WireGuard Selective Routing to External VPN Endpoint

Introduction

This how-to is designed to assist with setting up WireGuard on OPNsense to use selective routing to an external VPN peer - most commonly to an external VPN provider.

These circumstances may apply where only certain local hosts are intended to use the VPN tunnel. Or it could apply where multiple connections to the VPN provider are desired, with each connection intended to be used by different specific local hosts.

This how-to focuses on the configuration of OPNsense. You will also have to configure the peer at your VPN provider - consult your VPN provider's documentation as to how to do that.

Your OPNsense WireGuard Instance public key will need to be registered with your VPN provider, and you will need to get your VPN provider's endpoint public key and the VPN tunnel IP provided for your WireGuard Instance by your VPN provider. In some cases, you will not be able to get the Peer public key and VPN tunnel IP until you register your WireGuard Instance public key. In that case, create the OPNsense Instance configuration first, using a dummy tunnel IP and no peer selected, so that the public key is generated, and then update the configuration later once the other information is known.

For an example of configuring the peer at a VPN provider (Mullvad), see Step 1 of the how-to WireGuard MullvadVPN Road Warrior Setup.

This how-to primarily focuses on IPv4 configuration. It can be readily adapted for IPv6 as well. See Configuring IPv6 below.

Step 1 - Configure the peer

- Go to VPN ▸ WireGuard ▸ Peers
- Click + to add a new Peer
- Configure the Peer as follows (if an option is not mentioned below, leave it as the default):

Enabled	*Checked*
Name	*Call it whatever you want (eg* `VPNProviderName_Location` *)*
Public Key	*Insert the public key from your VPN provider*
Allowed IPs	*0.0.0.0/0*
Endpoint Address	*Insert the public IP address (desirably) or domain name of your VPN provider, as pr*
Endpoint Port	*Insert the port of your VPN provider, as provided by it*
Keepalive	*25*

- **Save** the Peer configuration, and then click **Save** again

Step 2 - Configure the WireGuard Instance

- Go to VPN ▸ WireGuard ▸ Instances
- Click **+** to add a new Instance configuration
- Turn on “advanced mode”
- Configure the Instance configuration as follows (if an option is not mentioned below, leave it as the default):

Enabled	*Checked*
Name	*Call it whatever you want (eg* `VPNProviderName` *)*
Public Key	*This will initially be blank; it will be populated once the configuration is saved*
Private Key	*This will initially be blank; it will be populated once the configuration is saved*
Listen Port	*51820 or a higher numbered unique port*
DNS Server	*Leave this blank, otherwise WireGuard will overwrite OPNsense’s DNS configuration*
Tunnel Address	*Insert the WireGuard Instance VPN tunnel IP provided by your VPN provider, in CIDR*
Peers	*In the dropdown, select the Peer you configured above*
Disable Routes	*Checked*
Gateway	*Specify an IP that is 1 number below your VPN tunnel IP, eg 10.24.24.9 - see note be*

❗ Note

The IP you choose for the Gateway is essentially arbitrary; pretty much any unique IP will do. The suggestion here is for convenience and to avoid conflicts

- **Save** the Instance configuration, and then click **Save** again

Step 3 - Turn on WireGuard

Turn on WireGuard under VPN ▸ WireGuard ▸ General if it is not already on

Step 4 - Assign an interface to WireGuard and enable it

- Go to Interfaces ▸ Assignments
- In the dropdown next to “New interface:”, select the WireGuard device (`wg0` if this is your first one)
- Add a description (eg `WAN_VPNProviderName`)
- Click **+** to add it, then click **Save**
- Then select your new interface under the Interfaces menu
- Configure it as follows (if an option is not mentioned below, leave it as the default):

Enable	*Checked*
Lock	*Checked if you wish to*
Description	*Same as under Assignments, if this box is not already populated*
IPv4 Configuration Type	*None*
IPv6 Configuration Type	*None*

- **Save** the interface configuration and then click **Apply changes**

Step 5 - Restart WireGuard

Now restart WireGuard - you can do this from the Dashboard (if you have the services widget) or by turning it off and on under VPN ▸ WireGuard ▸ General

Step 6 - Create a gateway

- Go to System ▸ Gateways ▸ Configuration
- Click **Add**
- Configure the gateway as follows (if an option is not mentioned below, leave it as the default):

Name	*Call it whatever you want, easiest to name it the same as the interface*
Description	*Add one if you wish to*
Interface	*Select your newly created interface in the dropdown*
Address Family	*Select IPv4 in the dropdown*
IP address	*Insert the gateway IP that you configured under the WireGuard Instan*
Far Gateway	*Checked*
Disable Gateway Monitoring	*Unchecked*
Monitor IP	*Insert the endpoint VPN tunnel IP (NOT the public IP) of your VPN pro*

Note

Specifying the endpoint VPN tunnel IP is preferable. As an alternative, you could include an external IP such as 1.1.1.1 or 8.8.8.8, but be aware that this IP will *only* be accessible through the VPN tunnel (OPNsense creates a static route for it), and therefore will not be accessible from local hosts that are not using the tunnel

Some VPN providers will include the VPN tunnel IP of the endpoint in the configuration data they provide. For others (such as Mullvad), you can get the IP by running a traceroute from a host that is using the tunnel - the first hop after OPNsense is the VPN provider's tunnel IP

- **Save** the gateway configuration and then click **Apply changes**

Step 7 - Create an Alias for the relevant local hosts that will access the tunnel

- Go to Firewall ▸ Aliases
- Click **+** to add a new Alias
- Configure the Alias as follows (if an option is not mentioned below, leave it as the default):

Enabled	*Checked*
Name	*Call it whatever your want, eg* `WG_VPN_Hosts`
Type	*Select either Host(s) or Network(s) in the dropdown, depending on whether you want spe*
Content	*Enter the host IPs, or the network in CIDR format*
Description	*Add one if you wish to*

- **Save** the Alias, and then click **Apply**

Step 8 - Create a firewall rule

The purpose of this step is to create a firewall rule to allow the relevant hosts to access the tunnel. At the same time, it also ensures that the relevant hosts using the tunnel can still access local resources as necessary - such as a local DNS server, or file storage

The step has two parts - first creating a second Alias for all local (private) networks, and then creating the firewall rule itself. The ultimate effect of these two steps is that only traffic from the relevant hosts that is destined for **non-local** destinations will be sent down the tunnel

Note

The rule below will mean that no local (private) IPs can be accessed over the tunnel. You may have a need however to access certain IPs or networks at the VPN endpoint, such as a DNS

server or monitor IP. In that case, you will need to create an additional firewall rule in OPNsense to ensure that requests to those IPs/networks use the tunnel gateway rather than the normal WAN gateway. This rule would be similar to that created below, except that the destination would be the relevant IPs/networks (or a new Alias for them) and the destination invert box would be unchecked. This rule would also need to be placed *above* the rule created below

❗ Warning

If the hosts that will use the tunnel are configured to use local DNS servers (such as OPNsense itself or another local DNS server), then the configuration below will likely result in DNS leaks - that is, DNS requests for the hosts will continue to be processed through the normal WAN gateway, rather than through the tunnel. See Dealing with DNS leaks for a discussion of potential solutions to this

- First go to Firewall ▸ Aliases
- Click + to add a new Alias
- Configure the Alias as follows (if an option is not mentioned below, leave it as the default):

Enabled	*Checked*
Name	*RFC1918_Networks*
Type	*Select Network(s) in the dropdown*
Content	*192.168.0.0/16 10.0.0.0/8 172.16.0.0/12*
Description	*All local (RFC1918) networks*

- **Save** the Alias, and then click **Apply**
- Then go to Firewall ▸ Rules ▸ [Name of interface for network in which hosts/network resides, eg LAN for LAN hosts]
- Click **Add** to add a new rule
- Configure the rule as follows (if an option is not mentioned below, leave it as the default):

Action	*Pass*
Quick	*Checked*
Interface	*Whatever interface you are configuring the rule on*
Direction	*in*
TCP/IP Version	*IPv4*
Protocol	*any*
Source / Invert	*Unchecked*
Source	*Select the relevant hosts Alias you created above in the dropdown (eg* `WG_VPN`
Destination / Invert	*Checked*
Destination	*Select the* `RFC1918_Networks` *Alias you created above in the dropdown*
Destination port range	*any*
Description	*Add one if you wish to*
Gateway	*Select the gateway you created above (eg* `WAN_VPNProviderName` *)*

- **Save** the rule, and then click **Apply Changes**
- Then make sure that the new rule is **above** any other rule on the interface that would otherwise interfere with its operation. For example, you want your new rule to be above the “Default allow LAN to any rule”

Step 9 - Configure routing

- Then go to Firewall ▸ Rules ▸ Floating
- Click **Add** to add a new rule
- Configure the rule as follows (if an option is not mentioned below, leave it as the default). You need to click the **Show/Hide** button next to “Advanced Options” to reveal the last setting:

Action	*Pass*
Quick	*Unchecked*
Interface	*Do not select any*
Direction	*out*
TCP/IP Version	*IPv4*
Protocol	*any*
Source / Invert	*Unchecked*
Source	*Select the interface address for your WireGuard VPN (eg* `WAN_VPNProviderName`
Destination / Invert	*Checked*
Destination	*Select the interface network for your WireGuard VPN (eg* `WAN_VPNProviderName`
Destination port range	*any*
Description	*Add one if you wish to*
Gateway	*Select the gateway you created above (eg* `WAN_VPNProviderName` *)*
allow options	*Checked*

- **Save** the rule, and then click **Apply Changes**

Step 10 - Create an outbound NAT rule

- Go to Firewall ▸ NAT ▸ Outbound
- Select “Hybrid outbound NAT rule generation” if it is not already selected, and click **Save** and then **Apply changes**
- Click **Add** to add a new rule
- Configure the rule as follows (if an option is not mentioned below, leave it as the default):

Interface	*Select the interface for your WireGuard VPN (eg* `WAN_VPNProviderName` *)*
TCP/IP Version	*IPv4*
Protocol	*any*
Source invert	*Unchecked*
Source address	*Select the Alias for the hosts/networks that are intended to use the tunnel (eg* `V`
Source port	*any*
Destination invert	*Unchecked*
Destination address	*any*
Destination port	*any*
Translation / target	*Interface address*
Description	*Add one if you wish to*

- **Save** the rule, and then click **Apply changes**

Step 11 - Add a kill switch (optional)

If the VPN tunnel gateway goes offline, then traffic intended for the VPN may go out the normal WAN gateway. There are a couple of ways to avoid this, one of which is outlined here:

- First, go back to the firewall rule you created under Step 7
- Click on the **Show/Hide** button next to “Advanced Options”
- Then, in the **Set local tag** field, add `NO_WAN_EGRESS`
- **Save** the rule, and then click **Apply changes**
- Then go to Firewall ▸ Rules ▸ Floating
- Click **Add** to add a new rule
- Configure the rule as follows (if an option is not mentioned below, leave it as the default). You need to click the **Show/Hide** button next to “Advanced Options” to reveal the last setting:

Action	*Block*
Quick	*Checked*
Interface	*WAN*
Direction	*out*
TCP/IP Version	*IPv4*
Protocol	*any*
Source / Invert	*Unchecked*
Source	*any*
Destination / Invert	*Unchecked*
Destination	*any*
Destination port range	*any*
Description	*Add one if you wish to*
Match local tag	*NO_WAN_EGRESS*

- **Save** the rule, and then click **Apply Changes**

Configuring IPv6

Some VPN providers (such as Mullvad) allow you to send both IPv4 and IPv6 traffic down the tunnel. This will be evident if you receive both an IPv4 and IPv6 tunnel IP in the configuration data provided by the VPN provider. The IPv6 tunnel IP is likely to be a ULA, ie within `fc00::/7`.

To configure the tunnel to use IPv6, you essentially need to replicate the steps above for IPv4. That is, you need to:

- add the IPv6 tunnel IP to Tunnel Address on the WireGuard Instance configuration (see further below)
- add `::/0` to the Allowed IPs on the WireGuard Endpoint configuration
- create an IPv6 gateway (see further below)
- add to the hosts alias the IPv6 addresses of the hosts/networks that are to use the tunnel
- if necessary, create a separate local IPs alias for IPv6, so they can be excluded from the IPv6 firewall rule destination
- create an IPv6 firewall rule (specifying the IPv6 gateway in the rule)
- configure an IPv6 floating rule for routing (specifying the IPv6 gateway in the rule)
- create an IPv6 outbound NAT rule
- (optionally) add the kill switch tag to the IPv6 firewall rule and change the associated Floating rule to IPv4+IPv6

Note, however, that there are a couple of differences:

1. First, the WireGuard Instance configuration will only accept one entry in the Gateway field. Just leave the IPv4 gateway address there.
2. Second, there is no concept of a Far Gateway for IPv6. So to successfully set up a gateway for IPv6, you need to do two things:

    - When adding the IPv6 address to Tunnel Address in the WireGuard Instance configuration, specify a /127 mask, rather than a /128
    - Then, when creating an IPv6 Gateway for the tunnel, specify the IP address to be another IPv6 address that is within the /127 subnet of the Tunnel Address

Dealing with DNS leaks

As noted in Step 8, if your network is configured to use a local DNS server - for example, unbound on OPNsense or on another local host - this how-to is likely to result in DNS requests from the hosts using the tunnel to be routed through the normal WAN gateway, rather than through the tunnel. This will result in the WAN IP being exposed.

If you wish to avoid that, there are several possible solutions. Obviously what solution works best will depend on your network configuration and desired outcomes.

The solutions include:

1. Force the local DNS server to use the tunnel as well. For a local DNS server that is not OPNsense, include the local IPs of that server in the Alias created in Step 7 for the relevant VPN hosts. For OPNsense itself, configure the DNS server to use the tunnel gateway. Implementing this solution will mean that all DNS traffic for your network will go through the tunnel, not just the DNS traffic for the hosts that are in the Alias (and, indeed, for a local DNS server that is not OPNsense, all traffic from that server, not just DNS traffic, will be forced through the tunnel). This may not be desirable for your circumstances
2. If possible, intercept DNS traffic coming from the relevant hosts using the tunnel, and forward that traffic (by using a port forward rule in OPNsense) to a DNS server supplied by your VPN provider (see note below), or to a public DNS server. Note that this will break local DNS resolution. Note also that this will not always be possible to do - if the local DNS server that is configured generally for your network is not OPNsense itself and is on the same subnet as the hosts using the tunnel, then DNS requests will not be routed through OPNsense and so a port forward on OPNsense will not work
3. Assuming you have configured DHCP static mappings in OPNsense for the hosts using the tunnel, specify in that configuration either the DNS servers supplied by your VPN provider (see note below), or public DNS servers. This will override the network-wide DNS settings for those hosts
4. Configure public DNS servers for your whole local network, rather than local DNS servers
5. Manually override the DNS settings on the relevant hosts themselves (assuming that is possible) so that the DNS servers provided by DHCP are ignored, and either the DNS servers supplied by

your VPN provider (see note below), or public DNS servers, are used instead

Note

If the DNS servers supplied by your VPN provider are local IPs (ie, within the scope of the `RFC1918_Networks` Alias created in Step 8), then, as discussed in Step 8, you will need to create an additional firewall rule in OPNsense to ensure that requests to those servers use the tunnel gateway rather than the normal WAN gateway. This rule would be similar to that created in Step 8, except that the destination would be your VPN provider's DNS server IPs and the destination invert box would be unchecked. This rule would also need to be placed *above* the rule created in Step 8