OpenVPN traffic split

Contents

Introduction

OpenVPN is an open-source software application that implements virtual private network (VPN) techniques for creating secure point-to-point or site-to-site connections in routed or bridged configurations and remote access facilities.

This article provides a guide on how to configure an OpenVPN client on a RUTxxx router in such a way that a part of the router's LAN clients reach the Internet through the OpenVPN server (web proxy) and the other part reaches the Internet through the router's WAN (mobile, wired or WiFi connection). It should also be noted that this guide is aimed at more advanced users and, therefore, skips some of the more self-explanatory steps in order to preserve the overall coherence of the article. For example, the step-by-step configuration of the OpenVPN client instance will be left out and only comments on certain relevant parameters will be provided. If you feel this lack of information impedes your ability to configure the setup, we suggest you check out our **OpenVPN configuration examples** for reference and our guide on **generating TLS certificates**.

Configuration overview and prerequisites

Before we begin, let's overview the configuration that we are attempting to achieve and the prerequisites that make it possible.

Prerequisites:

- Working OpenVPN connection with OpenVPN server with zone allowing traffic to internet
- At least one end device (PC, Laptop) to configure the routers and test the set up

---

Topology scheme:

The figure above depicts the OpenVPN traffic split scheme. A RUTxxx router acts as an OpenVPN client (virtual IP: **172.16.0.2**) that is connected to a remote OpenVPN server (virtual IP **172.16.0.1**). The routers LAN/WiFi LAN IP addresses range from 192.168.1x.1 to 192.168.1x.254.

When the scheme is realized, Devices in the LAN range reach the Internet via the router's WAN and devices in the WiFi LAN range reach the Internet via the OpenVPN server effectively "adopting" the server's Public IP address.

Router configuration

Most of the router's configuration will be done via a command line interface. You can find detailed instruction on all command line interfaces supported by RUTxxx routers **here**. Choose one that is available or most preferred by you and you will still be able to follow the guide step-by-step regardless of which method you chose as the commands used will remain identical.

OpenVPN client

- First, you must create an OpenVPN client instance on your router. You can do this either via command line or from the router's WebUI, **Services → VPN → OpenVPN** section. We will not go into further detail on this because the client's configuration will depend on the OpenVPN server that you are connecting to. You can find detailed instructions on how to create and configure an OpenVPN client instance in our **OpenVPN configuration examples** article, which also contains information on how to configure an OpenVPN server on a RUTxxx router, if that is what you are using for this configuration.

- Once you have configured your OpenVPN client, you should probably test whether the OpenVPN connection is operational as this will make troubleshooting easier later on. The easiest way to do so is to login to the router's WebUI and check OpenVPN status in **Status → Services:**:

- If the connection was successful, we can start the traffic split configuration. First, we'll need to Edit LAN network to use IP address 192.168.10.1. It can be done in section **Network → Interfaces → General settings:** 

Wireless LAN interface

- Next, we'll need to create wireless interface to use a custom network (wifi_lan) and disable encryption for convenience. In order to do this, navigate **Network → Wireless** and click **edit:**

- In section **Network** select **Custom** and add your preferred interface name. In this example we use WIFI_LAN:

- Disable wifi encryption in **Wireless security** section, by choosing encryption type **No encryption.** Once you're finished, press **Save & Apply** and interface configuration windows will appears.
- In **general settings** edit wifi_lan interface to specify IPv4 address (e.g. 192.168.11.1). Press **Save & Apply.**

VPN interface

- Next, we'll need to create new interface (e.g named VPN). In **Physical settings** add a tunnel interface name as custom. Tunnel interface name can be checked via *ifconfig* command via SSH/CLI. In this case it is named "tun_c_Testas". Don’t forget to save configuration.

- When your done with the configuration run SSH client or CLI and connect to the router. Once connected execute these commands:

```
opkg update
```

- This command will update all opkg packages in router. Once update is finished install VPN policy routing using command:

```
opkg install vpn-policy-routing
```

- After successful installation time to configure VPN traffic splitting. In order to do so **one by one** execute the following uci commands (be aware that your configuration may vary):

```
uci set vpn-policy-routing.config.enabled="1"
while uci -q delete vpn-policy-routing.@policy[0]; do :; done
uci add vpn-policy-routing policy
uci set vpn-policy-routing.@policy[-1].dest_addr="192.168.10.0/24
192.168.11.0/24"
uci set vpn-policy-routing.@policy[-1].interface="ignore"
uci add vpn-policy-routing policy
uci set vpn-policy-routing.@policy[-1].src_addr="192.168.11.0/24"
uci set vpn-policy-routing.@policy[-1].interface="VPN"
uci commit
```

- When your done with the configuration, restart VPN policy routing service using:

```
/etc/init.d/vpn-policy-routing restart
```

Testing the setup

If you've followed the steps presented above, your configuration should be finished. But as with any other configuration, it is always wise to test the setup in order to make sure that it works properly.

In order to test this particular configuration, check whether devices with IPs from different interfaces reach the Internet through the default gateway. According to our configuration, if a device are connected to WAN interface, its Public IP should be that of the router's or SIM; if the device connected to WiFi LAN, its Public IP should be that of the OpenVPN server. You can check the Public IP address in **this website.**

If all of the above is in order, congratulations, your configuration works!

See also

- Other OpenVPN related articles from our wiki:
  - How to generate TLS certificates (Windows)? - a guide on generating TLS certificates for Windows users
  - OpenVPN Manual section - OpenVPN section of the router's manual
  - OpenVPN configuration examples - basic OpenVPN configuration scenarios with detailed examples
  - OpenVPN client on Windows - an example describing how to configure an OpenVPN client on a Windows computer
  - OpenVPN server on Windows - an example describing how to configure an OpenVPN server on a Windows computer

External links

- http://www.whatsmyip.org/ - for checking your Public IP address