OpenWRT Policy Based Routing (PBR)

Latest iteration can be found at: https://github.com/egc112/OpenWRT-egc-add-on/tree/main/notes
Version 4

Index

Introduction

When using a VPN usually all traffic is routed via the VPN. But sometimes you want to make an exempt for some traffic to use the WAN instead of the VPN.
This is done with the help of Policy Based Routing (PBR).
Policy Based Routing (PBR) works by creating routing tables and ip rules that specify which routing table to use when certain criteria are met.
Criteria for routing decision can be source and destination ip address, port, interface, domain name, fwmark etc. see: ip rule man page

A good starting point for using PBR in OpenWRT is the OpenWRT PBR Wiki.

The OpenWRT PBR Wiki covers three main items which can be used for PBR.
This document will discuss the manual method using netifd.
The PBR app which is the swiss army knife of PBR including a GUI is covered in the excellent PBR app install and user guide. For simple needs e.g. a specific lan client, interface or port, the manual method using netifd will do but if you have more elaborate needs installing the full PBR app is the way to go.
For MWAN3 see the MWAN3 wiki.

Testbed
Dynalink DL-WRX36 main/snapshot build form 25-jul-2025
WireGuard VPN client to mullvad setup according to the WireGuard Client Setup guide

This guide uses LuCi to set things up but the resulting **config files** are also listed. The screenshots are made with OpenWRT2020 theme but that is not much different from the default theme.

Manual method with netifd

Default route

First we cover the default routing.
A typical VPN setup using Wireguard or OpenVPN will route all traffic via the VPN.
If most of the traffic indeed uses the VPN and if you want to keep it that way then proceed to Creating Routing tables via the WAN.
But if you want most of the traffic to use the WAN then disable the default route via the VPN and make a routing table and ip rules to manually route the desired traffic via the VPN.

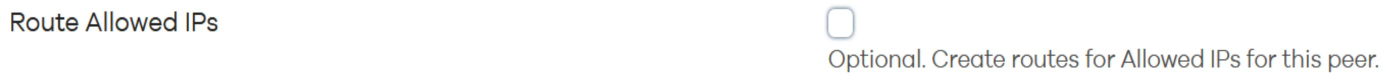
Manual Method removing Default route via the VPN

WireGuard disable default route via VPN:

There are three ways to disable the WireGuard VPN default route

1. Disable (untick) *Route Allowed IPs*

LuCi > Interfaces > WireGuard Interface > Peers > *Edit Peer*



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

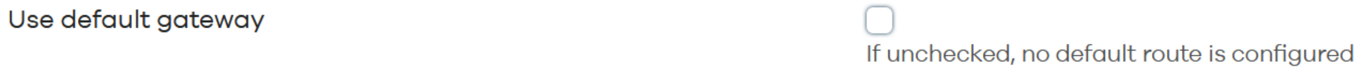
/etc/config/network:

```
config wireguard_wg_mullv_us
	option description 'mullvad-us-bos-wg-002.conf'
	option public_key 'LXXXXXX'
	option persistent_keepalive '25'
	option endpoint_host '43.225.189.162'
	option endpoint_port '51820'
	list allowed_ips '0.0.0.0/0'
	list allowed_ips '::0/0'
	list allowed_ips '8000::/1'
	list allowed_ips '::0/1'
	option route_allowed_ips '0'
```

Note Disabled is the default so you can also remove the `option route_allowed_ips`

2. Uncheck *Default Gateway*

LuCi > Interfaces > WireGuard interface > Advanced Settings:



Use default gateway □
If unchecked, no default route is configured

/etc/config/network:

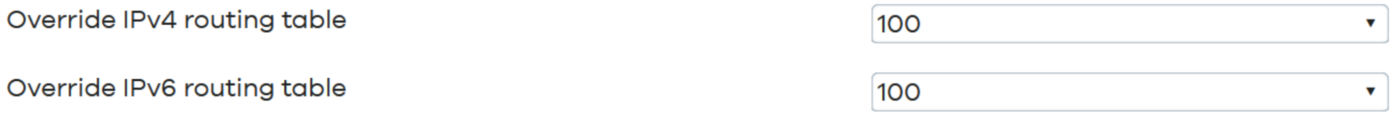
```
config interface 'wg_mullv_us'
	option proto 'wireguard'
	option private_key 'YGaBrXXXXXX'
	list addresses '10.68.89.7/32'
	list addresses 'fc00:bbbb:bbbb:bb01::5:5906/128'
	list dns '10.64.0.1'
	option defaultroute '0'
```

This method can be useful if you have multiple routes in the Allowed IPs which you want to use and only want to stop default routing but keep the other routes enabled.

3. Use *OptionTable*

This method will remove the default route via the VPN **and** make an alternate routing table with default route via the VPN.
Important is that you keep the default route via the interface intact as the OptionTable method will move the default route from the main routing table to an alternate routing table.
This only works if you have one VPN with a default route, if you have multiple VPN's then see above, use Disable (untick) Route Allowed IPs or Uncheck Default Gateway

To remove the default route via the VPN and make a new alternate routing table with table number 100:
LuCi > interfaces > WireGuard interface > Advanced options:



Override IPv4 routing table: 100
Override IPv6 routing table: 100

/etc/config/network:

```
config interface 'wg_mullv_us'
        option proto 'wireguard'
        option private_key 'YGaBrMXXXXXX'
        list addresses '10.68.89.7/32'
        list addresses 'fc00:bbbb:bbbb:bb01::5:5906/128'
        list dns '10.64.0.1'
        option ip4table '100'
        option ip6table '100'
```

For various reasons it is not feasible to use the option table method to make an alternate routing table via the **WAN,** if you want this then use the other methods mentioned.

OpenVPN disable default route via VPN:

Add in the OpenVPN config:
For IPv4 only:

pull-filter ignore "redirect-gateway"

For IPv6:

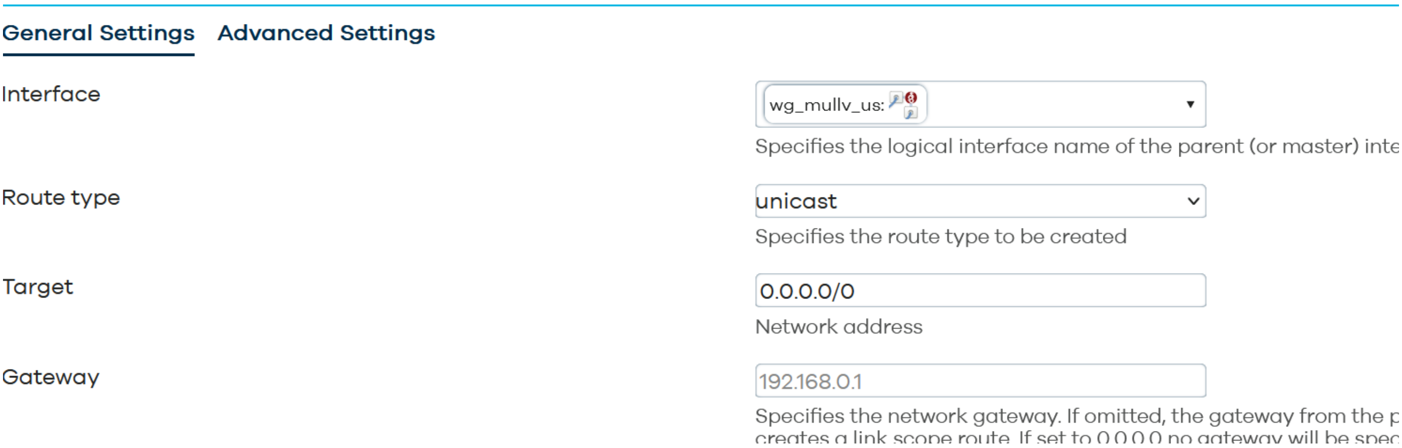
pull-filter ignore "redirect-gateway ipv6"
pull-filter ignore "route-ipv6 0000::/2"
pull-filter ignore "route-ipv6 4000::/2"
pull-filter ignore "route-ipv6 8000::/2"
pull-filter ignore "route-ipv6 C000::/2"
pull-filter ignore "route-ipv6 2000::/3"

Creating Routing tables via the VPN

If you use the OptionTable method then the routing table is automatically created, for the other methods you must create the routing tables as described below.

Create alternate VPN routing table with default route via the VPN:
Luci > Network > Routing > Static IPv4 Routes > *Add*:

You specify
Interface: *<your WireGuard interface>*
Target: *0.0.0.0/0*
On Advanced Settings the table number:
Table: *100* (this is an arbitrary number)
Table: *100*



Interface: wg_mullv_us:
Specifies the logical interface name of the parent (or master) inte

Route type: unicast
Specifies the route type to be created

Target: 0.0.0.0/0
Network address

Gateway: 192.168.0.1
Specifies the network gateway. If omitted, the gateway from the p
creates a link scope route. If set to 0.0.0.0 no gateway will be spec

Advanced Settings:

Table
100
Routing table into which to insert this rule.

You do the same for Static IPv6 routes but you use as Target: *::0/0*

/etc/config/network:
```
config route
	option interface 'wg_mullv_us'
	option target '0.0.0.0/0'
	option table '100'

config route6
	option interface 'wg_mullv_us'
	option target '::/0'
	option table '100'
```

For OpenVPN you have to add the OpenVPN interface in the Network section and then use that interface for making your routing table.
As device you use the device which is set in your OpenVPN config e.g. : *dev tun0* if only *dev tun* is set then OpenVPN uses the first available tunnel number so it will be *tun0*

Creating Routing tables via the WAN

This works exactly as making routing tables via the VPN the only difference is that you use as interface" *wan/wan6* and set the gateway. The gateway is the nexthop which can be get from the command line with
For IPv4: *ifstatus wan* | *grep nexthop*
For IPv6: i*fstatus wan6* | *grep nexthop*
Interface: *wan/wan6*
Gateway: <*next hop*>

/etc/config/network:
```
config route
	option interface 'wan'
	option target '0.0.0.0/0'
	option gateway '192.168.0.1'

config route6
	option interface 'wan6'
	option target '::0/0'
	option gateway 'fe80::bef5:81ff:fc4e:82a2'
	option table '100'
```

You can check routing with (from command line):
ip route show
ip route show table 100
ip -6 route show
ip -6 route show table 100

For the record I also added the use of *Option Table* to remove Default route of the VPN and making an alternate routing table see: Using Option Table to remove VPN default route

Creating ip rules

Now that we have an alternate routing table we are going to create rules which are used to create policies with which you specify which clients, ports, interfaces etc are going to use these alternate routing tables.

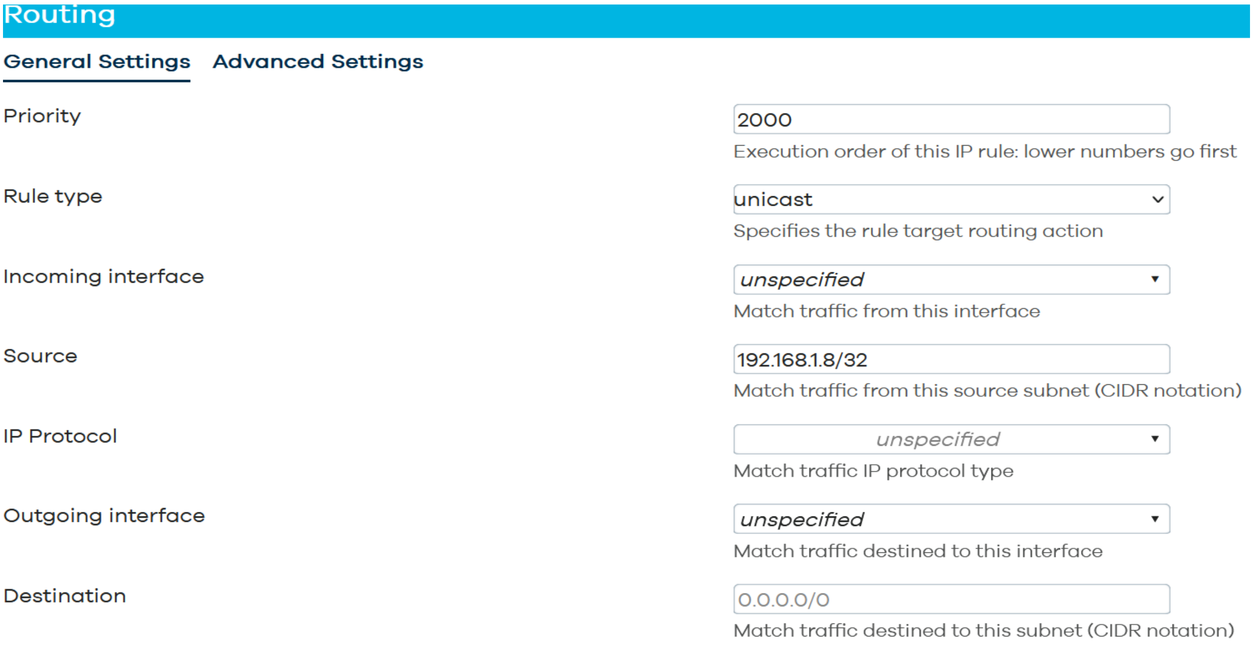
IPv4

Excample for setting one LAN client (192.168.1.8) to use the Alternate routing table 100

LuCi > Network > routing > IPv4 Rules: *Add*

General Settings:

- **Priority:** *2000* (Optional)
- **Rule Type:** *unicast* (default)
- **Incoming interface:** *unspecified* (default)
- **Source:** *192.168.1.8/32* (the IP address of your LAN client in CIDR notation, you can also set a whole subnet)
- **IP protocol:** *unspecified* (default)
- **Outgoing interface:** *unspecified* (default)
- **Destination:** *0.0.0.0/0* (default)



Priority: 2000
Execution order of this IP rule: lower numbers go first

Rule type: unicast
Specifies the rule target routing action

Incoming interface: *unspecified*
Match traffic from this interface

Source: 192.168.1.8/32
Match traffic from this source subnet (CIDR notation)

IP Protocol: *unspecified*
Match traffic IP protocol type

Outgoing interface: *unspecified*
Match traffic destined to this interface

Destination: 0.0.0.0/0
Match traffic destined to this subnet (CIDR notation)

Advanced Settings:
Table: *100* (Dropdown box: choose *custom* and set the table number of the alternate routing table)

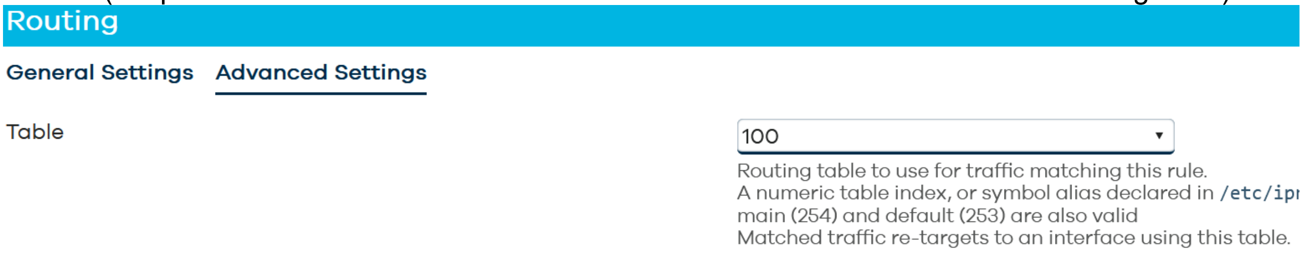


Table: 100
Routing table to use for traffic matching this rule.
A numeric table index, or symbol alias declared in `/etc/ipr`
main (254) and default (253) are also valid
Matched traffic re-targets to an interface using this table.

For a **whole subnet** use as **Source:** *192.168.1.0/24*
For an **interface** e.g. your lan interface use: **Outgoing interface:** *lan*
For a **source port**, useful if you want to route the port of a WireGuard server via the wan in case you also have a Wireguard client with default route via the VPN, on Advanced Settings > **Source Port:** *51820*

/etc/config/network:

```
config rule
        # for ip source:
        option src '192.168.1.8/32' or '192.168.1.0/24'
        # destination e.g. from all to dest
        option dest '25.52.71.40/32'
        # for interface
        option in 'lan'
        # for proto
        option ipproto 'icmp`
        # for source port
        option sport '51820'
        # for destination port
        option dport '116'
        #table number to use for lookup
        option lookup '100'
        option priority '2000'
```

If you want to use local routes in your new routing table then use suppress prefix length with lower priority then routing rules:

/etc/config/network:

```
config rule 'policy_localroutes'
        option lookup 'main'
        option suppress_prefixlength '0'
        option priority '1000'
```

IPv6

For IPv6 you do the same on **LuCi > Network > routing > IPv6 Rules**
Using source addresses is much more difficult as your lan clients will often use temporary IPv6 addresses so use ULA or LL addresses

/etc/config/network:

```
config rule6
        # for ip source: this is difficult as IPv6 uses temporary addresses, consider
using the ULA or LL addresses
        # for interface
        #option in 'lan'
        # for proto
        option ipproto 'icmp`
        # for source port
        option sport '51820'
        # for destination port
        option dport '116'
        #table number to use for lookup
        option lookup '100'
        option priority '2000'
```

For Local IPv6 routes:

```
config rule6
        option priority '1000'
        option lookup 'main'
        option suppress_prefixlength '0'
```