Ideally, you would like to have all your machines on all networks to be able to "speak" to each other using DNS and not just via their IP addresses. This section describes the procedure you use to integrate DNS resolution in your routed VPN structure.

Let's say we have 2 subnets. Subnet1, with network ID 192.168.1.0/24 served by Router1 (ip: 192.168.1.1) and Subnet2, with network ID 192.168.2.0/24 served by Router2 (ip: 192.168.2.1).

We would like to configure the two subnets as two domains: Domain1 and Domain2, assigning Domain1 to Subnet1 and Domain2 to Subnet2. Our target is to get Router1 to transfer all requests for Domain2 to Router2 and Router2 to transfer all requestes for Domain1 to Router1.

Router1 DNS setup

First we will need to configure the DNSMasq options on Router1. Go to the Services configuration page 'Services' -> 'Services' and perform the following modifications:

```
- Set the DHCPserver to use domain on LAN & WLAN.
- Set the LAN domain to be domain1.
- Enable DNSMasq.
- Enable Local DNS.
- No DNS Rebind -- Disable ***NOTE
```

- NOTE There are some options that may depend on your dd-wrt build. Two options in particular you need to be concerned with that will effect the ability of your router to receive

DNS lookups from your openvpn-linked private network router: stop-dns-rebind, rebind-domain-ok. Older builds (such as 13064) do not support rebind-domain-ok and have stop-dns-rebind disabled by default. These builds DNS will work fine. Mid time builds (such as 14896 mega) do not support rebind-domain-ok, but enable stop-dns-rebind by default and provide no gui interaction to disable it. These builds your router will not accept results from its peer and will not log the dropped query. The newest builds provide a radio button to disable stop-dns-rebind, and it must be selected to allow private nameserver responses. Looking in the source i believe still do not support rebind-domain-ok -- which is a shame, as this would very much help protect you against the type of attack that 'stop-dns-rebind' is supposed to protect you against. Hopefully in the future this will be included.

This will instruct your router to use local domains when resolving adresses and turn the local DNS service on the router on.

Next we need to configure Router1 to act as a DNS on both subnets (so it will answer Router2 requests as well). To perform this operation, you need to add the following options in the 'Additional DNSMasq Options' text box:

```
interface=br0,tun0
no-dhcp-interface=tun0
server=/domain2/192.168.2.1
```

The first line instructs DNSMasq to listen for request from the Subnet2 on the tunnel tun0. The second line ensures that the DHCP will not respond to remote subnet requests. And finally, the last line will instruct DNSMasq to redirect any requests for Domain2 entries to Router2.

In addition, we will also need to open port 53 by adding the 2 lines to the firewall section in our configuration. To do this, you need to go to 'Administration' --> 'Commands' and add the following lines to your firewall configuration:

```
iptables -I INPUT 1 -i tun0 -p tcp --dport 53 -j ACCEPT
iptables -I INPUT 1 -i tun0 -p udp --dport 53 -j ACCEPT
```

This will allow the firewall to pass DNS request from Subnet2 to Router1.

Finally, reboot rooter1.

Router2 DNS Setup

On Router2, you need to replicate the configuration you performed for Router1. This means that you need to add the same frewall rule:

```
iptables -I INPUT 1 -i tun0 -p tcp --dport 53 -j ACCEPT
iptables -I INPUT 1 -i tun0 -p udp --dport 53 -j ACCEPT
```

As well as: **(Watch out, it's domain2 and NOT domain1)**

```
- Set the DHCPserver to use domain on LAN & WLAN.
- Set the LAN domain to be domain2.
- Enable DNSMasq.
- Enable Local DNS.
```

Finally, you'll need to include the DNSMasq options, **watch out for the server line, the ip address is now 192.168.1.1 instead of 192.168.2.1**

```
interface=br0,tun0
no-dhcp-interface=tun0
server=/domain1/192.168.1.1
```

You should notice that the options are very similar to the ones in Router1, but in this case we are forwarding all requests to *.domain1 to Router1.

Again, reboot Router2.

Testing DNS

To test your configuration, simply go to your Router1 status page 'Status' -> 'LAN'. You should see the list of available hosts that have a DNS registration.

Next, go to a PC located on Subnet1 and try the following:

```
nslookup pconsubnet1.domain1   (Replace pconsubnet1 with the hostname of any PC on Subnet 1)
```

```
nslookup pconsubnet2.domain2   (Replace pconsubnet2 with the hostname of any PC on Subnet 2)
```

Both resolutions should work fine. You can go ahead and try the same operation from a PC located on Subnet 2.

The DNS query should return the correct IP adresses. Otherwise, check your configuration.

Troubleshooting / FAQ

I will be including in this section any troubleshooting questions I received as well as their solution (If I have one or if the sender found a fix for it).

DMZ feature is used on your DD-WRT router

(Thanks TJ T. for that one)
If you decide to run OpenVPN on your DD-WRT based router, make sure to disable any DMZ as the DMZ will override the usual port forwarding needed by your OpenVPN clients/server and would forward all connection requests to the DMZ host.

My tunnel is up but I cannot ping the remote endpoint

(Thanks Ben G. for that one)
Yes, this is normal if the router is set to be not 'pingable' (The option is set by default). To rectify this and allow your server-side and client-side hosts to ping both routers' endpoints while making sure that external hosts (Not belonging to your networks) still don't ping your routers' interfaces, add the following entry to your Firewall

section in **both** routers:

```
iptables -I INPUT 3 -i tun0 -p icmp -j ACCEPT
```

Great! How about internal remote administration through HTTP?

(Thanks Marc D. for that question)
Well, in that case you will have to do the same operation as in the previous issue (i.e. Cannot ping the remote endpoint) and add an extra iptables command in your Firewall section in **both** routers:

```
iptables -I INPUT 1 -i tun0 -p tcp --dport 80 -j ACCEPT
```

Why should I use a routed configuration and not a bridged configuration

Interesting question. Well, a bridged configuration will 'join' both networks together as one, same subnet, same IP range... Looks easier, but the problem here would be that all kinds of packets, including the infamous broadcasts will be traveling from one side of the network to the other, resulting in less-than-optimized usage of your precious bandwidth. On the other hand, a routed network will only send directed packets from one side of the network to the other.

I have a problem connecting my VPNs while using Chillispot, what's the issue?

(Thanks Chris A. for bringing this one up)
The problem here is that Chillispot insists on using 'tun0' as a communication tunnel. The easiest solution is to simply replace your 'tun0' with another tunnel ('tun2', 'tun3', etc...)
Also, you need to make sure that **both** your firewall and startup sections are updated accordingly.

DHCP Forwarder / DHCP Server feature of DD-WRT

Please take note that this VPN configuration will not work if your router(s) is/are set up as DHCP forwarders. They must be DHCP servers in order for the VPN to connect properly.

Remarks

This should get you right in business and activate Site-to-Site routing between all Clients and the Server. I'm personally running 12 simultaneous VPN connections using the above mentioned model without any issue whatsoever!

Please share your thoughts, comments and experiences!

Thanks - Wadih.

.....

Retrieved from "http://www.dd-wrt.com/wiki/index.php/OpenVPN_-_Site-to-Site_routed_VPN_between_two_routers"

Categories: Tunneling | Advanced tutorials