OpenVPN

From MikroTik Wiki

Contents

Generic

Why to use OpenVPN ?

OpenVPN has been ported to various platforms, including Linux and Windows, and it's configuration is throughout likewise on each of these systems, so it makes it easier to support and maintain. Also, OpenVPN is one of the few VPN protocols that can make use of a proxy, which might be handy sometimes.

Download OpenVPN

Debian provides OpenVPN packages as part of the standard distribution, just install them by typing **apt-get install openvpn**.
For a server, you want additionally to install the **openssl** package.
For easy client access, you would want to install **network-manager**, **network-manager-openvpn** and **network-manager-gnome** or **network-manager-kde**. This is a nice gui for handling wired and wireless network connections, connections via openvpn and cisco vpn (vpnc) and ppp connections (like a regular or 3g modem for example).

RouterOS requires v3.x and you will need to install and enable the *ppp* package. There is one limitation to using OpenVPN on the RouterOS platform: currently only *tcp* is supported. *udp* will not work.

For **Windows** you probably also want the GUI, that allows you to choose and activate certain VPN configuration from a simple click in the systray. A complete package for installation of OpenVPN incl. OpenVPN GUI can be downloaded at http://www.openvpn.se/download.html .

Features

OpenVPN V2.1 Features of RouterOS V4.2

Supported

- TCP
- bridging (tap device)
- routing (tun device)
- certificates
- p2p mode (refer to OpenVPN V2.1 manual page)

Unsupported

- UDP
- LZO compression

Certificates

OpenVPN works with SSL certificates. You can either use http://cacert.org to issue these or use the easy-rsa scripts, that come with most OpenVPN distributions. On RouterOS, all you have to do is to upload them via ftp (ca certificate and router certificate and private key) and import them with */certificate import* .

Creating Certificates with CAcert.org

Make sure you have created an account at CAcert.org. Login to your CAcert.org account and define your domain (Domains > Add). Note: You will need access to a root, postmaster, webmaster or other authoritive e-mail account to do this.

Add template with following command replacing name with one you desire (note that common-name = Domain name):

```
/certificate add name=client1-template common-name=client
```

And set template parameters described below:

```
Certificate request file name: certificate-request.pem
set name of private key file. if such file does not exist, it will be
created later.

rsa key bits: 1024  [Default]
Set values that make up distnguished name of your
certificate. you can leave some of them empty. CA may reject your certificate
request if some of these values are incorrect or missing, so please check what
are the requirements of your CA.

Set two character coutry code.

country name: [NOT IMPORTANT]
enter full name of state or province.

state or province name: [NOT IMPORTANT]
enter locality (e.g. city) name

locality name: [NOT IMPORTANT]

enter name of the organization

organization name: [NOT IMPORTANT]
enter organizational unit name

organization unit name: [NOT IMPORTANT]

Set common name. for ssl web servers this must be the fully qualified domain
name (FQDN) of the server that will use this certificate (like
www.someverysecuresitename.com) . this is checked by browsers.

common name: ovpnserver.mydomain.com  [IMPORTANT]
enter email address

email address: [NOT IMPORTANT]

now you can set challenge password. it's use depends on your CA. it may be
used to revoke this certificate.

challenge password: [NOT IMPORTANT]

you can set unstructured address, if your CA accepts or requires it.

unstructured address: [NOT IMPORTANT]

now private rsa key will be generated. no other certificate operations are
possible while generating key. 4096 bit key takes about 30 seconds on Celeron
800 system to generate. you will receive log message when it is done. download
by ftp from this router both private key and certificate request files. after
you receive your certificate from CA, upload it and the private key that will
be made now to a router and use "/certificate import" command to install it.
```

In RouterOS, open a New Terminal window and create a certificate request with the following command:

```
/certificate create-certificate-request
```

You will be asked for:

```
template: client1-template

passphrase: ****  [IMPORTANT]
verify passphrase: ****   [IMPORTANT]
enter number of bits for RSA key. longer keys take more time to generate.
```

As you can see, the only important fields are the Passphrase and Common Name fields, everything else can be left empty or default. This howto assumes you used "server" as common name. If not, you will have to replace it also in the command for the vpn server! After a few seconds you will receive notification that the Certificate Request file was created:

```
echo: system,info,critical certificate request file certificate-request.pem and private key file privat
```

Copy the **certificate-request.pem** file to your desktop and open it with Wordpad, Textpad, or any other text editor (except Notepad). Now go back to your CAcert.org account, and create a new Server Certificate (Server Certificates > New). Copy the entire contents of the **certificate-request.pem** file and Paste them into the "Paste Your CSR(Certificate Signing Request) below..." box on the CAcert.org site. Submit the form and if all goes well, you should be presented with a "Below is your Server Certificate" page with a bunch of text. Copy/Paste this text into a text file using Wordpad/Textpad (or anything except Notepad), and save it as **certificate-response.pem**. Upload this file to the router, and import it.

Warning: Generated private keys will be in pkcs8 format, which is not supported in RouterOS. To import such keys, run: openssl rsa -in private-key.key -text and write key output to new file. Upload new file to RouterOS and import

To import the keys, do this from the terminal: (you can also do it from Winbox - System -> Certificates -> Import)

```
/certificate
import file=certificate-response.pem
import file=private-key.key
```

Without converting and importing the private key you will get the dreaded "Couldn't change OVPN Server - no certificate found (6)" error as soon as you choose the certificate in OVPN Server!

Once you have imported the private key, your certificate should get a "KR" written next to it (K: decrypted-private-key, R: RSA). Now you will be able to use this key for OVPN.

Creating Certificates with Easy-RSA

Easy-RSA is part of OpenVPN package at [[1] (http://openvpn.net/index.php/open-source/downloads.html)]. As of OpenVPN version 2.1 the usage is as follows:

Initialisation on Linux:

```
cd easy-rsa
less README
vi vars
source vars
./clean-all
```

Create CA (Certificate Authority, required to sign client and server certificates)

```
./pkitool --initca
```

Create Server Certificate

```
./pkitool --pass --server RB450
```

Convert Server private key to .pem format

```
openssl rsa -in keys/RB450.key -out keys/RB450.pem
```

Create Client Certificate

```
./pkitool --pass client1
```

Convert Client private key to .pem format

```
openssl rsa -in keys/client1.key -out keys/client1.pem
```

Usage

Referring to easy-rsa example above upload following files via sftp to RouterBoard

```
RB450.crt
RB450.pem
ca.crt
```

Do not upload your private ca.key !!! Now import the certificate and then its key:

```
/certificate
import file=RB450.crt
import file=RB450.pem
import file=ca.crt
```

To the clients upload

```
ca.crt
client1.crt
client1.pem
```

And import the keys:

```
/certificate
import file=client1.crt
import file=client1.pem
import file=ca.crt
```

Naming Linux/Windows vs. RouterOS

There are two interface types within OpenVPN, that are used.

- **tun**, RouterOS defines this as **ip**.
- **tap**, which is needed for bridge mode gateways. RouterOS defines this as **ethernet**.

A few comments

The configuration files here are fully layed out for Debian and Ubuntu. If you're using something else, you'll have to do your own research, what you need. Hope they'll give a guideline.

Some new Linux- distributions use OpenSSL 1.0 (like Fedora 13) which is incompatible with older versions and (currently) MikroTik, it won't recognize the certificates generated with that version. Use OpenSSL version 0.9.8 instead.

Server configuration

Seperate segment for VPN and destination network

RouterOS

The network configuration of your box:

```
/ip address add address=10.15.30.31/24 interface=ether1 comment=Lan
/ip address add address=189.64.0.2/24 interface=ether2 comment=Internet
/ip route add dst-address=10.0.0.0/8 gateway=10.15.30.5 comment=Wan
/ip route add gateway=189.64.0.1 comment=Internet
```

Lan and Wan are the internal networks, Internet is obviously the Internet.
Altough it was explained here I still was confused since I didn't expect Internet on ether2. If your router is already working and online, all you need is the first line (/ip address add address=10.15.30.31/24 interface=ether1 comment=Lan) and replace interface=ether1 with your Lan interface) If NAT/masquerading is needed, this will do the job:

```
/ip firewall nat add chain=srcnat  out-interface=ether2 action=masquerade
```

Define an IP pool:

```
 /ip pool add name=ovpn-pool ranges=10.15.32.34-10.15.32.38
```

This pool is used for the OpenVPN clients.

Define a profile:

```
/ppp profile
add change-tcp-mss=default comment="" local-address=10.15.32.33 \
name="your_profile" only-one=default remote-address=ovpn-pool \
use-compression=default use-encryption=required use-vj-compression=default
```

Add a vpn user:

```
/ppp secret
add caller-id="" comment="" disabled=no limit-bytes-in=0 \
limit-bytes-out=0 name="username" password="password" \
routes="" service=any
```

Some might want to set **service** to **ovpn** to allow connection by this username only to openvpn server, not pppoe or pptp.

OpenVPN server configuration:

```
/interface ovpn-server server
set auth=sha1,md5 certificate=router_cert \
cipher=blowfish128,aes128,aes192,aes256 default-profile=your_profile \
enabled=yes keepalive-timeout=disabled max-mtu=1500 mode=ip netmask=29 \
port=1194 require-client-certificate=no
```

Firewall

If you have a firewall defined, that denies access, you would want to allow access to OpenVPN:

```
/ip firewall filter
add action=accept chain=input comment="OpenVPN" disabled=no dst-port=1194 protocol=tcp
```

Default Route

I haven't figured out, how to redistribute the default route from the OpenVPN server, so you'll have to add it yourself on the client by specifying the **add-default-route** option (if you have a RouterOS client).
If you have a Linux or a Windows client, you can use the **route-up** directive. Place it on your OpenVPN configuration (client) file with a command in append, and OpenVPN will execute it when the default route comes up.
For example, if you want to add a static route for 192.168.0.0 (obviously this net are on the remote side) through your OpenVPN gateway (IP 10.15.30.31), you have to add for Linux:
route-up "route add -net 192.168.0.0 netmask 255.255.255.0 gw 10.15.30.31"
or, for Windows:
route-up "route add 192.168.0.0 mask 255.255.255.0 10.15.30.31"

Linux

/etc/network/interfaces:

```
iface eth0 inet static
       address 10.15.30.31
```

```
       netmask 255.255.255.0
       network 10.15.30.0
       broadcast 10.15.30.255
       up /sbin/route add -net 10.0.0.0 netmask 255.0.0.0 gw 10.15.30.5
#
iface eth1 inet static
       address 189.64.15.2
       netmask 255.255.255.0
       gateway 189.64.15.1
       up echo "1" > /proc/sys/net/ipv4/ip_forward
```

eth0 is the network, that we want to get access to. eth1 is our outside interface.

/etc/openvpn/gw.conf:

```
port 1194
proto tcp
dev tun
ca keys/ca.crt
cert keys/vpngate.crt
key keys/vpngate.key
dh keys/dh1024.pem
server 10.15.32.32 255.255.255.224
ifconfig-pool-persist ipp.txt
keepalive 10 120
cipher none
#comp-lzo
user nobody
group nogroup
persist-key
persist-tun
status /var/log/openvpn/vpngate-status.log
verb 3
```

If you want to push a route to the client, this can be added:

```
push "route 10.0.0.0 255.0.0.0 10.15.32.33"
```

For a default gw to the client, usually, this is added:

```
push "redirect-gateway"
```

With RouterOS, this has no effect, whatsover, so if you want to push the default route from the server, please add:

```
push "route 0.0.0.0 0.0.0.0 10.15.32.33"
```

And to tell the client, what DNS servers to use, this will do the job:

```
push "dhcp-option DNS 10.15.15.10"
push "dhcp-option DNS 10.15.30.10"
```

Bridge mode

RouterOS

Create the bridge

```
/interface bridge add name=vpn-bridge
/interface bridge port add interface=ether1 bridge=vpn-bridge
```

The network configuration of your box:

```
/ip address add address=10.15.30.31/24 interface=vpn-bridge comment=Lan
/ip address add address=189.64.0.2/24 interface=ether2 comment=Internet
/ip route add dst-address=10.0.0.0/8 gateway=10.15.30.5 comment=Wan
/ip route add gateway=189.64.0.1 comment=Internet
```

Lan and Wan are the internal networks, Internet is obviously the Internet.
If NAT/masquerading is needed, this will do the job:

```
/ip firewall nat add chain=srcnat  out-interface=ether2 action=masquerade
```

Define an IP pool:

```
 /ip pool add name=ovpn-pool ranges=10.15.30.32-10.15.30.40
```

This pool is used for the OpenVPN clients.

Define a profile:

```
/ppp profile
add change-tcp-mss=default comment="" bridge=vpn-bridge \
name="your_profile" only-one=default remote-address=ovpn-pool \
use-compression=default use-encryption=required use-vj-compression=default
```

Add a vpn user:

```
/ppp secret
add caller-id="" comment="" disabled=no limit-bytes-in=0 \
limit-bytes-out=0 name="username" password="password" \
routes="" service=any
```

OpenVPN server configuration:

```
/interface ovpn-server server
set auth=sha1,md5 certificate=router_cert \
cipher=blowfish128,aes128,aes192,aes256 default-profile=your_profile \
enabled=yes keepalive-timeout=disabled max-mtu=1500 mode=ethernet netmask=24 \
port=1194 require-client-certificate=no
```

Before using *require-client-certificate* option, CA and correct server/client certificate must be imported to both OpenVpn server and client.

OpenVPN server Instance

At the moment, it looks like, that even though we've specified the vpn-bridge in the profile, RouterOS does not honour that fact. So we need to add a OpenVPN server Instance ourselfes for each user and add it to the bridge. *(Not required after RC11).*

```
/interface ovpn-server add name=ovpn-username user=username
/interface bridge port add interface=ovpn-username bridge=vpn-bridge
```

This will result in, that the dynamically created openvpn server instance automatically get's assigned to this interface and thus the bridge.

Firewall

If you have a firewall defined, that denies access, you would want to allow access to OpenVPN:

```
/ip firewall filter
add action=accept chain=input comment="OpenVPN" disabled=no dst-port=1194 protocol=tcp
```

Default Route

I haven't figured out, how to redistribute the default route from the OpenVPN server, so you'll have to add it yourself on the client by specifying the **add-default-route** option (if you have a RouterOS client).
If you have a Linux or a Windows client, you can use the **route-up** directive. Place it on your OpenVPN configuration (client) file with a command in append, and OpenVPN will execute it when the default route comes up.
For example, if you want to add a static route for 192.168.0.0 (obviously this net are on the remote side) through your OpenVPN gateway (IP 10.15.30.31), you have to add for Linux:
route-up "route add -net 192.168.0.0 netmask 255.255.255.0 gw 10.15.30.31"
or, for Windows:
route-up "route add 192.168.0.0 mask 255.255.255.0 10.15.30.31"

Linux

Packages

These packages are needed: **openvpn bridge-utils openssl**

Configuration

The configuration bits here are needed to set up a bridged gateway.

/etc/network/interfaces:

```
auto eth0 eth1 br0


# WAN interface
iface eth0 inet static
  address 10.15.30.31
  netmask 255.255.255.0
  network 10.15.30.0
  broadcast 10.15.30.255
  post-up route add -net 10.0.0.0/8 gw 10.15.30.5


# Internet interface
```

```
iface eth1 inet static
  address 189.64.15.2
  netmask 255.255.255.252
  gateway 189.64.15.1
  dns-nameservers 195.222.111.222 80.190.248.148 91.189.64.189
  pre-up echo 1 > /proc/sys/net/ipv4/ip_forward
  up /sbin/iptables -t nat -A POSTROUTING -o $IFACE -j MASQUERADE
  down /sbin/iptables -t nat -F
  post-down echo 0 > /proc/sys/net/ipv4/ip_forward


# OpenVPN interface
iface br0 inet manual
  up openvpn --mktun --dev tap0
  up ifconfig eth0 0.0.0.0 promisc up
  up ifconfig tap0 0.0.0.0 promisc up
  up brctl addbr br0
  up brctl setfd br0 0
  up brctl stp br0 off
  up brctl addif br0 eth0
  up brctl addif br0 tap0
  up ifconfig br0 10.15.30.31 netmask 255.255.255.0 up
  up route add -net 10.0.0.0/8 gw 10.15.30.5
  down ifconfig br0 down
  down brctl delif br0 tap0
  down brctl delif br0 eth0
  down brctl delbr br0
  down openvpn --rmtun --dev tap0
  down ifconfig eth0 10.15.30.31 netmask 255.255.255.0 broadcast 10.15.30.255 network 10.15.30.0
  down route add -net 10.0.0.0/8 gw 10.15.30.5
```

/etc/openvpn/bridge-gw.conf

```
port 1194
proto udp
dev tap0
ca keys/ca.crt
cert keys/bridge-gw.crt
key keys/bridge-gw.key
dh keys/dh1024.pem
ifconfig-pool-persist ipp.txt
server-bridge 10.15.30.31 255.255.255.0 10.15.30.100 10.15.30.119
keepalive 10 120
user nobody
group nogroup
persist-key
persist-tun
status openvpn-status-gw.log
verb 3
```

If you want to push a route to the client, this can be added:

```
push "route 10.0.0.0 255.0.0.0 10.15.30.5"
```

For a default gw to the client, usually, this is added:

```
push "redirect-gateway"
```

With RouterOS, this has no effect, whatsover, so if you want to push the default route from the server, please add:

```
push "route 0.0.0.0 0.0.0.0 10.15.32.33"
```

And to tell the client, what DNS servers to use, this will do the job:

```
push "dhcp-option DNS 10.15.15.10"
push "dhcp-option DNS 10.15.30.10"
```

Client configuration

RouterOS

client of a routed server (tun)

```
/interface ovpn-client \
  add name="ovpn-out1" connect-to=189.64.0.1 port=1194 mode=ip user="username" password="password" prof
  certificate=vpngate-client cipher=aes256 add-default-route=no
```

client of a bridged server (tap)

```
/interface ovpn-client \
  add name="ovpn-out1" connect-to=189.64.0.1 port=1194 mode=ethernet user="username" password="password
  certificate=vpngate-client cipher=aes256 add-default-route=no
```

Linux

client of a routed server (tun)

```
dev tun
proto tcp-client

remote openvpn.example.com 1194 # Remote OpenVPN Servername or IP address

ca   keys/ca.crt
cert keys/client.crt
key  keys/client.key

tls-client
port 1194

user nobody
group nogroup

#comp-lzo # Do not use compression. It doesn't work with RouterOS (at least up to RouterOS 3.0rc9)

# More reliable detection when a system loses its connection.
ping 15
ping-restart 45
ping-timer-rem
persist-tun
persist-key

# Silence  the output of replay warnings, which are a common false
# alarm on WiFi networks.  This option preserves the  security  of
# the replay protection code without the verbosity associated with
# warnings about duplicate packets.
mute-replay-warnings

# Verbosity level.
# 0 = quiet, 1 = mostly quiet, 3 = medium output, 9 = verbose
verb 3

cipher AES-256-CBC
auth SHA1
pull

auth-user-pass auth.cfg
```

The file auth.cfg holds your username/password combination. On the first line must be the username and on the second line your password.

```
username
password
```

client of a bridged server (tap)

Please replace *dev tun* with *dev tap*. Otherwise the configuration on the bridged client is exactly the same as the routed client.

Windows

client of a bridged server (tap)

```
proto tcp-client

remote openvpn.example.com 1194 # Remote OpenVPN Servername or IP address
dev tap

nobind
persist-key

tls-client
ca ca.crt # Root certificate in the same directory as this configuration file.
cert keys/client.crt
key  keys/client.key
```

```
ping 10
verb 3

cipher AES-256-CBC
auth SHA1
pull

auth-user-pass auth.cfg
```

The file auth.cfg holds your username/password combination. On the first line must be the username and on the second line your password.

```
username
password
```

Alternatively, if you don't specify the filename the client will prompt for the details.

Additional tweaks

Date & Time (certificate validity) - IMPORTANT

Don't forget to correctly set time and date on client and server. If possible, i suggest to use ntp to keep time in sync.

If time and date is not set correctly, certificates can be invalid. On RouterOS, openvpn will not connect, and you get just following lines in log, which are not very helpfull:

server (RouterOS)

```
... to be added ...
```

client (RouterOS)

```
... to be added ...
```


Category: VPN