Welcome to docs.openwrt.melmac.net!

VPN Policy-Based Routing

Description

This service allows you to define rules (policies) for routing traffic via WAN or your L2TP, Openconnect, OpenVPN, PPTP or Wireguard tunnels. Policies can be set based on any combination of local/remote ports, local/remote IPv4 or IPv6 addresses/subnets or domains. This service supersedes the `VPN Bypass` available on GitHub/jsDelivr service, by supporting IPv6 and by allowing you to set explicit rules not just for WAN interface (bypassing OpenVPN tunnel), but for L2TP, Openconnect, OpenVPN, PPTP and Wireguard tunnels as well.

Features

Gateways/Tunnels

- Any policy can target either WAN or a VPN tunnel interface.
- L2TP tunnels supported (with protocol names l2tp*).
- Openconnect tunnels supported (with protocol names openconnect*).
- OpenVPN tunnels supported (with device names tun*).[#1] [#2]
- PPTP tunnels supported (with protocol names pptp*).
- Wireguard tunnels supported (with protocol names wireguard*).

IPv4/IPv6/Port-Based Policies

- Policies based on local names, IPs or subnets. You can specify a single IP (as in `192.168.1.70`) or a local subnet (as in `192.168.1.81/29`) or a local device name (as in `nexusplayer`). IPv6 addresses are also supported.
- Policies based on local ports numbers. Can be set as an individual port number (`32400`), a range (`5060-5061`), a space-separated list (`80 8080`) or a combination of the above (`80 8080 5060-5061`). Limited to 15 space-separated entries per policy.
- Policies based on remote IPs/subnets or domain names. Same format/syntax as local IPs/subnets.
- Policies based on remote ports numbers. Same format/syntax and restrictions as local ports.
- You can mix the IP addresses/subnets and device (or domain) names in one field separating them by space (like this: `66.220.2.74 he.net tunnelbroker.net`).
- See Policy Options section for more information.

Domain-Based Policies

- Policies based on (remote) domain names can be processed in different ways, please review the Policy Options section and Footnotes/Known Issues section, specifically [#5] and any other information in that section relevant to domain-based routing/DNS.

Physical Device Policies

- Policies based on a local physical device (like a specially created wlan), please review the Policy Options section and Footnotes/Known Issues section, specifically [#6] and any other information in that section relevant to handling physical device.

DSCP Tag-Based Policies

You can also set policies for traffic with specific DSCP tag. On Windows 10, for example, you can mark traffic from specific apps with DSCP tags (instructions for tagging specific app traffic in Windows 10 can be found here).

Custom User Files

If the custom user file includes are set, the service will load and execute them after setting up ip tables and ipsets and processing policies. This allows, for example, to add large numbers of domains/IP addresses to ipsets without manually adding all of them to the config file.

Two example custom user-files are provided: `/etc/vpn-policy-routing.aws.user` and `/etc/vpn-policy-routing.netflix.user`. They are provided to pull the AWS and Netflix IP addresses into the `wan` ipset respectively.

Strict Enforcement

- Supports strict policy enforcement, even if the policy interface is down – resulting in network being unreachable for specific policy (enabled by default).

Use Resolver's ipset

- If supported on the system, service can be set to utilize resolver's ipset support. Currently supported resolver's ipset options are listed below.

Use DNSMASQ ipset

- Service can be set to utilize `dnsmasq`'s `ipset` support, which requires the `dnsmasq-full` package to be installed (see How to install dnsmasq-full). This significantly improves the start up time because `dnsmasq` resolves the domain names and adds them to appropriate `ipset` in background. Another benefit of using `dnsmasq`'s `ipset` is that it also automatically adds third-level domains to the `ipset`: if `domain.com` is added to the policy, this policy will affect all `*.domain.com` subdomains. This also works for top-level domains as well, a policy targeting the `at` for example, will affect all the `*.at` domains.
- Please review the Footnotes/Known Issues section, specifically [#5] and [#7] and any other information in that section relevant to domain-based routing/DNS.

Customization

- Can be fully configured with `uci` commands or by editing `/etc/config/vpn-policy-routing` file.
- Has a companion package (`luci-app-vpn-policy-routing`) so policies can be configured with Web UI.

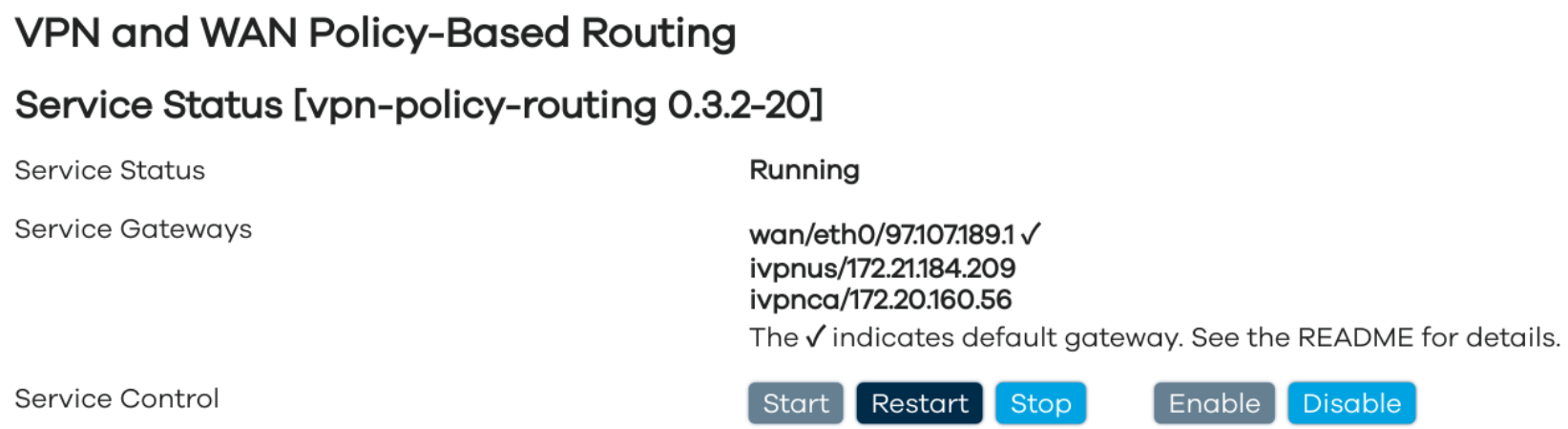
Other Features

- Doesn't stay in memory, creates the routing tables and `iptables` rules/ `ipset` entries which are automatically updated when supported/monitored interface changes.

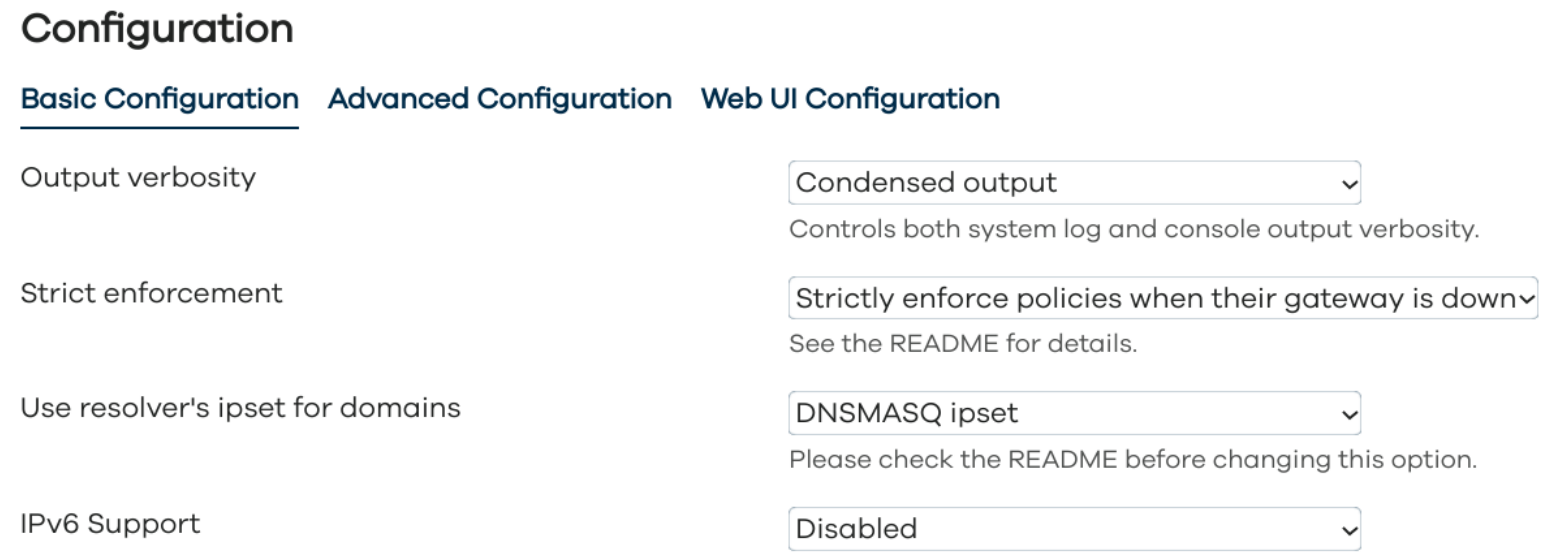
- Proudly made in 🍁 Canada 🍁, using locally-sourced electrons.

Screenshots (luci-app-vpn-policy-routing)

Service Status



Configuration - Basic Configuration



Configuration - Advanced Configuration

Configuration

Basic Configuration | Advanced Configuration | Web UI Configuration

WARNING: Please make sure to check the README before changing anything in this section! Change any of the settings below with extreme caution!

Supported Interfaces

Allows to specify the list of interface names (in lower case) to be explicitly supported by the service. Can be useful if your OpenVPN tunnels have dev option other than tun* or tap*.

Ignored Interfaces

vpnserver wgserver

wgserver

Allows to specify the list of interface names (in lower case) to be ignored by the service. Can be useful if running both VPN server and VPN client on the router.

Boot Time-out

30

Time (in seconds) for service to wait for WAN gateway discovery on boot.

The ipset option for remote policies

Disabled

Please check the README before changing this option.

The ipset option for local policies

Disabled

Please check the README before changing this option.

IPTables rule option

Append

Select Append for -A and Insert for -I.

Default ICMP Interface

No Change

Force the ICMP protocol interface.

WAN Table ID

201

Starting (WAN) Table ID number for tables created by the service.

WAN Table FW Mark

0x010000

Starting (WAN) FW Mark for marks used by the service. High starting mark is used to avoid conflict with SQM/QoS. Change with caution together with Service FW Mask.

Service FW Mask

0xff0000

FW Mask used by the service. High mask is used to avoid conflict with SQM/QoS. Change with caution together with WAN Table FW Mark.

Configuration - WebUI Configuration

Configuration

Basic Configuration | Advanced Configuration | Web UI Configuration

Show Enable Column

Enabled

Shows the enable checkbox column for policies, allowing you to quickly enable/disable specific policy without deleting it.

Show Protocol Column

Enabled

Shows the protocol column for policies, allowing you to assign a specific protocol to a policy.

Supported Protocols

tcp -
udp -
tcp udp -
icmp -
all -
+

Display these protocols in protocol column in Web UI.

Show Chain Column

Enabled

Shows the chain column for policies, allowing you to assign a PREROUTING, FORWARD, INPUT or OUTPUT chain to a policy.

Add IGNORE Target

Enabled

Adds `IGNORE` to the list of interfaces for policies, allowing you to skip further processing by VPN Policy Routing.

Show Up/Down Buttons

Disabled

Shows the Up/Down buttons for policies, allowing you to move a policy up or down in the list.

Policies

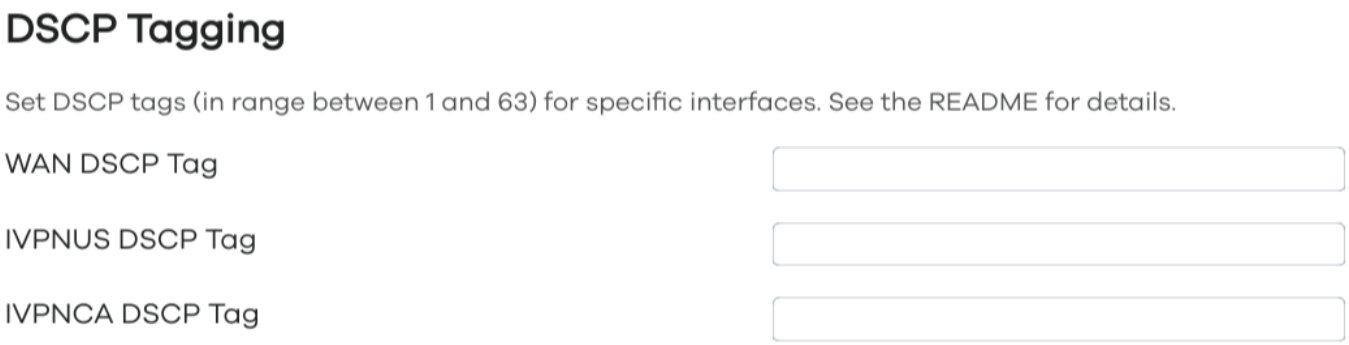
Policies

Comment, interface and at least one other field are required. Multiple local and remote addresses/devices/domains and ports can be space separated. Placeholders below represent just the format/syntax and will not be used if fields are left blank.

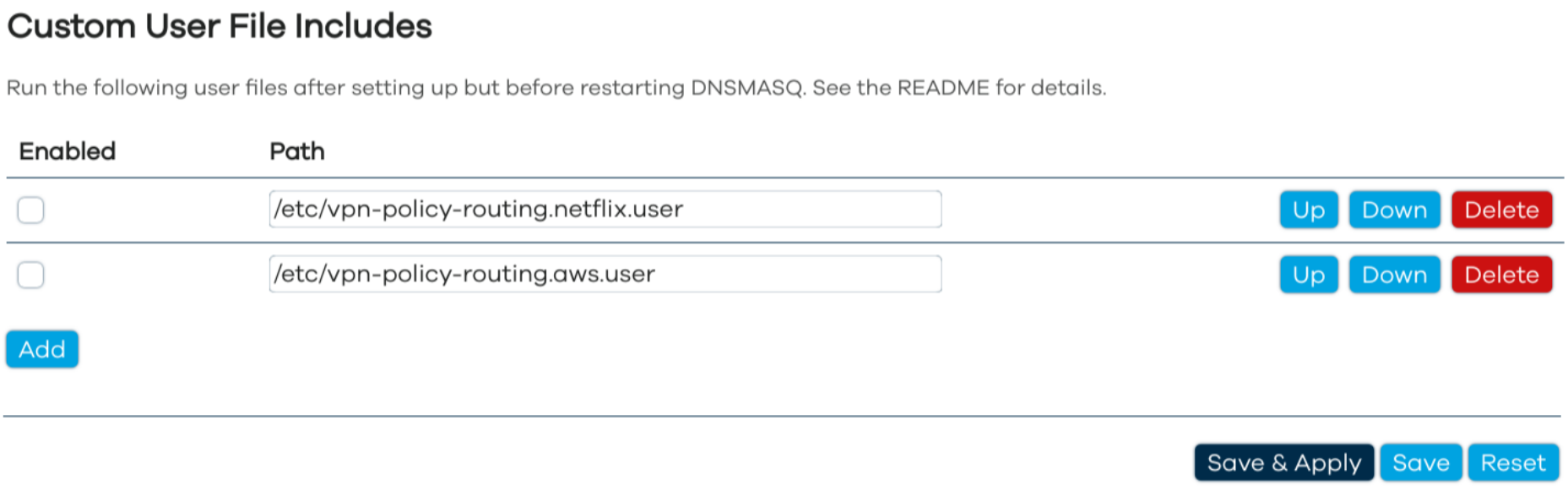
Enabled	Name	Local addresses / devices	Local ports	Remote addresses / domains	Remote ports	Protocol	Chain	Interface	
■	Ignore Local		0-65535	192.168.200.0/2	0-65535	AUTO	PRER	IGNORI	Delete
■	WireGuard S		51820	0.0.0.0/0	0-65535	UDP	OUTF	WAN	Delete
■	Local IP	192.168.1.70	0-65535	0.0.0.0/0	0-65535	AUTO	PRER	WAN	Delete
■	Local Subne	192.168.1.81/29	0-65535	0.0.0.0/0	0-65535	AUTO	PRER	WAN	Delete
■	Local Machi	dell-ubuntu	0-65535	0.0.0.0/0	0-65535	AUTO	PRER	WAN	Delete
■	Local MAC A	00:0F:EA:91:0	0-65535	0.0.0.0/0	0-65535	AUTO	PRER	WAN	Delete
■	Local Device	192.168.1.70 192	0-65535	0.0.0.0/0	0-65535	AUTO	PRER	WAN	Delete
■	Local IP	192.168.1.70	0-65535	0.0.0.0/0	0-65535	AUTO	PRER	WAN	Delete
■	Local Subne	192.168.1.81/29	0-65535	0.0.0.0/0	0-65535	AUTO	PRER	WAN	Delete
■	Local Machi	dell-ubuntu	0-65535	0.0.0.0/0	0-65535	AUTO	PRER	WAN	Delete
■	Local MAC A	00:0F:EA:91:0	0-65535	0.0.0.0/0	0-65535	AUTO	PRER	WAN	Delete
■	Local Device	192.168.1.70 192	0-65535	0.0.0.0/0	0-65535	AUTO	PRER	WAN	Delete
■	Plex Local S		32400	0.0.0.0/0	0-65535	AUTO	PRER	WAN	Delete
■	Plex Remote		0-65535	plex.tv my.plex	0-65535	AUTO	PRER	WAN	Delete
■	Emby Local		8096 8920	0.0.0.0/0	0-65535	AUTO	PRER	WAN	Delete
■	Emby Remot		0-65535	emby.media ap	0-65535	AUTO	PRER	WAN	Delete

Add

DSCP Tagging



Custom User File Includes



How It Works

On start, this service creates routing tables for each supported interface (WAN/WAN6 and VPN tunnels) which are used to route specially marked packets. For the `mangle` table's `PREROUTING`, `FORWARD`, `INPUT` and `OUTPUT` chains, the service creates corresponding `VPR_*` chains to which policies are assigned. Evaluation and marking of packets happen in these `VPR_*` chains. If enabled, the service also creates the remote/local ipsets per each supported interface and the corresponding `iptables` rule for marking packets matching the `ipset`. The service then processes the user-created policies.

Processing Policies

Each policy can result in either a new `iptables` rule or, if `src_ipset` or `dest_ipset` or `resolver_ipset` are enabled, an `ipset` or a `dnsmasq`'s `ipset` entry.

- Policies with local MAC-addresses, IP addresses or local device names can be created as `iptables` rules or `ipset` entries.
- Policies with local or remote ports are always created as `iptables` rules.
- Policies with local or remote netmasks can be created as `iptables` rules or `ipset` entries.
- Policies with **only** remote IP address or a domain name can be created as `iptables` rules or `dnsmasq`'s `ipset` or an `ipset` (if enabled).

Policies Priorities

- If support for `src_ipset`, `dest_ipset` and `resolver_ipset` is disabled, then only `iptables` rules are created. The policy priority is the same as its order as listed in Web UI and `/etc/config/vpn-policy-routing`. The higher the policy is in the Web UI and configuration file, the higher its priority is.
- If support for `src_ipset`, `dest_ipset` or `resolver_ipset` is enabled, then the `ipset` / `dnsmasq.ipset` entries have the highest priority (irrelevant of their position in the policies list) and the other policies are processed in the same order as they are listed in Web UI and `/etc/config/vpn-policy-routing`.
- If there are conflicting `ipset` / `dnsmasq.ipset` entries for different interfaces, the priority is given to the interface which is listed first in the `/etc/config/network` file.
- If set, the `DSCP` policies trump all other policies, including the `ipset` / `dnsmasq.ipset` ones.
- If enabled, it is highly recommended that the policies with `IGNORE` target are at the top of the policies list.

How To Install

Please make sure that the requirements are satisfied and install `vpn-policy-routing` and `luci-app-vpn-policy-routing` from Web UI or connect to your router via ssh and run the following commands:

```
opkg update
opkg install vpn-policy-routing luci-app-vpn-policy-routing
```

If these packages are not found in the official feed/repo for your version of OpenWrt, you will need to add a custom repo to your router following instructions on GitHub/jsDelivr first.

These packages have been designed to be backwards compatible with OpenWrt 19.07, OpenWrt 18.06, OpenWrt Project 17.01 and OpenWrt 15.05. However, on systems older than OpenWrt 18.06.6 and/or a system which has deviated too far (or haven't been updated to keep in-sync) with official OpenWrt release you may get a message about missing `luci-compat` dependency, which (and only which) you can safely ignore and force-install the luci app using `opkg install --force-depends` command instead of `opkg install` .

Requirements

This service requires the following packages to be installed on your router: `ipset` , `resolveip` , `ip-full` (or a `busybox` built with `ip` support), `kmod-ipt-ipset` and `iptables` .

To satisfy the requirements, connect to your router via ssh and run the following commands:

```
opkg update; opkg install ipset resolveip ip-full kmod-ipt-ipset iptables
```

How to install dnsmasq-full

If you want to use `dnsmasq` 's `ipset` support, you will need to install `dnsmasq-full` instead of the `dnsmasq` . To do that, connect to your router via ssh and run the following command:

```
opkg update; cd /tmp/ && opkg download dnsmasq-full; opkg install ipset libnettle8 libnetfilter-conntrack3;
opkg remove dnsmasq; opkg install dnsmasq-full --cache /tmp/; rm -f /tmp/dnsmasq-full*.ipk;
```

Unmet dependencies

If you are running a development (trunk/snapshot) build of OpenWrt on your router and your build is outdated (meaning that packages of the same revision/commit hash are no longer available and when you try to satisfy the requirements you get errors), please flash either current OpenWrt release image or current development/snapshot image.

Service Configuration Settings

As per screenshots above, in the Web UI the `vpn-policy-routing` configuration is split into `Basic`, `Advanced` and `WebUI` settings. The full list of configuration parameters of `vpn-policy-routing.config` section is:

Web UI Section	Parameter	Type	Default	Description
Basic	enabled	boolean	0	Enable/disable the `vpn-policy-routing` service.
Basic	verbosity	integer	2	Can be set to 0, 1 or 2 to control the console and system log output verbosity of the `vpn-policy-routing` service.
Basic	strict_enforcement	boolean	1	Enforce policies when their interface is down. See Strict enforcement for more details.

Web UI Section	Parameter	Type	Default	Description
Basic	resolver_ipset	string	dnsmasq.ipset	Enable/disable use of the resolver ipset option for domain-only remote policies (policies with only a domain as a remote address and no other fields set). This speeds up service start-up and operation. Currently supported options are `none` and `dnsmasq.ipset` (see Use DNSMASQ ipset and #7 for more details). Make sure the requirements are met.
Basic	ipv6_enabled	boolean	0	Enable/disable IPv6 support.
Advanced	supported_interface	list/string		Allows to specify the space-separated list of interface names (in lower case) to be explicitly supported by the `vpn-policy-routing` service. Can be useful if your OpenVPN tunnels have dev option other than tun*.
Advanced	ignored_interface	list/string		Allows to specify the space-separated list of interface names (in lower case) to be ignored by the `vpn-policy-routing` service. Can be useful if running both VPN server and VPN client on the router.

Web UI Section	Parameter	Type	Default	Description
Advanced	boot_timeout	number	30	Allows to specify the time (in seconds) for `vpn-policy-routing` service to wait for WAN gateway discovery on boot. Can be useful on devices with ADSL modem built in.
Advanced	dest_ipset	boolean	0	Enable/disable use of one of the ipset options for compatible remote policies (policies with only a remote address and no other fields set). This speeds up service start-up and operation. Make sure the requirements are met.
Advanced	src_ipset	boolean	0	Enable/disable use of `ipset` entries for compatible local policies (policies with only a local IP or MAC address and no other fields set). Using `ipset` for local IPs/MACs is faster than using `iptables` rules, however it makes it impossible to enforce policies priority/order. Make sure the requirements are met.

Web UI Section	Parameter	Type	Default	Description
Advanced	iptables_rule_option	append/insert	append	Allows to specify the iptables parameter for rules: `-A` for `append` and `-I` for `insert`. Append is generally speaking more compatible with other packages/firewall rules. Recommended to change to `insert` only to enable compatibility with the `mwan3` package.
Advanced	icmp_interface	string		Set the default ICMP protocol interface (interface name in lower case). Use with caution.
Advanced	wan_tid	integer	201	Starting (WAN) Table ID number for tables created by the `vpn-policy-routing` service.
Advanced	wan_mark	hexadecimal	0x010000	Starting (WAN) fw mark for marks used by the `vpn-policy-routing` service. High starting mark is used to avoid conflict with SQM/QoS, this can be changed by user. Change with caution together with `fw_mask`.
Advanced	fw_mask	hexadecimal	0xff0000	FW Mask used by the `vpn-policy-routing` service. High mask is used to avoid conflict with SQM/QoS, this can be changed by user. Change with caution together with `wan_mark`.

Web UI Section	Parameter	Type	Default	Description
Hidden/Experimental	quick_table_create	boolean	0	When enabled, only carries the 'default' entry from the main routing table to the routing tables created for each supported interface.
Web UI	webui_enable_column	boolean	0	Shows `Enable` checkbox column for policies, allowing to quickly enable/disable specific policy without deleting it.
Web UI	webui_protocol_column	boolean	0	Shows `Protocol` column for policies, allowing to specify the protocol for `iptables` rules for policies.
Web UI	webui_supported_protocol	list	0	List of protocols to display in the `Protocol` column for policies.
Web UI	webui_chain_column	boolean	0	Shows `Chain` column for policies, allowing to specify `PREROUTING` (default), `FORWARD`, `INPUT`, or `OUTPUT` chain for `iptables` rules for policies.
Web UI	webui_show_ignore_target	boolean	0	Adds `IGNORE` to the list of interfaces for policies, allowing you to skip further processing by VPN Policy Routing.
Web UI	webui_sorting	boolean	1	Shows the Up/Down buttons for policies, allowing you to move a policy up or down in the list/priority.

Web UI Section	Parameter	Type	Default	Description
	wan_dscp	integer		Allows use of DSCP-tag based policies for WAN interface.
	{interface_name}_dscp	integer		Allows use of DSCP-tag based policies for a VPN interface.
	procd_reload_delay	integer	0	Time (in seconds) for PROCD_RELOAD_DELAY parameter.

Default Settings

Default configuration has service disabled (use Web UI to enable/start service or run `uci set vpn-policy-routing.config.enabled=1; uci commit vpn-policy-routing;`).

Policy Options

Each policy may have a combination of the options below, the `name` and `interface` options are required.

The `src_addr` , `src_port` , `dest_addr` and `dest_port` options supports parameter negation, for example if you want to **exclude** remote port 80 from the policy, set `dest_port="!80"` (notice lack of space between `!` and parameter).

Option	Default	Description
name		Policy name, it **must** be set.
enabled	1	Enable/disable policy. To display the `Enable` checkbox column for policies in the WebUI, make sure to select `Enabled` for `Show Enable Column` in the `Web UI` tab.
interface		Policy interface, it **must** be set.

Option	Default	Description
src_addr		List of space-separated local/source IP addresses, CIDRs, hostnames or mac addresses (colon-separated). You can also specify a local physical device (like a specially created wlan) prepended by an `@` symbol.
src_port		List of space-separated local/source ports or port-ranges.
dest_addr		List of space-separated remote/target IP addresses, CIDRs or hostnames/domain names.
dest_port		List of space-separated remote/target ports or port-ranges.
proto	auto	Policy protocol, can be any valid protocol from `/etc/protocols` for CLI/uci or can be selected from the values set in `webui_supported_protocol`. To display the `Protocol` column for policies in the WebUI, make sure to select `Enabled` for `Show Protocol Column` in the `Web UI` tab. Special cases: `auto` will try to intelligently insert protocol-agnostic policy and fall back to TCP/UDP if the protocol must be selected for specific policy; `all` will always insert a protocol-agnostic policy (which may fail depending on the policy).
chain	PREROUTING	Policy chain, can be either `PREROUTING`, `FORWARDING`, `INPUT` or `OUTPUT`. This setting is case-sensitive. To display the `Chain` column for policies in the WebUI, make sure to select `Enabled` for `Show Chain Column` in the `Web UI` tab.

Custom User Files Include Options

Option	Default	Description
path		Path to a custom user file (in a form of shell script), it **must** be set.
enabled	1	Enable/disable setting.

Example Policies

Single IP, IP Range, Local Machine, Local MAC Address

The following policies route traffic from a single IP address, a range of IP addresses, a local machine (requires definition as DHCP host record in DHCP config), a MAC-address of a local device and finally all of the above via WAN.

```
config policy
  option name 'Local IP'
  option interface 'wan'
  option src_addr '192.168.1.70'

config policy
  option name 'Local Subnet'
  option interface 'wan'
  option src_addr '192.168.1.81/29'

config policy
  option name 'Local Machine'
  option interface 'wan'
  option src_addr 'dell-ubuntu'

config policy
  option name 'Local MAC Address'
  option interface 'wan'
  option src_addr '00:0F:EA:91:04:08'

config policy
  option name 'Local Devices'
  option interface 'wan'
  option src_addr '192.168.1.70 192.168.1.81/29 dell-ubuntu 00:0F:EA:91:04:08'
```

Logmein Hamachi

The following policy routes LogMeIn Hamachi zero-setup VPN traffic via WAN.

```
config policy
  option name 'LogmeIn Hamachi'
  option interface 'wan'
  option dest_addr '25.0.0.0/8 hamachi.cc hamachi.com logmein.com'
```

SIP Port

The following policy routes standard SIP port traffic via WAN for both TCP and UDP protocols.

```
config policy
  option name 'SIP Ports'
  option interface 'wan'
  option dest_port '5060'
  option proto 'tcp udp'
```

Plex Media Server

The following policies route Plex Media Server traffic via WAN. Please note, you'd still need to open the port in the firewall either manually or with the UPnP.

```
config policy
  option name 'Plex Local Server'
  option interface 'wan'
  option src_port '32400'

config policy
  option name 'Plex Remote Servers'
  option interface 'wan'
  option dest_addr 'plex.tv my.plexapp.com'
```

Emby Media Server

The following policy route Emby traffic via WAN. Please note, you'd still need to open the port in the firewall either manually or with the UPnP.

```
config policy
  option name 'Emby Local Server'
  option interface 'wan'
  option src_port '8096 8920'

config policy
  option name 'Emby Remote Servers'
  option interface 'wan'
  option dest_addr 'emby.media app.emby.media tv.emby.media'
```

Ignore Requests (replace `append_src_rules`)

Since the `append_src_rules` option is no longer supported in vpn-policy-routing from version 0.3.x forward, replace:

```
config vpn-policy-routing 'config'
  ...
  append_src_rules '! -d 192.168.200.0/24'
```

With the following policy allowing you to skip processing of some requests (like traffic to an OpenVPN or Wireguard server running on the router):

```
config vpn-policy-routing 'config'
  ...
  option webui_show_ignore_target '1'

config policy
  option name 'Ignore Local Requests by Destination'
  option interface 'ignore'
  option dest_addr '192.168.200.0/24'
```

It's a good idea to keep the policies targeting `ignore` interface at the top of the config file/list of policies displayed in WebUI to make sure they are processed first.

Ignore Requests (replace `append_dest_rules`)

Since the `append_dest_rules` option is no longer supported in vpn-policy-routing from version 0.3.x forward, replace:

```
config vpn-policy-routing 'config'
  ...
  append_dest_rules '! -s 192.168.1.1/24'
```

With the following policy allowing you to skip processing of some requests:

```
config vpn-policy-routing 'config'
  ...
  option webui_show_ignore_target '1'

config policy
  option name 'Ignore Local Requests by Source'
  option interface 'ignore'
  option src_addr '192.168.1.1/24'
```

It's a good idea to keep the policies targeting `ignore` interface at the top of the config file/list of policies displayed in WebUI to make sure they are processed first.

Local OpenVPN Server + OpenVPN Client (Scenario 1)

If the OpenVPN client on your router is used as default routing (for the whole internet), make sure your settings are as following (three dots on the line imply other options can be listed in the section as well).

Relevant part of `/etc/config/vpn-policy-routing`:

```
config vpn-policy-routing 'config'
  list ignored_interface 'vpnserver'
  ...

config policy
  option name 'OpenVPN Server'
  option interface 'wan'
  option proto 'tcp'
  option src_port '1194'
  option chain 'OUTPUT'
```

The network/firewall/openvpn settings are below.

Relevant part of `/etc/config/network` (**DO NOT** modify default OpenWrt network settings for neither `wan` nor `lan`):

```
config interface 'vpnclient'
  option proto 'none'
  option ifname 'ovpnc0'

config interface 'vpnserver'
  option proto 'none'
  option ifname 'ovpns0'
  option auto '1'
```

Relevant part of `/etc/config/firewall` (**DO NOT** modify default OpenWrt firewall settings for neither `wan` nor `lan`):

```
config zone
  option name 'vpnclient'
  option network 'vpnclient'
  option input 'REJECT'
  option forward 'ACCEPT'
  option output 'REJECT'
  option masq '1'
```

```
    option mtu_fix '1'

config forwarding
    option src 'lan'
    option dest 'vpnclient'

config zone
    option name 'vpnserver'
    option network 'vpnserver'
    option input 'ACCEPT'
    option forward 'REJECT'
    option output 'ACCEPT'
    option masq '1'

config forwarding
    option src 'vpnserver'
    option dest 'wan'

config forwarding
    option src 'vpnserver'
    option dest 'lan'

config forwarding
    option src 'vpnserver'
    option dest 'vpnclient'

config rule
    option name 'Allow-OpenVPN-Inbound'
    option target 'ACCEPT'
    option src '*'
    option proto 'tcp'
    option dest_port '1194'
```

Relevant part of `/etc/config/openvpn`:

```
config openvpn 'vpnclient'
  option client '1'
  option dev_type 'tun'
  option dev 'ovpnc0'
  option proto 'udp'
  option remote 'some.domain.com 1197' # DO NOT USE PORT 1194 for VPN Client
  ...

config openvpn 'vpnserver'
  option port '1194'
  option proto 'tcp'
  option server '192.168.200.0 255.255.255.0'
  ...
```

Local OpenVPN Server + OpenVPN Client (Scenario 2)

If the OpenVPN client is **not** used as default routing and you create policies to selectively use the OpenVPN client, make sure your settings are as following (three dots on the line imply other options can be listed in the section as well). Make sure that the policy mentioned below is at the top of your policies list.

Relevant part of `/etc/config/vpn-policy-routing` :

```
config vpn-policy-routing 'config'
  list ignored_interface 'vpnserver'
  ...
config policy
  option name 'Ignore Local Traffic'
  option interface 'ignore'
  option dest_address '192.168.200.0/24'
  ...
```

The network/firewall/openvpn settings are below.

Relevant part of `/etc/config/network` (**DO NOT** modify default OpenWrt network settings for neither `wan` nor `lan`):

```
config interface 'vpnclient'
  option proto 'none'
  option ifname 'ovpnc0'

config interface 'vpnserver'
  option proto 'none'
  option ifname 'ovpns0'
  option auto '1'
```

Relevant part of `/etc/config/firewall` (**DO NOT** modify default OpenWrt firewall settings for neither `wan` nor `lan`):

```
config zone
  option name 'vpnclient'
  option network 'vpnclient'
  option input 'REJECT'
  option forward 'ACCEPT'
  option output 'REJECT'
  option masq '1'
  option mtu_fix '1'

config forwarding
  option src 'lan'
  option dest 'vpnclient'

config zone
  option name 'vpnserver'
  option network 'vpnserver'
  option input 'ACCEPT'
  option forward 'REJECT'
  option output 'ACCEPT'
  option masq '1'

config forwarding
```

```
    option src 'vpnserver'
    option dest 'wan'

  config forwarding
    option src 'vpnserver'
    option dest 'lan'

  config forwarding
    option src 'vpnserver'
    option dest 'vpnclient'

  config rule
    option name 'Allow-OpenVPN-Inbound'
    option target 'ACCEPT'
    option src '*'
    option proto 'tcp'
    option dest_port '1194'
```

Relevant part of `/etc/config/openvpn`:

```
  config openvpn 'vpnclient'
    option client '1'
    option dev_type 'tun'
    option dev 'ovpnc0'
    option proto 'udp'
    option remote 'some.domain.com 1197' # DO NOT USE PORT 1194 for VPN Client
    list pull_filter 'ignore "redirect-gateway"' # for OpenVPN 2.4 and later
    option route_nopull '1' # for OpenVPN earlier than 2.4
    ...

  config openvpn 'vpnserver'
    option port '1194'
    option proto 'tcp'
    option server '192.168.200.0 255.255.255.0'
    ...
```

Local Wireguard Server + Wireguard Client (Scenario 1)

Yes, I'm aware that technically there are no clients nor servers in Wireguard, it's all peers, but for the sake of README readability I will use the terminology similar to the OpenVPN Server + Client setups.

If the Wireguard tunnel on your router is used as default routing (for the whole internet), sadly no `vpn-policy-routing` rule will allow it to intercept and properly route the `UDP` traffic of Wireguard server, please either use the OpenVPN server and configure it to use `TCP` protocol or use the Scenario 2 below.

Local Wireguard Server + Wireguard Client (Scenario 2)

Yes, I'm aware that technically there are no clients nor servers in Wireguard, it's all peers, but for the sake of README readability I will use the terminology similar to the OpenVPN Server + Client setups.

If the Wireguard client is **not** used as default routing and you create policies to selectively use the Wireguard client, make sure your settings are as following (three dots on the line imply other options can be listed in the section as well). Make sure that the policy mentioned below is at the top of your policies list.

Relevant part of `/etc/config/vpn-policy-routing`:

```
config vpn-policy-routing 'config'
  list ignored_interface 'wgserver'
  ...
config policy
  option name 'Ignore Local Traffic'
  option interface 'ignore'
  option dest_address '192.168.200.0/24'
  ...
```

The recommended network/firewall settings are below.

Relevant part of `/etc/config/network` (**DO NOT** modify default OpenWrt network settings for neither `wan` nor `lan`):

```
config interface 'wgclient'
  option proto 'wireguard'
  ...

config wireguard_wgclient
  list allowed_ips '0.0.0.0/0'
  list allowed_ips '::0/0'
  option endpoint_port '51820'
  ...

config interface 'wgserver'
  option proto 'wireguard'
  option listen_port '61820'
  list addresses '192.168.200.1/24'
  ...

config wireguard_wgserver
  list allowed_ips '192.168.200.2/32'
  option route_allowed_ips '1'
  ...
```

Relevant part of `/etc/config/firewall` (**DO NOT** modify default OpenWrt firewall settings for neither `wan` nor `lan`):

```
config zone
  option name 'wgclient'
  option network 'wgclient'
  option input 'REJECT'
  option forward 'ACCEPT'
  option output 'REJECT'
  option masq '1'
  option mtu_fix '1'

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

```
config zone
  option name 'wgserver'
  option network 'wgserver'
  option input 'ACCEPT'
  option forward 'REJECT'
  option output 'ACCEPT'
  option masq '1'

config forwarding
  option src 'wgserver'
  option dest 'wan'

config forwarding
  option src 'wgserver'
  option dest 'lan'

config forwarding
  option src 'wgserver'
  option dest 'wgclient'

config rule
  option name 'Allow-WG-Inbound'
  option target 'ACCEPT'
  option src '*'
  option proto 'udp'
  option dest_port '61820'
```

Netflix Domains

The following policy should route US Netflix traffic via WAN. For capturing international Netflix domain names, you can refer to the getdomainnames.sh-specific instructions on GitHub/jsDelivr and don't forget to adjust them for OpenWrt. This may not work if Netflix changes things. For more reliable US Netflix routing you may want to consider also using custom user files.

```
config policy
  option name 'Netflix Domains'
  option interface 'wan'
  option dest_addr 'amazonaws.com netflix.com nflxext.com nflximg.net nflxso.net nflxvideo.net dvd.netflix.com'
```

Example Custom User Files Includes

```
config include
  option path '/etc/vpn-policy-routing.netflix.user'

config include
  option path '/etc/vpn-policy-routing.aws.user'
```

Basic OpenVPN Client Config

There are multiple guides online on how to configure the OpenVPN client on OpenWrt "the easy way", and they usually result either in a kill-switch configuration or configuration where the OpenVPN tunnel cannot be properly (and separately from WAN) routed, either way, incompatible with the VPN Policy-Based Routing.

Below is the sample OpenVPN client configuration for OpenWrt which is guaranteed to work. If you have already deviated from the instructions below (ie: made any changes to any of the `wan` or `lan` configurations in either `/etc/config/network` or `/etc/config/firewall`), you will need to start from scratch with a fresh OpenWrt install.

Relevant part of `/etc/config/vpn-policy-routing`:

```
config vpn-policy-routing 'config'
  list supported_interface 'vpnclient'
  ...
```

The recommended network/firewall settings are below.

Relevant part of `/etc/config/network` (**DO NOT** modify default OpenWrt network settings for neither `wan` nor `lan`):

```
config interface 'vpnclient'
  option proto 'none'
  option ifname 'ovpnc0'
```

Relevant part of `/etc/config/firewall` (**DO NOT** modify default OpenWrt firewall settings for neither `wan` nor `lan`):

```
config zone
  option name 'vpnclient'
  option network 'vpnclient'
  option input 'REJECT'
  option forward 'REJECT'
  option output 'ACCEPT'
  option masq '1'
  option mtu_fix '1'

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

If you have a Guest Network, add the following to the `/etc/config/firewall`:

```
config forwarding
  option src 'guest'
  option dest 'vpnclient'
```

Relevant part of `/etc/config/openvpn` (configure the rest of the client connection for your specifics by either referring to an existing `.ovpn` file or thru the OpenWrt uci settings):

```
config openvpn 'vpnclient'
  option enabled '1'
```

```
  option client '1'
  option dev_type 'tun'
  option dev 'ovpnc0'
  ...
```

Multiple OpenVPN Clients

If you use multiple OpenVPN clients on your router, the order in which their devices are named (tun0, tun1, etc) is not guaranteed by OpenWrt. The following settings are recommended in this case.

For `/etc/config/network`:

```
config interface 'vpnclient0'
  option proto 'none'
  option ifname 'ovpnc0'

config interface 'vpnclient1'
  option proto 'none'
  option ifname 'ovpnc1'
```

For `/etc/config/openvpn`:

```
config openvpn 'vpnclient0'
  option client '1'
  option dev_type 'tun'
  option dev 'ovpnc0'
  ...

config openvpn 'vpnclient1'
  option client '1'
  option dev_type 'tun'
  option dev 'ovpnc1'
  ...
```

For `/etc/config/vpn-policy-routing`:

```
config vpn-policy-routing 'config'
  list supported_interface 'vpnclient0 vpnclient1'
  ...
```

Footnotes/Known Issues

1. See note about multiple OpenVPN clients.
2. If your `OpenVPN` interface has the device name different from tun*, is not up and is not explicitly listed in `supported_interface` option, it may not be available in the policies `Interface` drop-down within WebUI.
3. If your default routing is set to the VPN tunnel, then the true WAN interface cannot be discovered using OpenWrt built-in functions, so service will assume your network interface ending with or starting with `wan` is the true WAN interface.
4. The service does **NOT** support the "killswitch" router mode (where if you stop the VPN tunnel, you have no internet connection). For proper operation, leave all the default OpenWrt `network` and `firewall` settings for `lan` and `wan` intact.
5. When using the `dnsmasq.ipset` option, please make sure to flush the DNS cache of the local devices, otherwise domain policies may not work until you do. If you're not sure how to flush the DNS cache (or if the device/OS doesn't offer an option to flush its DNS cache), reboot your local devices when starting to use the service and/or when connecting data-capable device to your WiFi.
6. When using the policies targeting physical devices, make sure you have the following packages installed: `kmod-br-netfilter`, `kmod-ipt-physdev` and `iptables-mod-physdev`. Also, if your physical device is a part of the bridge, you may have to set `net.bridge.bridge-nf-call-iptables` to `1` in your `/etc/sysctl.conf`.
7. Because the `ipset` command only adds a first resolved IP address of the domain on add, if the domain name is encountered as the `dest_addr` option of the policy (with no other fields set for the policy), it will be attempted to be added as `dnsmasq.ipset` (if `resolver_ipset` is set to `dnsmasq.ipset`), otherwise, the domain name will be resolved when the service starts up and the resolved IP addresses added as either `ipset` (if enabled) or `iptables` rules. Resolving a number of domains on start is a time consuming operation, that's why the use of `dnsmasq.ipset` value for `resolver_ipset` options is a preferred scenario.

8. When service is started, it subscribes to the supported interfaces updates thru the PROCD. While I was never able to reproduce the issue, some customers report that this method doesn't always work in which case you may want to set up iface hotplug script to reload service when relevant interface(s) are updated.

FAQ

You may find some useful information in sections below.

A Word About Default Routing

Service does not alter the default routing. Depending on your VPN tunnel settings (and settings of the VPN server you are connecting to), the default routing might be set to go via WAN or via VPN tunnel. This service affects only routing of the traffic matching the policies. If you want to override default routing, follow the instructions below.

OpenVPN tunnel configured via uci (/etc/config/openvpn)

To unset an OpenVPN tunnel as default route, set the following to the appropriate section of your `/etc/config/openvpn`:

- For OpenVPN 2.4 and newer client config:

```
list pull_filter 'ignore "redirect-gateway"'
```

- For OpenVPN 2.3 and older client config:

```
option route_nopull '1'
```

- For your Wireguard (client) config:

```
option route_allowed_ips '0'
```

OpenVPN tunnel configured with .ovpn file

To unset an OpenVPN tunnel as default route, set the following to the appropriate section of your `.ovpn` file:

- For OpenVPN 2.4 and newer client `.ovpn` file:

```
pull-filter ignore "redirect-gateway"
```

- For OpenVPN 2.3 and older client `.ovpn` file:

```
route-nopull "1"
```

Wireguard tunnel

To unset a Wireguard tunnel as default route, set the following to the appropriate section of your `/etc/config/network`:

- For your Wireguard (client) config:

```
option route_allowed_ips '0'
```

- Routing Wireguard traffic may require setting `net.ipv4.conf.wg0.rp_filter = 2` in `/etc/sysctl.conf`. Please refer to issue #41 for more details.

A Word About Cloudflare's 1.1.1.1 App

Cloudflare has released an app for iOS and Android, which can also be configured to route traffic thru their own VPN tunnel (WARP+).

If you use Cloudlfare's VPN tunnel (WARP+), none of the policies you set up with the VPN Policy Routing will take effect on your mobile device. Disable WARP+ for your home WiFi to keep VPN Policy Routing affecting your mobile device.

If you just use the private DNS queries (WARP), A Word About DNS-over-HTTPS applies. You can also disable WARP for your home WiFi to keep VPN Policy Routing affecting your mobile device.

A Word About DNS-over-HTTPS

Some browsers, like Mozilla Firefox or Google Chrome/Chromium have DNS-over-HTTPS proxy built-in. Their requests to web-sites cannot be affected if the `dnsmasq.ipset` is set for the `resolver_ipset` option. To fix this, you can try either of the following:

1. Disable the DNS-over-HTTPS support in your browser and use the OpenWrt's `net/https-dns-proxy` (README on GitHub/jsDelivr) package with optional `https-dns-proxy` WebUI/luci app. You can then continue to use `dnsmasq.ipset` setting for the `resolver_ipset` in VPN Policy Routing.

2. Continue using DNS-over-HTTPS in your browser (which, by the way, also limits your options for router-level AdBlocking as described in `dnsmasq.ipset` option description here of `net/simple-adblock` README on GitHub/jsDelivr), you than would either have to switch the `resolver_ipset` to `none`. Please note, you will lose all the benefits of `dnsmasq.ipset` option.

A Word About HTTP/3 (QUICK)

If you want to target traffic using HTTP/3 protocol, you can use the `AUTO` as the protocol (the policy will be either protocol-agnostic or `TCP/UDP`) or explicitly use `UDP` as a protocol.

A Word About IPv6 Routing

Due to the nature of IPv6, it's not supposed to be routed same way as IPv4 with this package, but a fellow user has graciously contributed a gist detailing their experience to get IPv6 routing working.

A Word About Routing Netflix/Amazon Prime/Hulu Traffic

There are two following scenarios with VPN connections and Netflix/Amazon Prime/Hulu traffic.

Routing Netflix/Amazon Prime/Hulu Traffic via VPN Tunnel

If you live in a country where Netflix/Amazon Prime/Hulu are not available and want to circumvent geo-fencing, this package can't help you. The Netflix/Amazon Prime/Hulu do a great job detecing VPN usage when accessing their services and circumventing geographical restrictions is not only dubiously legal, it's also technically very challenging.

Routing Netflix/Amazon Prime/Hulu Traffic via WAN

If you live in a country where Netflix/Amazon Prime/Hulu are available, you obviously do NOT want to use VPN tunnel for their traffic.

If the VPN tunnel is not used as a default gateway on your router, you should not have a problem accessing Netflix/Amazon Prime/Hulu (just make sure that your DNS requests are not routed via VPN tunnel either).

If the VPN tunnel is used as a default gateway, either:

- send ALL traffic from your multimedia devices (by using their IP addresses or device names in the `src_addr` option in config file or Local addresses /devices field in WebUI) accessing Netflix/Amazon Prime/Hulu to WAN; this is the more reliable and recommended method.
- use the Netflix/AWS custom user files in combination with the Netflix/Amazon Prime/Hulu domains and `dnsmasq.ipset` option to route traffic to Netflix/Amazon via WAN; this is definitely less reliable method and may not work in all regions.

Either way make sure that your DNS requests are not routed via VPN Tunnel!

A Word About Interface Hotplug Script

Sometimes[#8] the service doesn't get reloaded when supported interfaces go up or down. This can be an annoying experience since the service may start before all supported VPN connections are up and then not get updated when the VPN connections get established. In that case, run the following command from CLI to create the interface hotplug script to cause the service to be reloaded in interface updates:

```
mkdir -p /etc/hotplug.d/iface/
cat << 'EOF' > /etc/hotplug.d/iface/70-vpn-policy-routing
#!/bin/sh
logger -t vpn-policy-routing "Reloading due to $ACTION of $INTERFACE ($DEVICE)"
```

```
/etc/init.d/vpn-policy-routing reload
EOF
```

For `vpn-policy-routing` package version 0.3.4 and up you can use the following script which will only reload what's relevant to the reloaded interface:

```
mkdir -p /etc/hotplug.d/iface/
cat << 'EOF' > /etc/hotplug.d/iface/70-vpn-policy-routing
#!/bin/sh
logger -t vpn-policy-routing "Reloading $INTERFACE due to $ACTION of $INTERFACE ($DEVICE)"
/etc/init.d/vpn-policy-routing reload_interface "$INTERFACE"
EOF
```

Getting Help

If things are not working as intended, please include the following in your post:

- content of `/etc/config/dhcp`
- content of `/etc/config/firewall`
- content of `/etc/config/network`
- content of `/etc/config/vpn-policy-routing`
- the output of `/etc/init.d/vpn-policy-routing support`
- the output of `/etc/init.d/vpn-policy-routing reload` with verbosity setting set to 2

If you don't want to post the `/etc/init.d/vpn-policy-routing support` output in a public forum, there's a way to have the support details automatically uploaded to my account at paste.ee by running: `/etc/init.d/vpn-policy-routing support -p` . You need to have the following packages installed to enable paste.ee upload functionality: `curl libopenssl ca-bundle` .

WARNING: while paste.ee uploads are unlisted/not indexed at the web-site, they are still publicly available.

First Troubleshooting Step

If your router is set to use default routing via VPN tunnel and the WAN-targeting policies do not work, you need to stop your VPN tunnel first and ensure that you still have internet connection. If your router is set up to use the default routing via VPN tunnel and when you stop the VPN tunnel you have no internet connection, this package can't help you. You first need to make sure that you do have internet connection when the VPN tunnel is stopped.

Discussion

Please head to OpenWrt Forum for discussions of this service.

Thanks

I'd like to thank everyone who helped create, test and troubleshoot this service. Without contributions from @hnyman, @dibdot, @danrl, @tohojo, @cybrnook, @nidstigator, @AndreBL, @dz0ny, rigorous testing/bugreporting by @dziny, @bluenote73, @buckaroo, @Alexander-r, @n8v8R, psherman, @Vale-max, @ByteAndNibble, dscpl multiple contributions from @dl12345 and trendy and feedback from other OpenWrt users it wouldn't have been possible. Wireguard/IPv6 support is courtesy of Mullvad and IVPN.