OpenWRT WireGuard Client Setup guide using Luci

Latest iteration of this guide can be found at:
https://github.com/egc112/OpenWRT-egc-add-on/tree/main/notes

Introduction

These are my notes for setting up a WireGuard as a Client.
In essence WireGuard is a peer -to-peer protocol but because of differences in setup we still make a distinction between setting it up as a Client or as a Server, but a WireGuard interface can be setup to function as a Client and Server at the same time.

This guide was made on a NetGear R7800 running OpenWRT 24.10.0 and with OpenWRT2020 theme

My notes are using the easy way with a simple setup using LuCi although the corresponding configs are also shown. This simple setup is done by importing a config file (.conf) with necessary settings (see: config file).
Importing a config file is possible if you installed the *wg-installer-client* package (see Install WireGuard).
But just adding the settings manually will also do the trick.

Index

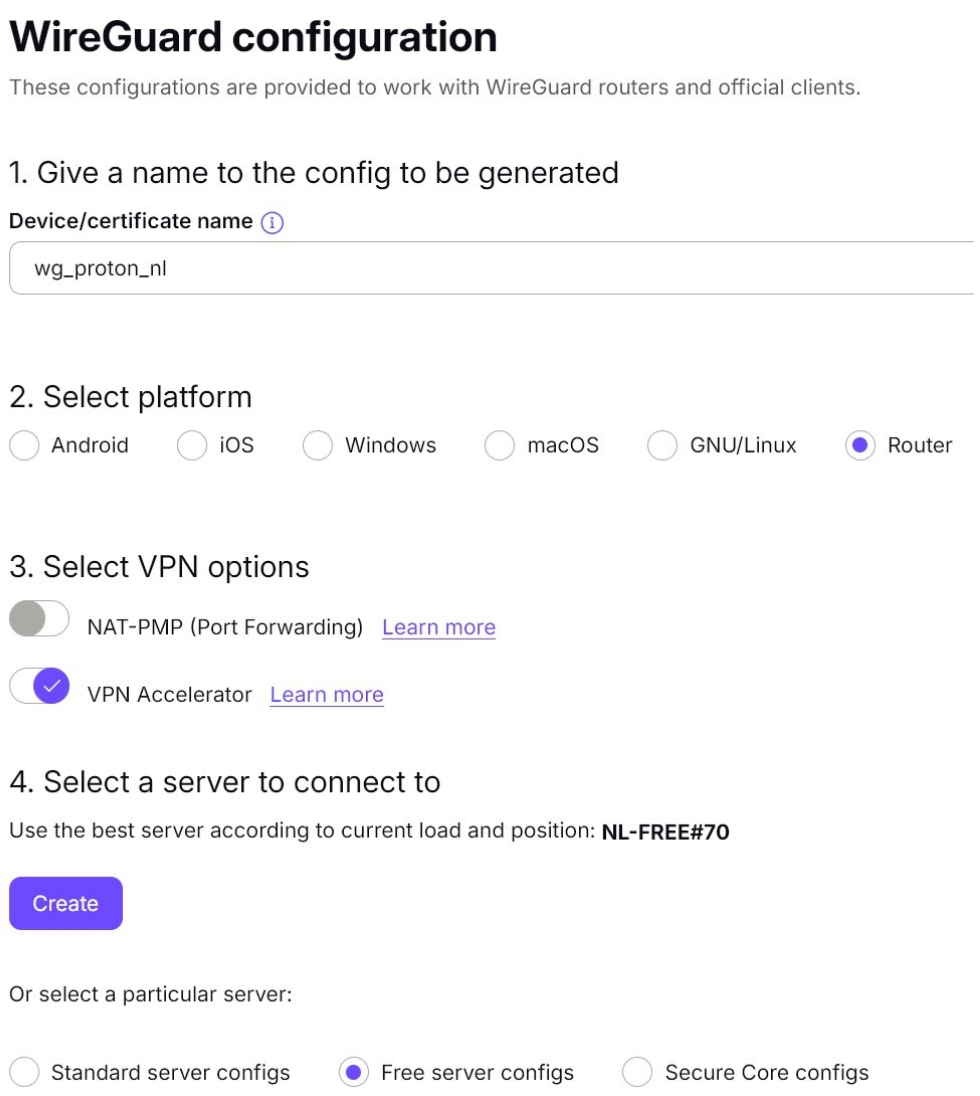
Install WireGuard

LuCi > System > Software: click *Update Lists*
Install: *luci-proto-wireguard*, *wireguard-tools* and *wg-installer-client.*

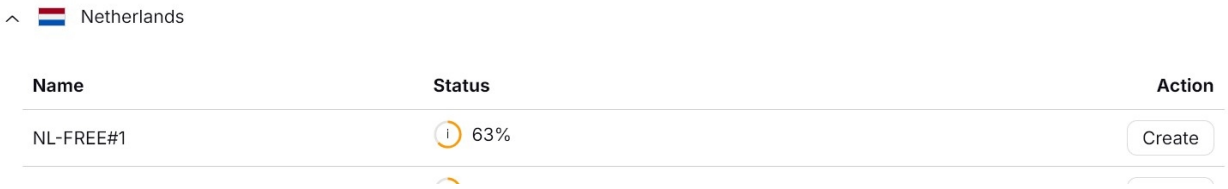
Download configuration

Download a WireGuard configuration file from your provider or WireGuard Server.
In this example we are going to download a WireGuard configuration file from Proton which is free but it will expire after a week or so:
Create an account on https://protonvpn.com/
Login
Go to Downloads and scroll to the bottom for the WireGuard configuration.

Give a name to your config and choose router for your Platform :



Scroll down to the server you want to connect to and Choose Create:



Download the config file to your computer, the config file (wg_proton_nl-NL-FREE-1.conf) looks like this:

```
[Interface]
# Key for wg_proton_nl
# Bouncing = 3
# NAT-PMP (Port Forwarding) = off
# VPN Accelerator = on
PrivateKey = UJmovcwC7KQ/vfgnradTHoHD30WJ6SonkvXYg23ex0A=
Address = 10.2.0.2/32
DNS = 10.2.0.1
[Peer]
# NL-FREE#1
PublicKey = vH2i8RY1qc66XfqwrixBpvH4K9GYJatkugJj0GHgoUQ=
AllowedIPs = 0.0.0.0/0
Endpoint = 217.23.3.76:51820
```

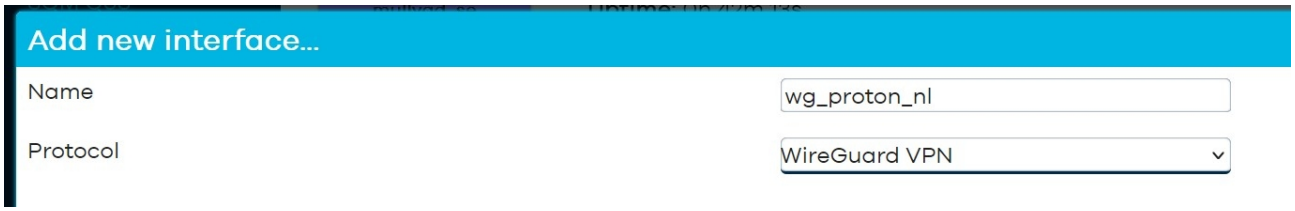
Add the `PersistentKeepAlive` so that the connection stays open:
PersistentKeepalive = 25 and if you use IPv6 add `::0/0` to allowed IPs:
AllowedIPs = 0.0.0.0/0, ::/0

The result:

```
[Interface]
# Key for wg_proton_nl
# Bouncing = 3
# NAT-PMP (Port Forwarding) = off
# VPN Accelerator = on
PrivateKey = UJmovcwC7KQ/vfgnradTHoHD30WJ6SonkvXYg23ex0A=
Address = 10.2.0.2/32
DNS = 10.2.0.1
[Peer]
# NL-FREE#1
PublicKey = vH2i8RY1qc66XfqwrixBpvH4K9GYJatkugJj0GHgoUQ=
AllowedIPs = 0.0.0.0/0, ::0/0
Endpoint = 217.23.3.76:51820
PersistentKeepalive = 25
```

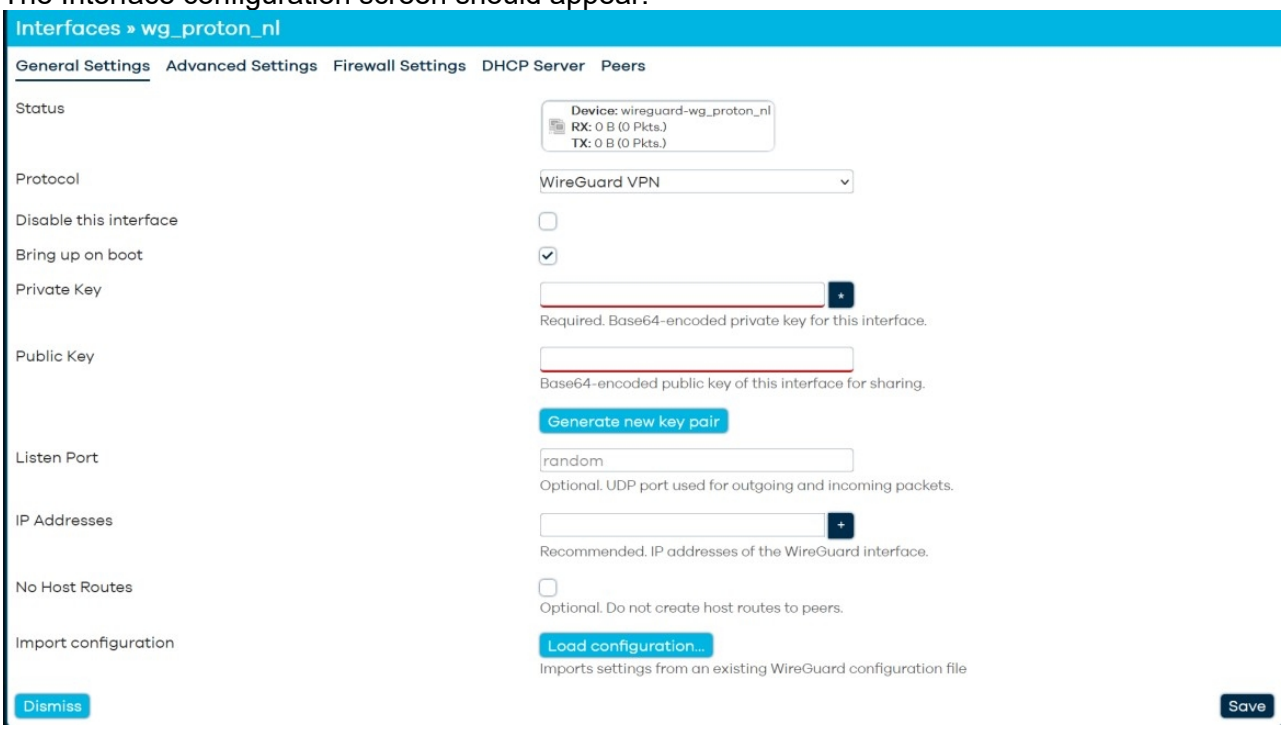
Create WireGuard interface

Network > Interfaces on the bottom **click**: *Add New interface*



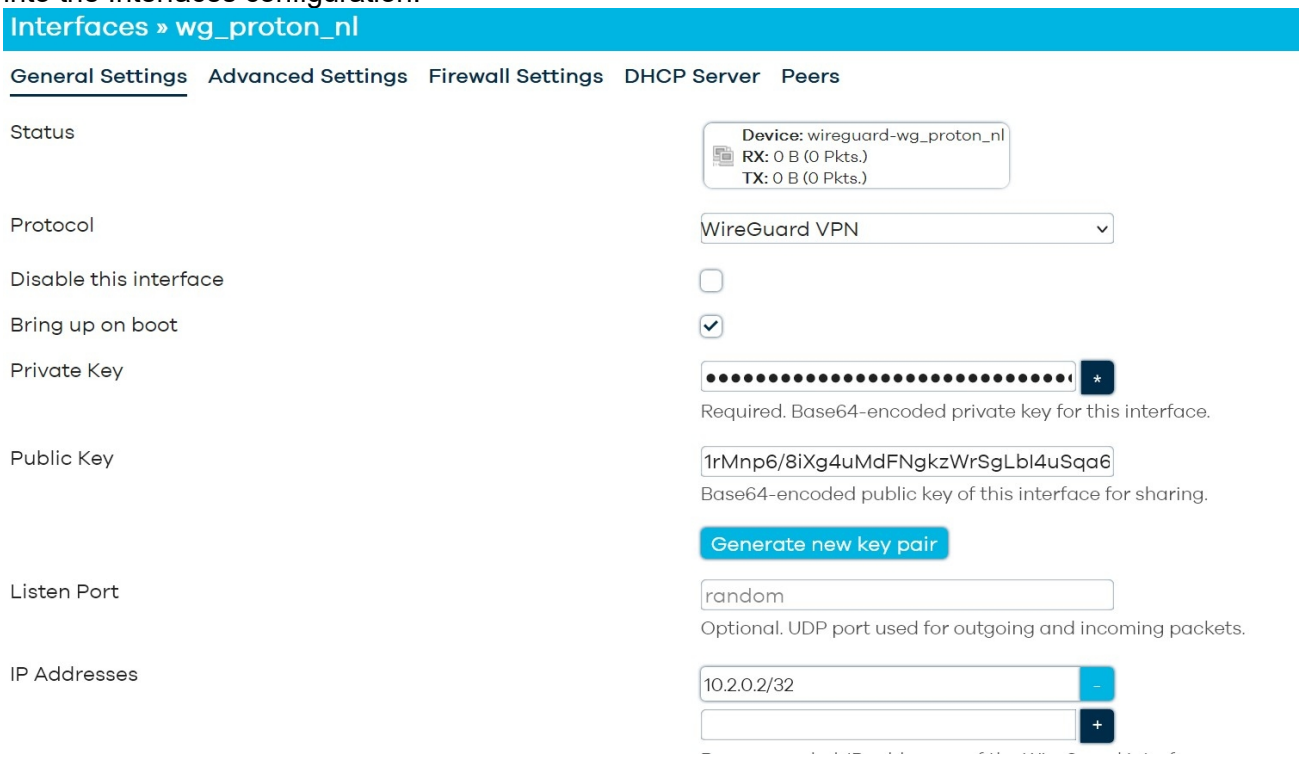
Name: give a descriptive name, hyphens are not allowed and the name has to be less than 15 characters!
Protocol: *WireGuard VPN*

Click: *Create interface*
The Interface configuration screen should appear:



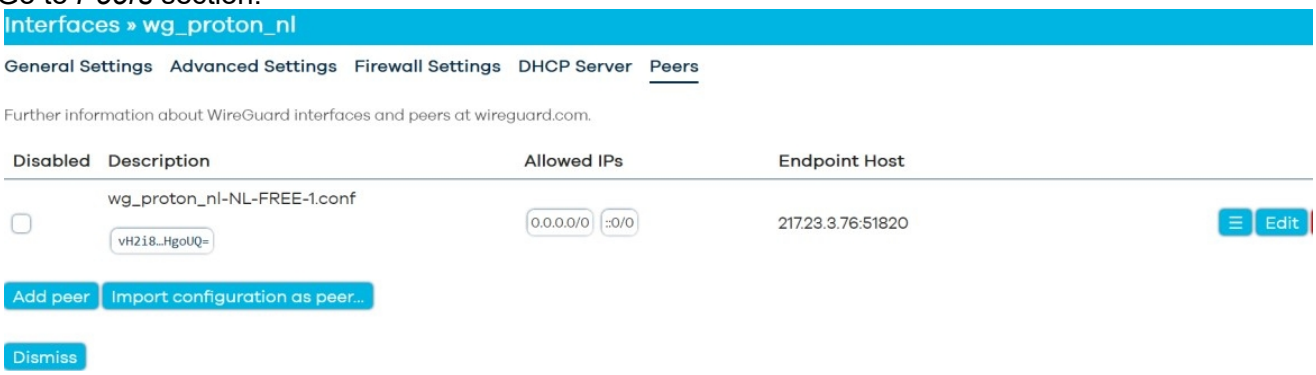
As the *wg-installer-client* is installed we can import our configuration file by clicking the button *Load configuration*

Click: *Load configuration*
Drop the configuration file from the file manager into this box and automagically the settings should appear into the Interfaces configuration:



Create WireGuard Peers section

Network > Interfaces > wg_proton_nl : **click** *edit*
Go to *Peers* section:



Click: *Edit* and the Peers section will open:

Interfaces » wg_proton_nl » Edit peer

Disabled	☐ Enable / Disable peer. Restart wireguard interface to apply chang
Description	wg_proton_nl-NL-FREE-1.conf Optional. Description of peer.
Public Key	vH2i8RY1qc66XfqwrixBpvH4K9GYJatkug Required. Public key of the WireGuard peer.
Private Key	* Optional. Private key of the WireGuard peer. The key is not require allows generating a peer configuration or QR code if available. It has been exported. Generate new key pair
Preshared Key	* Optional. Base64-encoded preshared key. Adds in an additional l post-quantum resistance. Generate preshared key
Allowed IPs	0.0.0.0/0 - ::0/0 - + Optional. IP addresses and prefixes that this peer is allowed to us tunnel IP addresses and the networks the peer routes through the
Route Allowed IPs	☐ Optional. Create routes for Allowed IPs for this peer.
Endpoint Host	217.23.3.76 Optional. Host of peer. Names are resolved prior to bringing up th
Endpoint Port	51820 Optional. Port of peer.
Persistent Keep Alive	25 Optional. Seconds between keep alive messages. Default is 0 (dis is behind a NAT is 25.

Now the most important part which is often overlooked:
Route Allowed IPs: *Enable (tick)*

Route Allowed IPs	☑ Optional. Create routes for Allowed IPs for this peer

Click: *Save*
In the next window **Click:** *Save* again
In the Interface window click *Save & Apply*

/etc/config/network:

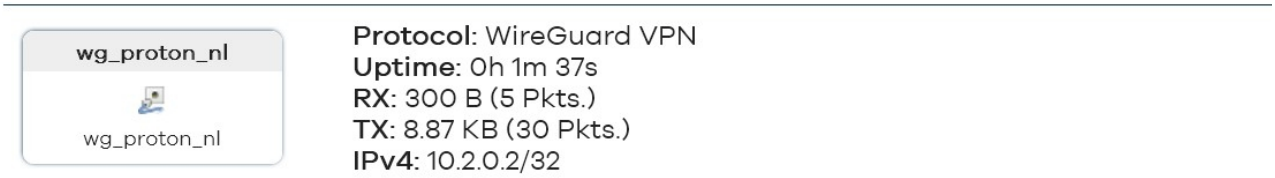
```
config interface 'wg_proton_nl'
        option proto 'wireguard'
        option private_key 'UJmovcwC7KQ/vfgnrasdfggdfgdfgdgddsgfdc='
        list dns '10.2.0.1'
        list addresses '10.2.0.2/24'

config wireguard_wg_proton_nl
        option description 'wg_proton_nl-NL-FREE-1.conf'
        option public_key 'vH2i8RY1qc66XfqwrixBpvH4K9dsfge4egdfgdfger='
        option endpoint_host '217.23.3.76'
        option endpoint_port '51820'
        list allowed_ips '0.0.0.0/0'
        list allowed_ips '::0/1'
```

```
list allowed_ips '8000::/0'
option route_allowed_ips '1'
option persistent_keepalive '25'
```

Note for IPv6 either use`::0/1` and `8000::/1` as Allowed IPs instead of ::0/0 to create a default route, or disable Source routing (Interface wan6 > `option sourcefilter '0'`) and set appropriate metrics on WG interface and higher metrics on default route in wan and wan6

After a few moments the interface appears and should be up and traffic should flow, both Tx and RX indicating the setup is correct:



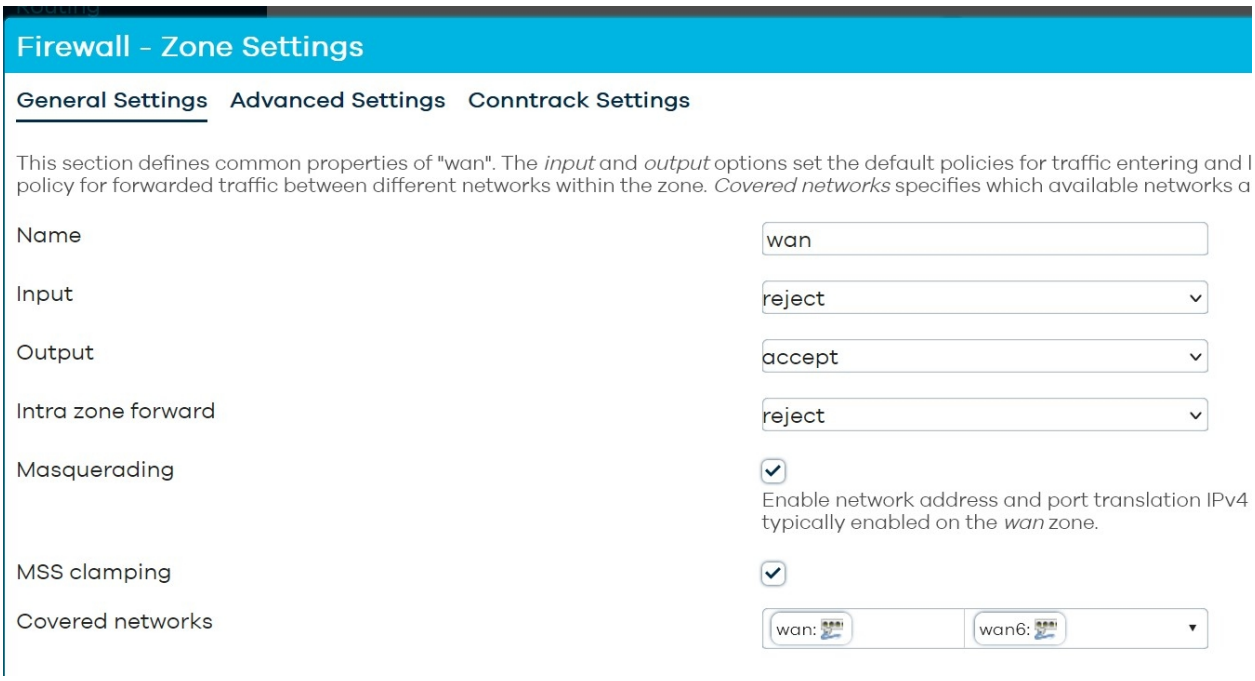
However this is depending on your default firewall setting with OUTPUT Accept, if not there will not be traffic yet.
Next up Firewall

Firewall

Easy method

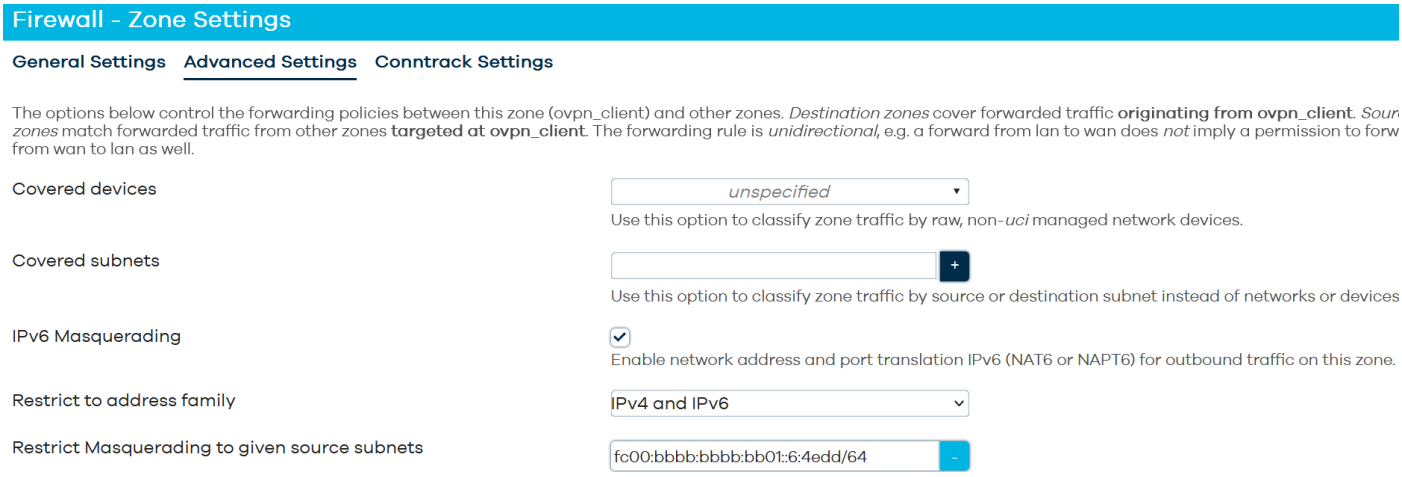
Easiest method is to just add the wg_proton_nl interface to the WAN zone

Network > Firewall > WAN zone > **Click**: *edit:*



Covered Networks: add *wg_proton_nl*

For IPv6 enable IPv6 Masquerading on the WireGuard firewall zone:
Advanced settings > Enable IPv6 Masquerading
but restrict this to the IPv6 subnet of the WireGuard interface




```
/etc/config/firewall:
config zone
        option name 'wan'
        option input 'REJECT'
        option output 'ACCEPT'
        option forward 'REJECT'
        option masq '1'
        option mtu_fix '1'
        option masq6 '1'
        list masq_src 'fc00:bbbb:bbbb:bb01::6:4edd/64'
        list network 'wan'
        list network 'wan6'
        list network 'wg_proton_nl'
```

Click: *Save* and click *Save & Apply*

This should give you a working WireGuard Client

Check from the routers console with *curl ipinfo.io* and/or from your LAN clients with *ipleak.net*

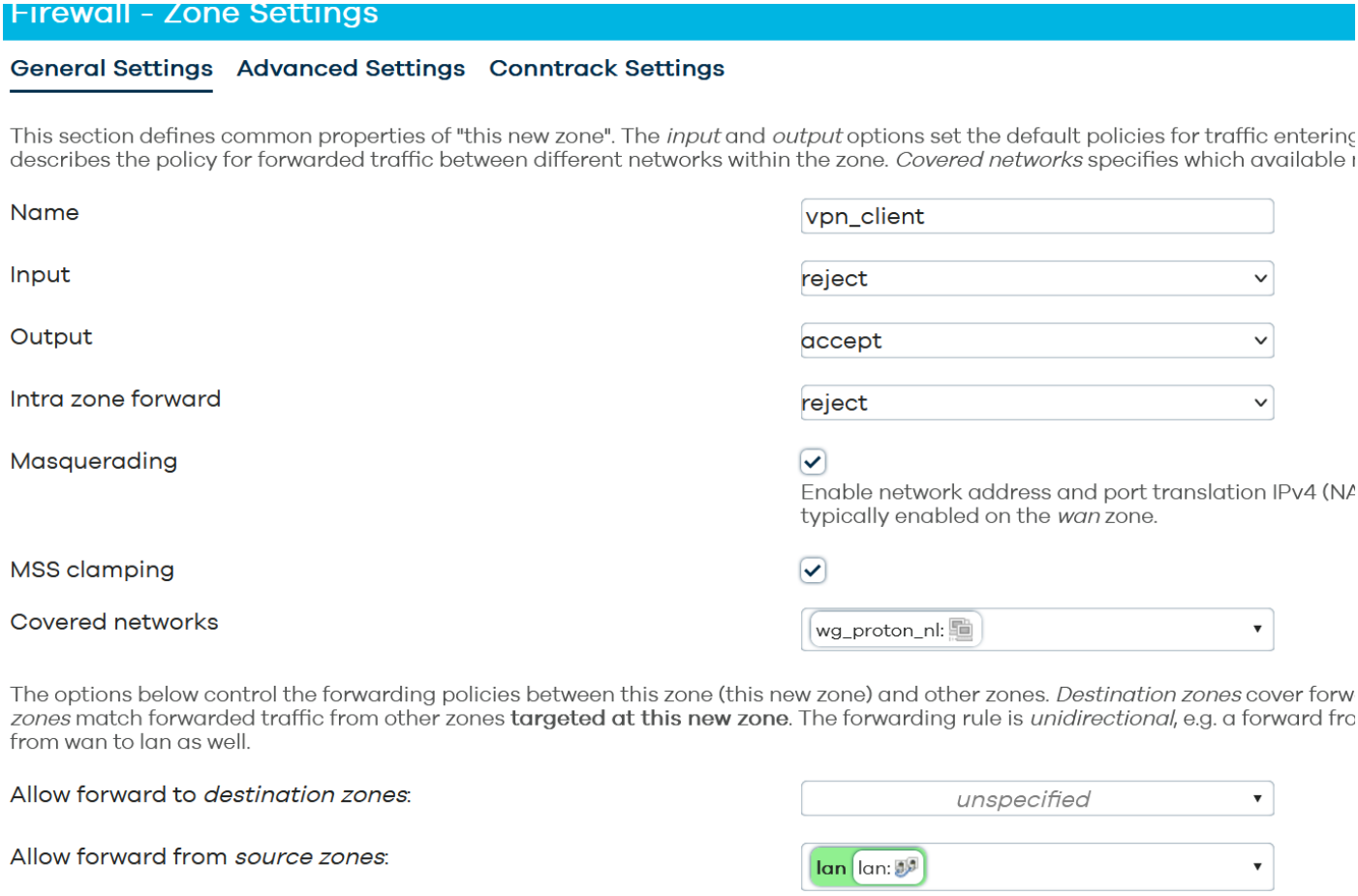
Alternative Method

The Alternative method is to make a separate firewall zone for the VPN interface.
This can be useful if you want to make a killswitch (prevent traffic going out of the wan) or setup a Wireguard client on a Bridged AP.

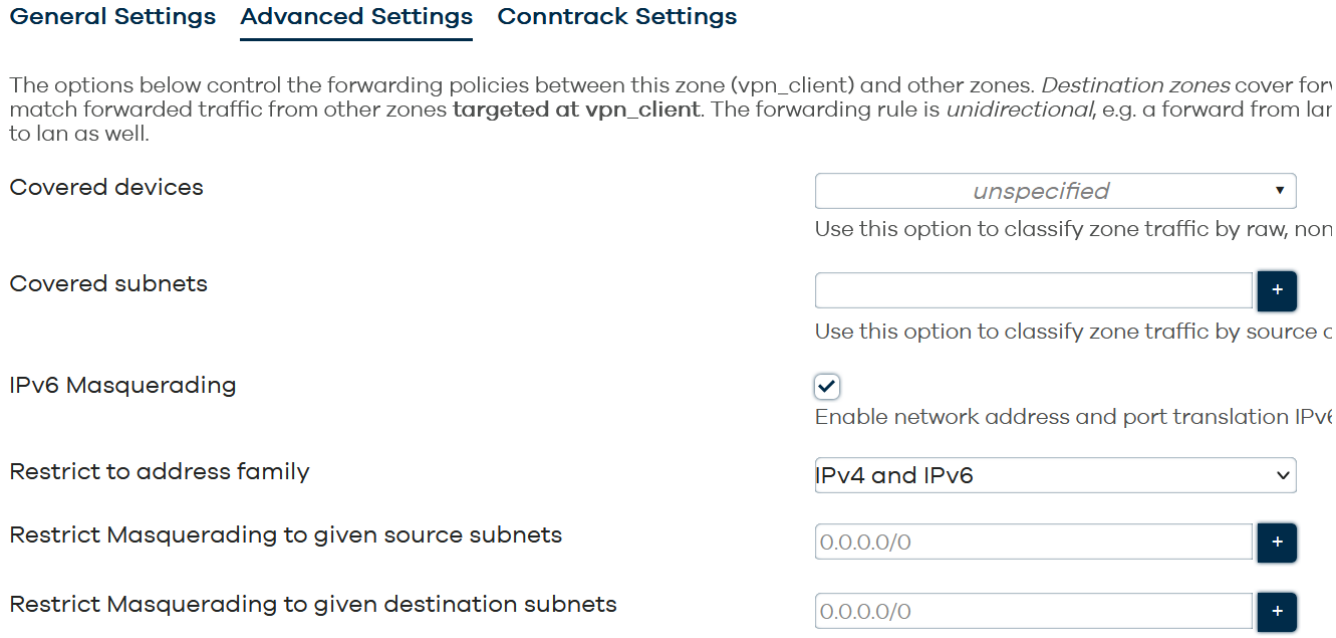
Note that a killswitch is not really necessary as the wireGuard interface stays up even if ther is no connection but it will add aen extra layer of security and guards against misconfiguration

Network > Firewall > **Click**: *Add*:
Name: *vpn_client*
Input: *reject*
Output: *accept*
Intra zone forward: *reject*
Masquerading: *enabled*
MSS clamping: *enabled*

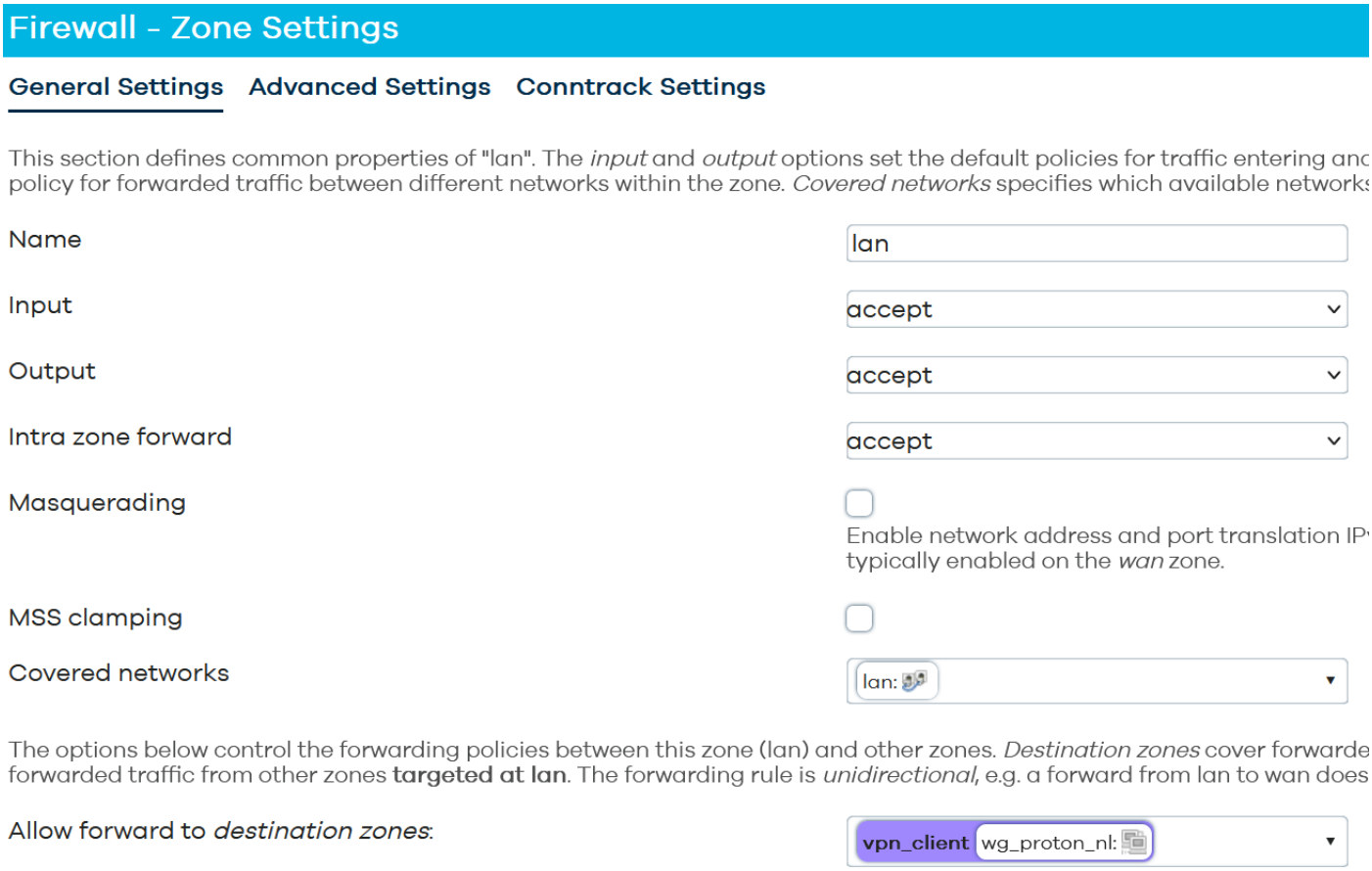
Allow forward from source zone: *lan*



If your VPN provider also supports IPv6 with ULA addresses then on Advanced tab:
IPv6 Masquerading: *enable*



To prevent traffic going out of the wan (the Killswitch) **Edit** the *lan* firewall zone and disable forwarding to *wan* and only allow forwarding to the *vpn_client* zone



/etc/config/firewall:

```
config zone
        option name 'vpn_client'
        option input 'REJECT'
        option output 'ACCEPT'
        option forward 'REJECT'
        option masq '1'
        option mtu_fix '1'
        list network 'wg_proton_nl'
        option masq6 '1'    # only for IPv6

config forwarding
        option src 'lan'
        option dest 'vpn_client'
```

DNS Leak

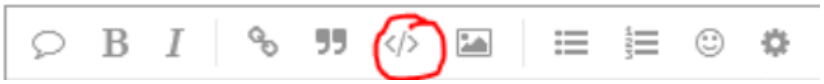
On a typical phone (Android, iOS) or Windows the DNS is just set on the WireGuard interface and the DNS set is used after the tunnel is up.
On the OpenWRT router things are much more complicated (in contrast to other third party firmwares which handles this much better)
For some background reading: https://github.com/egc112/OpenWRT-egc-add-on/tree/main/stop-dns-leak

Be very careful with using the DNS server from your VPN provider as sole DNS server if that DNS server is not publicly available as you might end up in a catch 22 situation because the router must have the correct time (more or less) before it can connect and to get the correct time it needs DNS resolving which is not available.
So in that case instead of a domain for time server use IP addresses (System > System > Time Synchronization)
Of course if you stop the tunnel you do not have DNS resolution in that case you need a scripting solution to use the VPN DNS server after the tunnel is up.

Asking for Help

You can ask for help at the OpenWRT forum.

If you do, it helps if we can have a look at your configs, so please connect to your OpenWRT device using ssh and copy the output of the following commands and post it on the forum using the "Preformatted text </> " button



Remember to redact keys, passwords, MAC addresses and any public IP addresses you may have:

- ubus call system board
- cat /etc/config/network
- cat /etc/config/wireless
- cat /etc/config/firewall
- wg show

References

https://openwrt.org/docs/guide-user/services/vpn/wireguard/start
https://openwrt.org/docs/guide-user/services/vpn/wireguard/basics
https://openwrt.org/docs/guide-user/services/vpn/wireguard/client

https://protonvpn.com/support/openwrt-wireguard