docs.opnsense.org

WireGuard Road Warrior Setup — OPNsense documentation

13–16 minutes

OPNsense

Introduction¶

WireGuard is a simple, fast VPN protocol using modern cryptography. It aims to be faster and less complex than IPsec whilst also being a considerably more performant alternative to OpenVPN. Initially released for the Linux kernel, it is now cross-platform and widely deployable.

This how-to describes setting up a central WireGuard server on OPNsense and configuring one or more clients to create a tunnel to it.

Step 1 - Install the WireGuard plugin¶

- Install the plugin via , selecting **os-wireguard**.
- Once the plugin is installed, refresh the browser page and you will find the WireGuard configuration menu via .

Step 2 - Configure the local peer (server)¶

- Go to
- Click **+** to add a new Local configuration

- Configure the Local configuration as follows (if an option is not mentioned below, leave it as the default):

Enabled	*Checked*
Name	*Call it whatever you want (eg* `HomeWireGuard` *)*
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*
Tunnel Address	*For example, 10.10.10.1/24. See note below*
Peers	*The (client) peers will be specified here; leave it blank initially until the Endpoint configuration is created in Step 3*
Disable Routes	*Unchecked*

Note

The tunnel address must be in CIDR notation and must be a unique IP and subnet for your network, such as if it was on a physically different routed interface. The subnet should be an appropriate size that includes all the client peers that will use the tunnel. For IPv4 it should be a private (RFC1918) address, for example 10.10.10.1/24. For IPv6, it could either be a unique

ULA /64 address, or a unique GUA /64 address derived from your prefix delegation. **Do not use a tunnel address that is a /32 (IPv4) or a /128 (IPv6)**

Note

Leave the DNS Server field (which appears if `advanced mode` is selected) blank. Otherwise WireGuard will overwrite OPNsense's DNS configuration

- **Save** the Local peer configuration, and then click **Save** again
- Re-open the Local peer configuration
- Copy the public key that has been generated in the configuration. This will be needed for the client device - see Step 7
- **Save** or **Cancel** to exit the configuration

Step 3 - Configure the endpoint (client peer)¶

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

Enabled	*Checked*
Name	*Call it whatever you want (eg* `Phone` *)*
Public Key	*Insert the public key from the client; if needed skip ahead and start Step 7 to generate the client public key*

Allowed IPs	*Unique tunnel IP address (IPv4 and/or IPv6) of client - it should be a /32 or /128 (as applicable) within the subnet configured on the local peer. For example, 10.10.10.2/32*

- **Save** the Endpoint configuration, and then click **Save** again
- Now go back to
- Open the Local configuration that was created in Step 1 (eg `HomeWireGuard`)
- In the Peers dropdown, select the newly created Endpoint (eg `Phone`)
- **Save** the Local peer configuration again, and then click **Save** once more
- Repeat this Step 3 for as many clients as you wish to configure

Step 4 - Turn on/restart WireGuard¶

- Turn on WireGuard under if it is not already on (click **Apply** after checking the checkbox)
- Otherwise, restart WireGuard - you can do this by turning it off and on under (click **Apply** after both unchecking and checking the checkbox)

Step 5 - Assignments and routing¶

Note

The steps outlined in Steps 5(a) and 5(b) below may not be required at all in your circumstances. Strictly speaking, if you only intend for your clients to use the tunnel to access local IPs/subnets behind OPNsense, then neither step is actually

necessary. If you intend to use the WireGuard tunnel to also access IPs outside of the local network, for example the public internet, then at least one, and perhaps both, of the steps will be required. This is explained below

However, it is useful to complete Step 5(a) anyway, for the reasons explained in that step

Step 5(a) - Assign an interface to WireGuard (recommended)¶

Hint

This step is not strictly necessary in any circumstances for a road warrior setup. However, it is useful to implement, for several reasons:

First, it generates an alias for the tunnel subnet(s) that can be used in firewall rules. Otherwise you will need to define your own alias or at least manually specify the subnet(s)

Second, it automatically adds an IPv4 outbound NAT rule, which will allow the tunnel to access IPv4 IPs outside of the local network (if that is desired), without needing to manually add a rule

Finally, it allows separation of the firewall rules of each WireGuard instance (each wgX device). Otherwise they all need to be configured on the default WireGuard group that OPNsense creates. This is more an organisational aesthetic, rather than an issue of substance

- Go to
- In the dropdown next to “New interface:”, select the WireGuard device (`wg1` if this is your first one)
- Add a description (eg `HomeWireGuard`)

- 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*
Description	*Same as under Assignments, if this box is not already populated*
IPv4 Configuration Type	*None*
IPv6 Configuration Type	*None*

Note

There is no need to configure IPs on the interface. The tunnel address(es) specified in the Local configuration for your WireGuard server will be automatically assigned to the interface once WireGuard is restarted

- **Save** the interface configuration and then click **Apply changes**
- Restart WireGuard - you can do this by turning it off and on under (click **Apply** after both unchecking and checking the checkbox)

Tip

When assigning interfaces, gateways can be added to them. This is useful if balancing traffic across multiple tunnels is required or in more complex routing scenarios. To do this, go to

and add a new gateway. Choose the relevant WireGuard interface and set the Gateway to **dynamic**. These scenarios are otherwise beyond the scope of this how-to

Tip

If Unbound DNS is configured with all interfaces registered it requires a reload of Unbound DNS to get the new Wireguard interface added. This is necessary to get DNS working through the VPN tunnel.

Step 5(b) - Create an outbound NAT rule¶

Hint

This step is only necessary (if at all) to allow client peers to access IPs outside of the local IPs/subnets behind OPNsense - see the note under Step 5. If an interface has already been assigned under Step 5(a), then it is not necessary for IPv4 traffic, and is only necessary for IPv6 traffic if the tunnel uses IPv6 ULAs (IPv6 GUAs don't need NAT). So in many use cases this step can be skipped

- Go to
- 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	*WAN*
TCP/IP Version	*IPv4 or IPv6 (as applicable)*

Protocol	*any*
Source invert	*Unchecked*
Source address	*If you assigned an interface under Step 5(a), select the generated alias for the interface subnet(s) (eg* `HomeWireGuard net` *) - see note below if you didn't assign this interface*
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**
- Restart WireGuard - you can do this by turning it off and on under (click **Apply** after both unchecking and checking the checkbox)

  Hint

  If you didn't assign an interface as suggested in Step 5(a), then you will need to manually specify the source IPs/subnet(s) for

the tunnel (for example, 10.10.10.0/24). It's probably easiest to define an alias (via) for those IPs/subnet(s) and use that. If you have only one local WireGuard instance and only one WireGuard endpoint configured, you can use the default `WireGuard net`, although this is generally not recommended due to unexpected behaviour

Step 6 - Create firewall rules¶

This will involve two steps - first creating a firewall rule on the WAN interface to allow clients to connect to the OPNsense WireGuard server, and then creating a firewall rule to allow access by the clients to whatever IPs they are intended to have access to.

- Go to
- 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	*WAN*
Direction	*in*
TCP/IP Version	*IPv4 or IPv4+IPv6 (as desired, depending on how you want clients to connect to the server; note this is distinct from what type of traffic is allowed in the tunnel once established)*

Protocol	*UDP*
Source / **Invert**	*Unchecked*
Source	*any*
Destination / **Invert**	*Unchecked*
Destination	*WAN address*
Destination port range	*The WireGuard port specified in the Local configuration in Step 2*
Description	*Add one if you wish to*

- **Save** the rule, and then click **Apply Changes**
- Then go to - see note below if you didn't assign this interface
- 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 (eg* `HomeWireGuard` *) - see note below*
Direction	*in*

TCP/IP Version	*IPv4 or IPv4+IPv6 (as applicable)*
Protocol	*any*
Source / **Invert**	*Unchecked*
Source	*If you assigned an interface under Step 5(a), select the generated alias for the interface subnet(s) (eg* `HomeWireGuard net` *) - see note below if you didn't assign this interface*
Destination / **Invert**	*Unchecked*
Destination	*Specify the IPs that client peers should be able to access, eg "any" or specific IPs/subnets*
Destination port range	*any*
Description	*Add one if you wish to*

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

  Note

  If you didn't assign an interface as suggested in Step 5(a), then the second firewall rule outlined above will need to be configured on the automatically created `WireGuard` group that appears once the Local configuration is enabled and WireGuard is started. You will also need to manually specify the source IPs/subnet(s) for the tunnel. It's probably easiest to define an

alias (via) for those IPs/subnet(s) and use that. If you have only one local WireGuard instance and only one WireGuard endpoint configured, you can use the default `WireGuard net`, although this is generally not recommended due to unexpected behaviour

Step 7 - Configure the WireGuard client¶

Tip

Key generation can be performed on an appropriate device with WireGuard client tools installed. A one-liner for generating a matching private and public keypair is `wg genkey | tee private.key | wg pubkey > public.key`. Alternatively, WireGuard apps that can be used on some devices can automate key generation for you

Client configuration is largely beyond the scope of this how-to since there is such a wide array of possible targets (and corresponding configuration methods). An example client (and server) configuration is in the Appendix. The key pieces of information required to configure a client are described below:

[Interface]	
Address	*Refers to the IP(s) specified as Allowed IPs in the Endpoint configuration on OPNsense. For example, 10.10.10.2/32*
PublicKey	*Refers to the public key that (along with a private key) needs to be manually or automatically generated on the client. The public key must then be copied into the Endpoint configuration on OPNsense for the relevant client peer - see Step 3*

DNS	*Refers to the DNS servers that the client should use for the tunnel - see note below*
[Peer]	
PublicKey	*Refers to the public key that is generated on OPNsense. Copy the public key from the Local configuration on OPNsense - see Step 2*
Endpoint	*Refers to the public IP address or publicly resolvable domain name of your OPNsense host, and the port specified in the Local configuration on OPNsense*
AllowedIPs	*Refers to the traffic (by destination IPs/subnets) that is to be sent via the tunnel. For example, if all traffic on the client is to be sent through the tunnel, specify 0.0.0.0/0 (IPv4) and/or ::/0 (IPv6)*

Note

If the DNS server(s) specified are only accessible over the tunnel, or you want them to be accessed over the tunnel, make sure they are covered by the AllowedIPs

Appendix - Example configurations¶

Warning

Do not re-use these example keys!

An example client configuration file:

[Interface]
PrivateKey = 8GboYh0YF3q/hJhoPFoL3HM
/ObgOuC8YI6UXWsgWL2M=

```
Address = 10.10.10.2/32, fd00:1234:abcd:ef09:10:2/128
DNS = 192.168.1.254, fd00:1234:abcd:ef09:1:254

[Peer]
PublicKey =
OwdegSTyhlpw7Dbpg8VSUBKXF9CxoQp2gAOdwgqtPVI=
AllowedIPs = 0.0.0.0/0, ::/0
Endpoint = opnsense.example.com:51820
```

An example server configuration file:

```
[Interface]
Address = 10.10.10.1/24, fd00:1234:abcd:ef09:10:1/64
ListenPort = 51820
PrivateKey =
YNqHwpcAmVj0lVzPSt3oUnL7cRPKB/geVxccs0C0kk0=

[Peer]
PublicKey =
CLnGaiAfyf6kTBJKh0M529MnlqfFqoWJ5K4IAJ2+X08=
AllowedIPs = 10.10.10.2/32, fd00:1234:abcd:ef09:10:2/128
```