



[Translation\(s\)](#): [English](#) - [Français](#) - [Русский](#) - [Polski](#)

Inhaltsverzeichnis

1. [OpenVPN Overview](#)
2. [Installation](#)
3. [Configuration](#)
 1. [Test VPN](#)
 1. [server test](#)
 2. [client test](#)
 2. [Static-Key VPN](#)
 3. [TLS-enabled VPN](#)
 4. [Debian Server with Android / iOS devices](#)
4. [Forward traffic via VPN](#)
5. [Auto-start](#)
6. [Application to a VPN passing through a http proxy](#)
 1. [TODO](#)
7. [See also](#)

OpenVPN Overview

OpenVPN is an SSL/TLS VPN solution. It is able to traverse NAT connections and firewalls. This page explain briefly how to configure a VPN with OpenVPN, from both server-side and client-side.

Installation

Install the *openvpn* package on both client and server.

```
# apt-get install openvpn
```

To enable OpenVPN in the Gnome [NetworkManager](#), the additional package *network-manager-openvpn-gnome* has to be installed:

```
# apt-get install network-manager-openvpn-gnome
```

Configuration

OpenVPN can authenticate users via user/pass, pre-shared key, certificates, etc.

Test VPN

Test a raw connection.

server test

From a server shell, run

```
# openvpn --remote CLIENT_IP --dev tun1 --ifconfig 10.9.8.1 10.9.8.2
```

if your client has a static IP#, otherwise, run

```
# openvpn --dev tun1 --ifconfig 10.9.8.1 10.9.8.2
```

You should see console output resembling

```
Wed Mar  7 06:03:03 2012 NOTE: OpenVPN 2.1 requires '--script-security 2' or higher to ca
Wed Mar  7 06:03:03 2012 ***** WARNING *****: all encryption and authentication featu
Wed Mar  7 06:03:03 2012 TUN/TAP device tun1 opened
...
```

While `openvpn` is running, check your network configuration with `sudo ifconfig -a`. Output should include

```
tun1      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:10.9.8.1  P-t-P:10.9.8.2  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:13 errors:0 dropped:0 overruns:0 frame:0
          TX packets:16 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:2262 (2.2 KiB)  TX bytes:1819 (1.7 KiB)
```

Note that, if you kill `openvpn` (e.g., with Control-c in its console), you will not see the above network interface.

client test

```
# openvpn --remote SERVER_IP --dev tun1 --ifconfig 10.9.8.2 10.9.8.1
...
Wed Mar  7 18:05:30 2012 Peer Connection Initiated with [AF_INET]SERVER_IP:PORT
Wed Mar  7 18:05:30 2012 Initialization Sequence Completed
...
```

You may also test with *ping*.

Static-Key VPN

In the server's `/etc/openvpn` directory, run the following command to generate a static key:

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

Copy this static key to the clients `/etc/openvpn` directory using a secure channel like `scp` or `sftp`.

On the server, create a new `/etc/openvpn/tun0.conf` file and add the following:

```
dev tun0
ifconfig 10.9.8.1 10.9.8.2
secret /etc/openvpn/static.key
```

Where 10.9.8.x is your VPN subnetwork, 10.9.8.1 will be IP of the server, 10.9.8.2 is IP of client.

On the client, copy `/etc/openvpn/static.key` from server and create a new `/etc/openvpn/tun0.conf` file and add the following:

```
remote your-server.org
dev tun0
ifconfig 10.9.8.2 10.9.8.1
secret /etc/openvpn/static.key
```

On the server's firewall, open up UDP 1194 (default port).

If you are using `shorewall`, on both devices, add a new VPN zone to represent `tun0` and create a default policy for it. This means adding something to the following files in `/etc/shorewall`:

- zone
- interfaces
- policy

Bear in mind that 90% of all connection problems encountered by new OpenVPN users are firewall-related.

Start OpenVPN by hand on both sides with the following command:

```
# openvpn --config /etc/openvpn/tun0.conf --verb 6 // verbose output.
```

You should probably configure your route at this step.

To verify that the VPN is running, you should be able to ping 10.9.8.2 from the server and 10.9.8.1 from the client.

TLS-enabled VPN

In **server**, copy key generating script from `openvpn` example to `/etc/openvpn` and add executable permission:

```
# cd /etc/openvpn
# mkdir easy-rsa
```

In Jessie and above `easy-rsa` is a separate package. So you'll have to install that in addition to `openvpn`.

On Wheezy:

```
# cp -R /usr/share/doc/openvpn/examples/easy-rsa/2.0/* easy-rsa/
```

On Jessie and above:

```
# apt-get install easy-rsa
# cp -R /usr/share/easy-rsa/* easy-rsa/
```

Edit `/etc/openvpn/easy-rsa/vars` bottom according to your organization.

```
export KEY_COUNTRY="US"
export KEY_PROVINCE="CA"
export KEY_CITY="SanFrancisco"
export KEY_ORG="Fort-Funston"
export KEY_EMAIL="mail@domain"
export KEY_EMAIL=mail@domain
```

Execute the following command:

```
# cd easy-rsa/
# mkdir keys
# touch keys/index.txt
# echo 01 > keys/serial
# . ./vars # set environment variables
# ./clean-all
```

Remember:

- only .key files should be kept confidential.
- .crt and .csr files can be sent over insecure channels such as plaintext email.
- do not need to copy a .key file between computers.
- each computer will have its own certificate/key pair.

Generate *CERTIFICATE AUTHORITY (CA) CERTIFICATE/KEY*:

```
# ./build-ca
```

It will generate `ca.crt` and `ca.key` in `/etc/openvpn/easy-rsa/keys/` directory.

Generate *BUILD AN INTERMEDIATE CERTIFICATE AUTHORITY CERTIFICATE/KEY* (optional):

```
# ./build-key-server server
```

It will generate *server.crt* and *server.key* in */etc/openvpn/easy-rsa/keys/*, and signed with your root certificate.

Generate *BUILD DIFFIE-HELLMAN PARAMETERS* (necessary for the server end of a SSL/TLS connection):

```
./build-dh
```

Generate key for each **client**:

```
./build-key clientname
```

Generate key with password (this protect the key and request the password every time that you connect to the server), for each **client**:

```
./build-key-pass clientname
```

It will generate keys in */etc/openvpn/easy-rsa/keys/*

Copy the *ca.crt*, *clientname.crt*, *clientname.key* from **Server** to **Client** */etc/openvpn/easy-rsa/keys/* directory.

Check [OpenVPN RSA Key](#) and [code.mixpanel.com VPN](#) for details.

Test the connectivity from command line.

Server:

```
openvpn --dev tun1 --ifconfig 10.9.8.1 10.9.8.2 --tls-server --dh /etc/openvpn/easy-rsa/keys/
```

Client:

```
openvpn --remote SERVER_IP --dev tun1 --ifconfig 10.9.8.2 10.9.8.1 --tls-client --ca /etc
```

If the connection is successful create file configuration.

In Server create */etc/openvpn/server.conf* as follows:

```
port 1194
proto udp
dev tun

ca /etc/openvpn/easy-rsa/keys/ca.crt # generated keys
cert /etc/openvpn/easy-rsa/keys/server.crt
key /etc/openvpn/easy-rsa/keys/server.key # keep secret
```

```
dh /etc/openvpn/easy-rsa/keys/dh2048.pem



server 10.9.8.0 255.255.255.0 # internal tun0 connection IP
ifconfig-pool-persist ipp.txt

keepalive 10 120

comp-lzo # Compression - must be turned on at both end
persist-key
persist-tun

status log/openvpn-status.log

verb 3 # verbose mode
client-to-client
```

Check  [code.mixpanel.com VPN](https://code.mixpanel.com) and  [rackspace OpenVPN](https://rackspace.com) for details.

Create log directory:

```
# cd /etc/openvpn
# mkdir -p log/
# touch log/openvpn-status.log
```

Restart OpenVPN.

```
# service openvpn restart
```

Note that the `/etc/init.d/openvpn` script will start an openvpn server for every `.conf` file in `/etc/openvpn/`, so if you still have the `tun0.conf` file from above, rename it to something else than `*.conf`. In the case of systemd only one openvpn server is started by default.

In Client create `/etc/openvpn/client.conf` as follows:

(note: you may use graphical vpn tool network-manager UI by providing the key and certificates)

```
client
dev tun
port 1194
proto udp

remote VPNSERVER_IP 1194 # VPN server IP : PORT
nobind

ca /etc/openvpn/easy-rsa/keys/ca.crt
cert /etc/openvpn/easy-rsa/keys/clientname.crt
```

```
key /etc/openvpn/easy-rsa/keys/clientname.key

comp-lzo
persist-key
persist-tun

verb 3
```

Restart OpenVPN:

```
# service openvpn restart
```

Debian Server with Android / iOS devices

OpenVPN can be configured to use with Android / iOS devices.

In Debian Server, create required certificates if you have a fresh installation of ?OpenVpn:

```
# cd /usr/share/doc/openvpn/examples/easy-rsa/2.0
# . ./vars
# ./clean-all
# ./build-ca
# ./build-key-server server
# ./build-key client
# ./build-dh
# cd keys
# mv *.pem *.crt *.csr *.key /etc/openvpn
# cd /usr/share/doc/openvpn/examples/sample-config-files
# gunzip -c server.conf.gz > /etc/openvpn/server.conf
```

Modify below lines in /etc/openvpn/server.conf:

```
...
proto tcp
push "redirect-gateway def1 bypass-dhcp"
push "dhcp-option DNS 8.8.8.8"
user nobody
group nogroup
...
```

8.8.8.8 is Google DNS server. You may change to your preferred DNS server.

When completed, restart OpenVPN server to use the new configuration:

```
# /etc/init.d/openvpn restart
```

Or on systems using systemd:

```
# service openvpn restart
```

Create client profile file `/etc/openvpn/client.ovpn` and attach certificates to it:

```
# cd /etc/openvpn
# cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf client.ovpn
# echo "key-direction 1" >> client.ovpn
# echo "<ca>" >> client.ovpn
# cat ca.crt | grep -A 100 "BEGIN CERTIFICATE" | grep -B 100 "END CERTIFICATE" >> client.ovpn
# echo "</ca>" >> client.ovpn
# echo "<cert>" >> client.ovpn
# cat client.crt | grep -A 100 "BEGIN CERTIFICATE" | grep -B 100 "END CERTIFICATE" >> client.ovpn
# echo "</cert>" >> client.ovpn
# echo "<key>" >> client.ovpn
# cat client.key | grep -A 100 "BEGIN PRIVATE KEY" | grep -B 100 "END PRIVATE KEY" >> client.ovpn
# echo "</key>" >> client.ovpn
```

Modify below lines in client profile file `/etc/openvpn/client.ovpn`:

```
...
proto tcp
remote YourServerIp YourServerPort
mute-replay-warnings
# ca ca.crt
# cert client.crt
# key client.key
key-direction 1
...
```

where `?YourServerIp` and `?YourServerPort` should be changed to your server. Three lines (`#ca`, `#cert`, `#key`) are remarked as the required certificates were attached to the profile file instead of individual files.

e-mail or upload the client configuration file `/etc/openvpn/client.ovpn` to google drive in order to download to iPhone.

For iOS devices, install [!\[\]\(d8ab143e904bfa3467271eec5af75a9b_img.jpg\) OpenVPN Connect](#) client. Then transfer the client configuration file `/etc/openvpn/client.ovpn` to the device by e-mail or by Google Drive. Open the configuration file in Mail apps or Google Drive apps.

For Android devices, install [!\[\]\(4688aadfd656ded00cd6bdfae55089a9_img.jpg\) OpenVPN Connect](#) client. Then copy the client configuration file `/etc/openvpn/client.ovpn` to the storage of the device. Open the configuration file in OpenVPN apps.

Forward traffic via VPN

In Server enable runtime IP forwarding:


```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Edit `/etc/sysctl.conf` uncomment the following line to make it permanent:

```
net.ipv4.ip_forward = 1
```

Execute the following command in server for testing:

```
iptables -A FORWARD -i eth0 -o tun0 -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -s 10.9.8.0/24 -o eth0 -j ACCEPT
iptables -t nat -A POSTROUTING -s 10.9.8.0/24 -o eth0 -j MASQUERADE
```

You may also use the `rc.firewall-iptables` script from  [TLDP Masquerade](https://tldp.org/HOWTO/html_pages/howto-256.html) as an alternative.

In client:

```
# ip route add VPNSERVER_IP via LOCALGATEWAY_IP dev eth0 proto static
# ip route change default via 10.9.8.5 dev tun0 proto static //client tun0 10.9.8.5
```

If you use graphical client generally you may not need to execute these command.

If everything is working fine, save the iptables rules:

```
# iptables-save > /etc/iptables.up.rules
```

To restore:

```
# iptables-restore < /etc/iptables.up.rules
```

add this to startup script. Debian wiki [iptables](https://wiki.debian.org/Iptables) page for details.

Auto-start

By default, all configured VPNs are started during system boot. Edit `/etc/default/openvpn` to start specific VPNs or to disable this behavior. Systemd users may need to run `systemctl daemon-reload` once to enable new VPNs.

openvpn [DebianPkg: ifupdown](#) hooks are also available for starting/stopping tunnels using `/etc/network/interfaces`, e.g.:

```
auto dsl
iface dsl inet ppp
    provider dsl-provider
    openvpn work_vpn
```

See `/usr/share/doc/openvpn/README.Debian.gz` for more information.

To automatically start a VPN located in `/etc/openvpn/client/` or `/etc/openvpn/server/`, enable `openvpn-client@<name>.service` or `openvpn-server@<name>.service`. For instance, a client configuration located in `/etc/openvpn/client/vpn0.conf` would be automatically started by enabling `openvpn-client@vpn0.service`.

Application to a VPN passing through a http proxy

This part describe how to configure a VPN to pass through a http proxy, which allow only traffic on port 443 (and 80). This use the `http_proxy` of OpenVPN.


1. First of all, check that the port 443 isn't already used by another service on your server.
2. Configure OpenVPN on server side by adding *port 443* and *proto tcp-server* to the configuration file.
3. Configure OpenVPN on the client side by adding *port 443*, *proto tcp-client* and *http-proxy 1.1.1.1 8080* to the configuration file.

Where 1.1.1.1 and 8080 are IP and port of your proxy.









1. Now you should launch OpenVPN on the server and next on the client.
2. At this time, you should configure routes to use the VPN tunnel:
 - Remove the default route through the proxy: *route del default eth0*
 - Add default route through your VPN: *route add default gw 10.9.8.1 dev tun0*
 - You should keep the route to the proxy with: *route add 1.1.1.1 eth0*

Update your `/etc/resolv.conf` according to your needs.

TODO

1. Explain how to enable the management interface ( <http://openvpn.net/index.php/open-source/documentation/miscellaneous/79-management-interface.html>)

See also

-  [OpenVPN home-page](#)
-  [code.mixpanel.com VPN](#)
-  [rackspace OpenVPN](#)
-  [openvpn pki how to](#)
-  [RSA key Management OpenVPN](#)
-  [OpnVPN Howto](#)
-  [Ubuntu OpenVPN](#)
-  [TLDP Masquerade](#)