OpenVPN configuration examples

□

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 contains various OpenVPN configuration examples that require more in depth explanations in order to achieve a successful configuration. All of the examples given concern two or more RUT routers. For more basic explanations on the OpenVPN WebUI section, visit our **VPN**

manual page.

OpenVPN configuration type

Before configuring anything you should first know what type of OpenVPN connection suits your needs the best. The key things to be considered here are the type of connection (**TUN** (**tunnel**) or **TAP** (**bridged**)), the data transfer protocol (**User Datagram Protocol** (**UDP**) or **Transmission Control Protocol** (**TCP**)) and authentication type (**TLS** or **Static key**). Here is a short overview of the differences:

- Type
  - TUN (tunnel) - simulates a network layer device and it operates with layer 3 packets like IP packets. TUN is used for routing and connecting **multiple** clients to a single server.
  - TAP (bridged) - simulates a link layer device and it operates with layer 2 packets like Ethernet frames. TAP is used for creating a network bridge between **two** Ethernet segments in different locations.
- Protocol
  - UDP - is used by apps to deliver a faster stream of information by doing away with error-checking.
  - TCP - a suite of protocols used by devices to communicate over the Internet and most local networks. It provides apps a way to deliver (and receive) an ordered and error-checked stream of information packets over the network.
- Authentication
  - TLS - uses SSL/TLS + certificates for authentication and key exchange.
  - Static key - uses a pre-shared Static key. Can only be used between two peers.

Overviews on most of these types and variations are provided in this article. Concerning TCP vs UDP, we will be using UDP for all examples. Choosing between TCP and UDP doesn't affect the rest of the configuration, so you can still follow the given examples no matter which protocol you are using. Simply choose the one that suits your purposes.

TLS Authentication

This section provides a guide on how to configure a successful OpenVPN connection between an OpenVPN Client and Server, using the **TLS** Authentication method on RUTxxx routers.

Generating TLS certificates/keys

A connection that uses TLS requires multiple certificates and keys for authentication:

- OpenVPN server
  - The root certificate file (Certificate Authority)
  - Server certificate
  - Server key
  - Diffie Hellman Parameters
- OpenVPN client
  - The root certificate file (Certificate Authority)

- Client certificate
- Client key

Before you continue you'll to obtain the necessary certificates and keys. When you use a third party OpenVPN service, they should provide you with their certificates and even configuration files.

If you're creating your own server, you'll have to generate these files yourself. To get detailed instruction on how to generate TLS certificates and keys check out our article on the topic for **Windows TLS certificate generation**.

Configuration

Now we can start configuring OpenVPN Server and Client instances. For this example we will be creating a TUN (Tunnel) type connection that uses the UDP protocol for data transfer and TLS for Authentication. We will be using two RUT routers: **RUT1** (**Server**; LAN IP: **192.168.1.1**; WAN (Public static) IP: **193.186.223.42**) and **RUT2** (**Client**; LAN IP: **192.168.2.1**); that will be connected into virtual network (with the virtual address: **10.0.0.0**):

To sum up, just make sure the Server and the Clients use the same parameters (same authentication, same port, same protocol, etc.). Another important aspect is the **Virtual network IP address** (10.0.0.0 in this case). The Server and the connected Clients will be given IP address that belong to this network. If you're creating an exceptionally large network, you might want to change the **Virtual network netmask**.

From the Client side, make sure to enter the correct **Remote host/IP address** (193.186.223.42 in this case). This is the Server's Public IP address, not the virtual IP address.

Static key Authentication

This section provides a guide on how to configure a successful OpenVPN connection between an OpenVPN Client and Server, using the **Static key** Authentication method on RUT routers.

Creating a Static key

A Static key connection uses a pre-shared for authentication between a Server and one Client. You can generate a Static key within the router itself or with PC that uses a Linux-based OS.

Within the router

In order to generate a Static key within the router connect to the device via the **Command Line Interface** (**CLI**) or **SSH** (the default username is **root**, the password is your router's admin password, **admin01** by default). CLI can be found in the router's WebUI, under Services. To connect to the router via SSH, use Terminal app (type ssh **root@192.168.1.1**; replace 192.168.1.1 with your

router's LAN IP address) if you're using a Linux-based OS. Or download **PuTTY**, a free SSH and telnet client, if you're using Windows.

When you have connected to the router, relocate to the directory (for example, **cd /etc/easy-rsa/keys/**) where you want to store your Static key and use this command:

```
# openvpn --genkey --secret static.key
```

The newly generated Static key will appear in the same directory where you issued the command above.

Extracting the key

If you are using a Linux-based OS, extracting files from the router is simple. Just go to the directory on your PC where you want to relocate the files, right click anywhere and choose the **Open in Terminal** option. In the Terminal command line use the **Secure Copy** (**scp**) command to copy the files from the router. The full command should look something like this:

```
$ scp root@192.168.1.1:/etc/easy-rsa/keys/static.key ./
```

The **root@192.168.1.1:/etc/easy-rsa/keys/static.key** specifies the path to where the Static key is located (replace the IP address with your router's LAN IP); the **./** denotes that you want to copy the contents to the directory you are in at the moment.

If you are using Windows, you can copy files from the router using **WinSCP**, an Open source freeware SFTP, SCP and FTP client for Windows OS. Use the same login information with WinSCP as with CLI or SSH. Once you've connected to the router with WinSCP, copying the files should be simple enough: just relocate to directory where you generated the key, select the Static key file and drag it to directory on your PC where you would like to store it.

Please note: You must select **SCP** as File Protocol in WinSCP Session settings.

On a Linux PC

To generate a Static key on a Linux PC, go to the directory where you want the key to appear, right click anywhere in that directory and chose the option **Open in Terminal**. In the Terminal window execute this command:

```
$ openvpn --genkey --secret static.key
```

The newly generated key should then appear in the directory you were in.

Configuration

When you have a Static key, you can start configuring OpenVPN Server and Client instances. For this example we will be creating a TUN (Tunnel) type connection that uses the UDP protocol for data transfer and Static key for Authentication. We will be using two RUT routers: **RUT1** (**Server**; LAN

IP: **192.168.1.1**; WAN (Public static) IP: **193.186.223.42**) and **RUT2** (**Client**; LAN IP: **192.168.2.1**); the two routers will be connected via OpenVPN; the Server's Virtual IP address will be **10.0.0.1**; the Client's - **10.0.0.2**:

To sum up, just make sure the Server and the Clients use the same parameters (same authentication, same port, same protocol, etc.). Other important aspects are the **Local tunnel endpoint IP** and the **Remote tunnel endpoint IP**. Take note these two particular parameter values are reversed for the individual Client and the Server configurations since these values represent opposite things depending on the instance's perspective.

From the Client side, make sure to enter the correct **Remote host/IP address** (193.186.223.42 in this case). This is the Server's Public IP address, not the virtual IP address.

TAP (bridged) OpenVPN

This section provides a guide on how to configure a successful OpenVPN TAP (bridged) connection between an OpenVPN Client and Server on RUT routers.

Configuration

TAP is used for creating a network bridge between Ethernet segments in different locations. For this example we will be creating a TAP (bridged) type connection that uses the UDP protocol for data transfer and TLS for Authentication. We will be using two RUT routers: **RUT1** (**Server**; LAN IP: **192.168.1.1**; WAN (Public static) IP: **193.186.223.42**) and **RUT2** (**Client**; LAN IP: **192.168.1.2**); the two routers will be connected via OpenVPN.

To sum up, just make sure the Server and the Clients use the same parameters (same authentication, same port, same protocol, etc.). Since the OpenVPN interface that comes up is bridged with the LAN interface, make sure the routers are in the **same subnet** (192.168.1.0 in this case). While making sure of that, don't forget that the routers can't have the same IP address, just the same subnet (for example, if both routers have the LAN IP 192.168.1.1, the connection won't work; if one has, for example, 192.168.1.1 and the other 192.168.1.100, then the connection will work).

For this example we used TLS Authentication. If you want to use a different Authentication method, refer to the relevant section of this article. The authentication configuration will not be different because of the chosen OpenVPN type (TUN or TAP).

From the Client side, make sure to enter the correct **Remote host/IP address** (193.186.223.42 in this case). This is the Server's Public IP address, not the LAN IP address.

Testing an OpenVPN connection

The most important thing after configuration is making sure that the newly established connection works. You can check the status of an OpenVPN connection in the **Status → Network → OpenVPN**

page:

Another method of testing pinging the other instance's virtual IP address. You can send ping packets via CLI, SSH or from the **System → Administration → Diagnostics** section of the router's WebUI:

Ping the Server's virtual IP address from the Client or vice versa. If the ping packets are transmitted successfully, congratulations, you OpenVPN connection is working.

Additional configuration

This section will provide examples of some additional OpenVPN related configurations like how to reach another OpenVPN instance's private LAN or how to use an OpenVPN instance as a Proxy.

Reaching a device's LAN network

You may want your OpenVPN Clients to be able to reach devices that are in the Server device's private network (LAN) or vice versa. This section will provide directions on how to do that.

Server from Client

To reach another OpenVPN instance's LAN network, you have to have a **route** to that network with the **Virtual remote endpoint** as the **gateway. You can add** Static routes **via command line, but these routes are removed automatically when router reboots or when connection goes down even if only for a moment. To solve this, you add permanent static routes via the router's WebUI in the** Network → Routing → Static Routes **page. But this method is also not foolproof since it means that if an address ever changes, you would have to also modify the static route on all related devices.**

Another method of reaching the OpenVPN Server's private network from the Client is specifying the network in the OpenVPN Client's configuration. To do so, open the Client's configuration window and fill in these two fields:

As you can see, the two fields in question are **Remote network IP address** and **Remote network IP netmask**. The values placed in these fields specify the Server's LAN address and having them filled will automatically add the necessary route into the routing table when the OpenVPN connection goes up. However, if your OpenVPN Server has multiple Clients, you would need to do this for all of them. If that is the case, use this next method.

Even another method is pushing the necessary routes via the OpenVPN Server. This method is the most foolproof because it will generate a route to the Server's private network for all connecting Clients. Therefore, in case of configuration changes you would only have to edit one field in the Server's configuration instead of having to edit all of the Clients' configurations.

To accomplish this, go to OpenVPN Server's configuration window and locate the **Push option** field. Let's say that the Server's LAN IP address is 192.168.1.1. In this case use the line **route 192.168.1.0 255.255.255.0**

Modify the information so that it reflects your own configuration. Do not specify the gateway, because the command will not work. The correct gateway will be assigned automatically.

Clients from Server

Reaching OpenVPN Clients' private networks from the Server is a bit trickier than the opposite, because in order to do so the Server has to be aware of the different specific addresses and Common Names of specific Clients.

To accomplish this, we can use the **TLS Clients** function. TLS Clients is a way to more specifically differentiate Clients by their Common Name (CN) found in the client certificate file. It can be used to assign specific VPN addresses to specific Clients and bind them to their LAN addresses so that other devices in the Client’s LAN can be reached from the Server.

In other words, TLS Clients binds Common Names (found in Client certificates) to Clients' private networks. If the certificate hasn't been tampered with in any after generation, the Common name should be the same as the file name (without the file type extension). For example, a certificate called **client1.crt** will likely have the Common Name of **client1**. But just to be sure you can open the certificate and check:

Once you know the Common Names and LAN IP Addresses of your OpenVPN Clients, you can create TLS Clients instances for each of them:

In addition, with TLS Clients you can manually assign Virtual local and remote endpoint addresses for the Clients. But these fields are not mandatory and the addresses will be assigned automatically if they are left unchecked.

Client to Client

For Client to Client communication to work you have to do three things:

- Create unique TLS Clients instances for each of the Clients

- Push the necessary routes via the Push option field
- Enable Client to Client functionality in the Server's configuration

TLS Clients

First, configure TLS Clients. You can find the description on how to do that in the section before this one (**here**). This is necessary in the case of multiple Clients because the Server will not only be pushing the routes of other Clients but also the routes to the Clients' own networks to their routing tables. This would cause the Clients' routers to be unreachable until the OpenVPN connection is terminated.

TLS Clients solves this problem, because the configuration then "tells" the router not push certain routes to certain Clients. For example, if a router pushes the route **192.168.5.0 255.255.555.0** to Client whose LAN IP address is 192.168.5.1, that Client will not be able to reach its network. TLS Clients prevents this - if a Client, for example, has the LAN IP address of 192.168.5.1, he will not receive the *route 192.168.5.0 255.255.555.0*.

Push options

Next, configure the necessary push options. You will have to include all Clients' networks if you want them all to communicate with each other. For the sake of argument, lets say you have three Clients that belong to three distinct LAN networks:

- 192.168.5.0
- 192.168.6.0
- 192.168.7.0

To give them all the necessary routes, you would have to include these three push options:

```
route 192.168.5.0 255.255.255.0
route 192.168.6.0 255.255.255.0
route 192.168.7.0 255.255.255.0
```

The configuration should look something like this:

Enable Client to Client

The next and final step is to enable the Client to Client functionality. To do this, go to the OpenVPN server's configuration window and put a check mark at the **Client to client** option:

If you did so and followed all of the previous steps in section, your OpenVPN Clients should now be able to communicate with each other.

OpenVPN Proxy

OpenVPN Servers can be used as Proxies by OpenVPN Clients. This means that the client will be assigned the Public IP address of the OpenVPN server and will be seen as using that IP address when browsing the Internet, transferring data or doing any other online activities. This section provides direction on how to set up and OpenVPN Proxy on RUT routers.

Push options

The first thing that you have to do is configure Push options in the OpenVPN Server configuration that will change the Clients' default WAN route to OpenVPN and set the DNS server to the OpenVPN Server's LAN IP. To do so open the OpenVPN configuration window and add these options to the Push option field:

```
redirect-gateway def1
dhcp-option DNS 192.168.1.1
```

In this context 192.168.1.1 is the OpenVPN Server's LAN IP address. Replace this value with your own Server's LAN IP address.

Firewall Zone Forwarding

Next, go to the **Network → Firewall → Zone Forwarding section. Click the** Edit **button located next to the** vpn **rule and in the subsequent window add a check mark next to** wan **as such:**

This will redirect all WAN traffic through the OpenVPN tunnel.

To test this out, on device behind the OpenVPN Client go to **http://www.whatsmyip.org/**. If the website shows the Public IP address of the OpenVPN server, it means the Proxy works.

Remote configuration

If you don't have physical or local access in general to the router, there are a few options to configure OpenVPN instances remotely.

Remote HTTP

You can access your router's WebUI from remote locations by enabling the **Remote HTTP** option in the **System → Administration → Access Control**. This will only work, however, if you have a Public Static or Public Dynamic IP (not Public Shared; more on IP address types **here**.

You can also enable the SMS Utilities **web** rule. More on that **here**.

Note: before enabling any type of remote access it is highly recommended that you change the router's default admin password to minimize the risk of malicious remote connections. You can change your password in the **System → Administration → General** section.

Remote Configuration (SMS Utilities)

You can send OpenVPN configurations via **Remote Configuration** tool located in the **Services → SMS Utilities** section. This method allows you to configure OpenVPN (among other things) just as you would in the OpenVPN section and then send these configurations to another router via SMS. The configuration method is identical to regular OpenVPN configuration. Therefore, additional instructions will not be provided here, but you can find more information on the subject of Remote Configuration **here**.

UCI

Yet another method would be using the SMS Utilities **uci** rule. You can find information on the rule itself **SMS Utilities manual article** and more detailed information the UCI System in general **here**.

See also

- How to generate TLS certificates (Windows)?
- OpenVPN client on Windows
- OpenVPN client on Linux
- OpenVPN server on Windows
- OpenVPN traffic split
- Other types of VPNs suported by RUTxxx devices:
  - IPsec configuration examples
  - GRE Tunnel configuration examples
  - PPTP configuration examples
  - L2TP configuration examples

External links

https://github.com/OpenVPN/easy-rsa-old - Easy-RSA download

https://winscp.net/eng/download.php - WinSCP download

https://openvpn.net/index.php/open-source/documentation/howto.html - some additional information on OpenVPNs

http://www.whatsmyip.org/