

# 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, screenshots are made with OpenWRT2020 theme which is not much different from the default theme.

My notes are using the easy way with a simple setup using LuCi although the corresponding config files are also listed.

This simple setup is done by importing a config file (.conf) from your VPN provider 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

OpenWRT WireGuard Client Setup guide using Luci.....	1
Introduction.....	1
Install WireGuard.....	1
Download configuration.....	1
Create WireGuard interface.....	4
MTU (Maximum Transmission Unit).....	5
Create WireGuard Peers section.....	5
Firewall.....	7
Easy method.....	7
Alternative Method.....	8
DNS Leak.....	11
WireGuard Client on a BridgedAP.....	11
Asking for Help.....	12
References.....	12
Miscellaneous.....	12
Setup IPv6 on a bridgedAP.....	12

## 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 :

## WireGuard configuration

These configurations are provided to work with WireGuard routers and official clients.

### 1. Give a name to the config to be generated

Device/certificate name ⓘ

wg\_proton\_nl

### 2. Select platform

☐ Android ☐ iOS ☐ Windows ☐ macOS ☐ GNU/Linux ☒ Router

### 3. Select VPN options

☐ NAT-PMP (Port Forwarding) [Learn more](#)

☒ VPN Accelerator [Learn more](#)

### 4. Select a server to connect to


Use the best server according to current load and position: **NL-FREE#70**


Create

Or select a particular server:

☐ Standard server configs ☒ Free server configs ☐ Secure Core configs

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

^  Netherlands

Name	Status	Action
NL-FREE#1	 63%	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]**

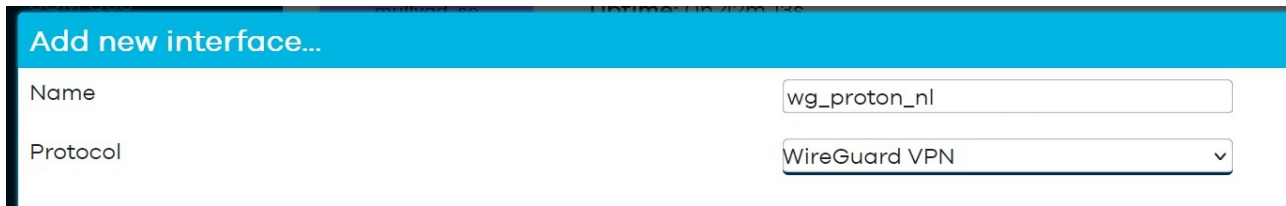
```
# Key for wg_proton_nl
# Bouncing = 3
# NAT-PMP (Port Forwarding) = off
# VPN Accelerator = on
PrivateKey = UJmovcwC7KQ/vfgnradTHoHD30WJ6SonkvXYg23ex0A=
Address = 10.2.0.2/32
DNS = 10.2.0.1
```

**[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*



Add new interface...

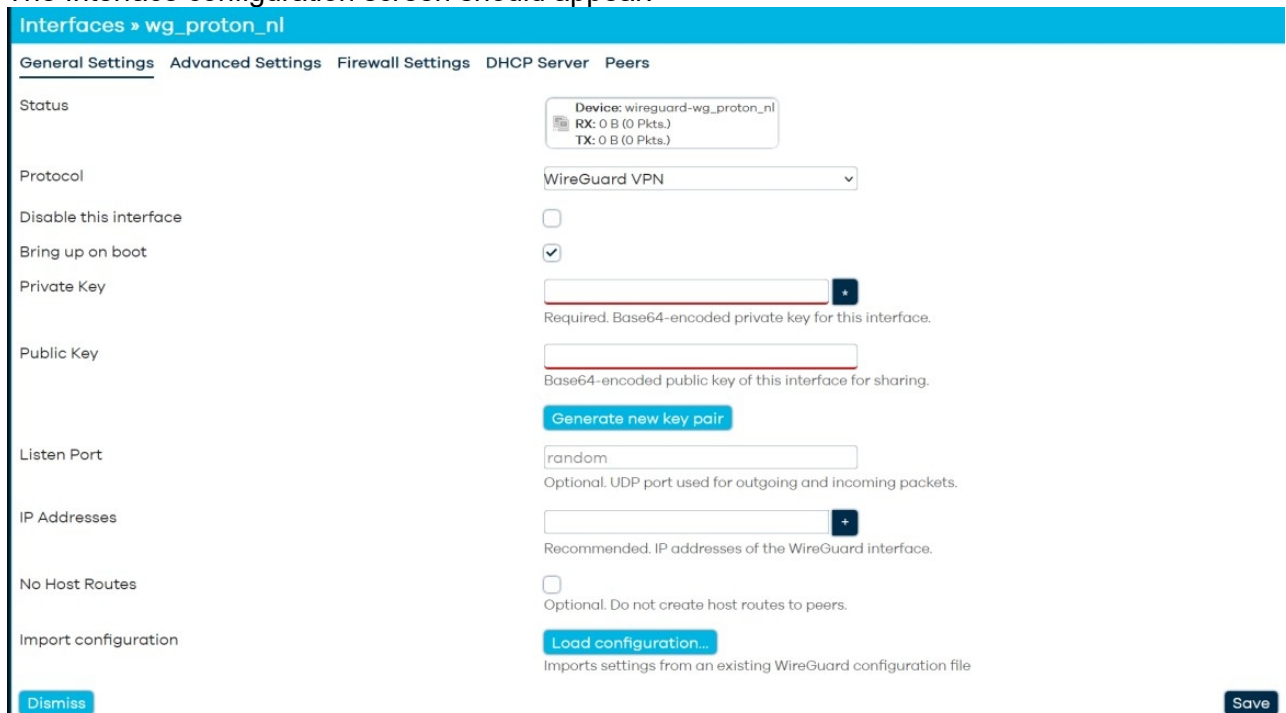
Name: wg\_proton\_nl

Protocol: WireGuard VPN

**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:



Interfaces > wg\_proton\_nl

General Settings Advanced Settings Firewall Settings DHCP Server Peers

Status: Device: wireguard-wg\_proton\_nl  
RX: 0 B (0 Pkts.)  
TX: 0 B (0 Pkts.)

Protocol: WireGuard VPN

Disable this interface: ☐

Bring up on boot: ☒

Private Key:  [Generate new key pair](#)  
Required. Base64-encoded private key for this interface.

Public Key:  [Generate new key pair](#)  
Base64-encoded public key of this interface for sharing.

Listen Port: random  
Optional. UDP port used for outgoing and incoming packets.

IP Addresses:  [+](#)  
Recommended. IP addresses of the WireGuard interface.

No Host Routes: ☐  
Optional. Do not create host routes to peers.

Import configuration: [Load configuration...](#)  
Imports settings from an existing WireGuard configuration file

[Dismiss](#) [Save](#)

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 automatically the settings should appear into the Interfaces configuration:

Interfaces » wg\_proton\_nl

General Settings Advanced Settings Firewall Settings DHCP Server Peers

Status

Device: wireguard-wg\_proton\_nl  
RX: 0 B (0 Pkts.)  
TX: 0 B (0 Pkts.)

Protocol

WireGuard VPN

Disable this interface

☐

Bring up on boot

☒

Private Key

.....\*

Required. Base64-encoded private key for this interface.

Public Key

1rMnp6/8iXg4uMdFNgzWrSgLb14uSqa6

Base64-encoded public key of this interface for sharing.

Generate new key pair

Listen Port

random

Optional. UDP port used for outgoing and incoming packets.

IP Addresses

10.2.0.2/32

-

+

### MTU (Maximum Transmission Unit)

Interfaces » wg\_proton\_nl

General Settings Advanced Settings Firewall Settings DHCP Server Peers

Force link

☐  
Set interface properties regardless of the link carrier (If set handlers).

MTU

1420

Optional. Maximum Transmission Unit of tunnel interface.

MTU setting on the Advanced Settings tab can usually be left at its default setting (1420 or 1412 for PPPoE).

On occasion if you have **slow or hanging connections** especially when streaming you have to lower the MTU, start lowering to 1280, but sometimes you have to go even lower.

Note that you also have to enable MSS clamping on the firewall zone the WireGuard interface is added to

## Create WireGuard Peers section

Network > Interfaces > wg\_proton\_nl : **click edit**

Go to **Peers** section:

Interfaces » wg\_proton\_nl

General Settings Advanced Settings Firewall Settings DHCP Server Peers

Further information about WireGuard interfaces and peers at wireguard.com.

Disabled	Description	Allowed IPs	Endpoint Host	
<input type="checkbox"/>	wg_proton_nl-NL-FREE-1.conf vH2i8_HgoUQ=	0.0.0.0/0 :0/0	217.23.3.76:51820	<div>Edit</div>

Add peer

Import configuration as peer...

Dismiss

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

**Interfaces » wg\_proton\_nl » Edit peer**

Disabled	<input type="checkbox"/>	Enable / Disable peer. Restart wireguard interface to apply changes.
Description	<input type="text" value="wg_proton_nl-NL-FREE-1.conf"/>	Optional. Description of peer.
Public Key	<input type="text" value="vH2i8RY1qc66XfqwrixBpvH4K9GYJatkug"/>	Required. Public key of the WireGuard peer.
Private Key	<input type="text"/> *	Optional. Private key of the WireGuard peer. The key is not required; allows generating a peer configuration or QR code if available. It has been exported.
	<button>Generate new key pair</button>	
Preshared Key	<input type="text"/> *	Optional. Base64-encoded preshared key. Adds in an additional layer of post-quantum resistance.
	<button>Generate preshared key</button>	
Allowed IPs	<div><input type="text" value="0.0.0.0/0"/> -</div> <div><input "::0="" 0"="" type="text" value=""/> -</div> <div><input type="text"/> +</div>	Optional. IP addresses and prefixes that this peer is allowed to use tunnel IP addresses and the networks the peer routes through the
Route Allowed IPs	<input type="checkbox"/>	Optional. Create routes for Allowed IPs for this peer.
Endpoint Host	<input type="text" value="217.23.3.76"/>	Optional. Host of peer. Names are resolved prior to bringing up the
Endpoint Port	<input type="text" value="51820"/>	Optional. Port of peer.
Persistent Keep Alive	<input type="text" value="25"/>	Optional. Seconds between keep alive messages. Default is 0 (disabled). Behind a NAT is 25.

**Now the most important part which is often overlooked:**

**Route Allowed IPs: *Enable (tick)***

Route Allowed IPs	<input checked="" type="checkbox"/>	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/vfgnrasdffgdfgdfgdgddsgfddc='
    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' # leave in place for PBR
```

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

**Note for IPv6** add `::/1` and `8000::/1` as Allowed IPs 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:

wg\_proton\_nl



wg\_proton\_nl

Protocol: WireGuard VPN

Uptime: 0h 1m 37s

RX: 300 B (5 Pkts.)

TX: 8.87 KB (30 Pkts.)

IPv4: 10.2.0.2/32

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:**

Firewall - Zone Settings

General Settings

Advanced Settings

Conntrack Settings

This section defines common properties of "wan". The *input* and *output* options set the default policies for traffic entering and policy for forwarded traffic between different networks within the zone. *Covered networks* specifies which available networks a

Name

wan

Input

reject

Output

accept

Intra zone forward

reject

Masquerading

☒

Enable network address and port translation IPv4 typically enabled on the wan zone.

MSS clamping

☒

Covered networks

wan:

wan6:

wg\_proton\_nl:

**Covered Networks:** add wg\_proton\_nl

Covered networks

wan:

wan6:

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

## Firewall - Zone Settings

General Settings   **Advanced Settings**   Conntrack Settings

The options below control the forwarding policies between this zone (ovpn\_client) and other zones. *Destination zones* cover forwarded traffic **originating from ovpn\_client**. *Source zones* match forwarded traffic from other zones **targeted at ovpn\_client**. The forwarding rule is *unidirectional*, e.g. a forward from lan to wan does *not* imply a permission to forward from wan to lan as well.

Covered devices	<input type="text" value="unspecified"/>	Use this option to classify zone traffic by raw, non-uci managed network devices.
Covered subnets	<input type="text"/> +	Use this option to classify zone traffic by source or destination subnet instead of networks or devices
IPv6 Masquerading	<input checked="" type="checkbox"/>	Enable network address and port translation IPv6 (NAT6 or NAPT6) for outbound traffic on this zone.
Restrict to address family	<input type="text" value="IPv4 and IPv6"/>	
Restrict Masquerading to given source subnets	<input type="text" value="fc00:bbbb:bbbb:bb01::6:4edd/64"/>	

```
/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_n1'
```

**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 there is no connection but it will add an extra layer of security and guards against mis-configuration.

Furthermore a killswitch is not compatible with [Policy Based Routing \(PBR\)](#).

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*



## Firewall - Zone Settings

### General Settings Advanced Settings Contrack Settings

This section defines common properties of "this new zone". The *input* and *output* options set the default policies for traffic entering and exiting the zone. The *forward* option describes the policy for forwarded traffic between different networks within the zone. *Covered networks* specifies which available networks are covered by this zone.

Name	<input type="text" value="vpn_client"/>
Input	<input type="text" value="reject"/>
Output	<input type="text" value="accept"/>
Intra zone forward	<input type="text" value="reject"/>
Masquerading	<input checked="" type="checkbox"/> Enable network address and port translation IPv4 (NAT) typically enabled on the <i>wan</i> zone.
MSS clamping	<input checked="" type="checkbox"/>
Covered networks	<input type="text" value="wg_proton_nl: [icon]"/>

The options below control the forwarding policies between this zone (this new zone) and other zones. *Destination zones* cover forwarded traffic from other zones **targeted at this new zone**. The forwarding rule is *unidirectional*, e.g. a forward from *wan* to *lan* as well.

Allow forward to <i>destination zones</i> :	<input type="text" value="unspecified"/>
Allow forward from <i>source zones</i> :	<input type="text" value="lan lan: [icon]"/>

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

### General Settings Advanced Settings Contrack Settings

The options below control the forwarding policies between this zone (*vpn\_client*) and other zones. *Destination zones* cover forwarded traffic from other zones **targeted at vpn\_client**. The forwarding rule is *unidirectional*, e.g. a forward from *lan* to *lan* as well.

Covered devices	<input type="text" value="unspecified"/> Use this option to classify zone traffic by raw, non
Covered subnets	<input type="text" value=""/> Use this option to classify zone traffic by source c
IPv6 Masquerading	<input checked="" type="checkbox"/> Enable network address and port translation IPv6
Restrict to address family	<input type="text" value="IPv4 and IPv6"/>
Restrict Masquerading to given source subnets	<input type="text" value="0.0.0.0/0"/>
Restrict Masquerading to given destination subnets	<input type="text" value="0.0.0.0/0"/>

```
/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_n1'
option masq6 '1'
```

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

## Firewall - Zone Settings

### General Settings Advanced Settings Conntrack Settings

This section defines common properties of "lan". The *input* and *output* options set the default policies for traffic entering and policy for forwarded traffic between different networks within the zone. *Covered networks* specifies which available networks

Name	lan
Input	accept
Output	accept
Intra zone forward	accept
Masquerading	<input type="checkbox"/> Enable network address and port translation IP typically enabled on the <i>wan</i> zone.
MSS clamping	<input type="checkbox"/>
Covered networks	lan:

The options below control the forwarding policies between this zone (lan) and other zones. *Destination zones* cover forwarded traffic from other zones **targeted at lan**. The forwarding rule is *unidirectional*, e.g. a forward from lan to wan does

Allow forward to *destination zones*:

vpn\_client wg\_proton\_n1:

/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_n1'
    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.

## WireGuard Client on a BridgedAP

If you want setup a WireGuard Client on a BridgedAP, then be aware that normal traffic from your clients just bypasses your BridgedAP, so will not use the WireGuard tunnel unless you point the gateway of your clients to the BridgedAP (by using DNSMasq tagging with option 3 or iptables redirect) or setup a Guest Wifi on the BridgedAP.

In case of using a Guest wifi all clients using your Guest wifi will automatically use the WireGuard tunnel, so this is the more easier option.

First double check that you have setup your BridgedAP correctly see:

<https://openwrt.org/docs/guide-user/network/wifi/wifiextenders/bridgedap>

For a Guest Wifi see:

[https://openwrt.org/docs/guide-user/network/wifi/guestwifi/guestwifi\\_dumbap](https://openwrt.org/docs/guide-user/network/wifi/guestwifi/guestwifi_dumbap)

Setup the Guest wifi and **check that it is working without WireGuard!**

Note: do not forget to Enable Masquerading on the LAN zone

Setup a WireGuard client the regular way as described, but for firewall settings I recommend to use a separate zone for the WireGuard Interface ([Alternative Method](#)) as you need to enable MSS Clamping and then make a Forward rule to Forward from *guest* zone to *vpn\_client* zone.

If you then remove the forwarding from guest zone to lan zone you will have an effective killswitch.

For IPv6 make sure your lan has an IPv6 with prefix delegated, the Guest interface will then gets its own IPv6 address from Lan

## 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>

## Miscellaneous

### Setup IPv6 on a bridgedAP

`/etc/config/network:`

```
config interface 'lan6'
    option ifname '@lan'
    option proto 'dhcpv6'
    #option reqprefix 'no'
    option reqprefix '62'    #for ipv6 guest interface
    #option sourcefilter '0' # disable source routing for WG server routing of IPv6
```