How To Install OpenVPN On CentOS 7

By **Jijo**

Security is most important aspect in internet. Outsiders can monitor internet traffic between your computer and the web. Here the importance of **VPN** comes. VPN, or virtual private network, is a secure method of connecting remote internet resources together as if they were under the same LAN. **OpenVPN** is a popular open source application that implements a virtual private network. works on Linux, Windows, and Mac operating systems. It can be utilized to create a secure connection between physically distributed servers.

This Article explains How to install and configure OpenVPN in centOS 7 server.

Prerequisites

- CentOS 7 server.
- **root** access to the server.
- Domain or sub-domain that resolves to your server that you can use for the certificates

OpenVpn isn’t available in the default CentOS repositories. So we need to install Enterprise Linux (EPEL) repository. Use the following command to install EPEL repository.

```
yum install epel-release
```

Step 1 — Installing OpenVPN

First, We are going to install in the server by issuing the following command.

```
yum install openvpn -y
```

Step 2 — Install Easy RSA

For generating our SSL key pairs, which will secure our VPN connections. Execute the following command:

```
yum install easy-rsa -y
```

Step 3 — Configuring OpenVPN

We can find an example configuration file in its documentation directory**.** We need to copy the **sampleserver.conf** by the following command.

```
cp /usr/share/doc/openvpn-*/sample/sample-config-files/server.conf  /etc/openvpn
```

Open the file in your favorite editor, I'm using editor,

```
vi /etc/openvpn/server.conf
```

Most of the lines just need to be uncommented (remove the 😉 and some of there are to be changed.

Do the following changes.

We need to change the dh file name to **dh2048.pem**. Because the default Diffie-Hellman encryption length for Easy RSA will be 2048 bytes. We will do the key generation in next step.

```
dh dh2048.pem
```

Next, uncomment the **push "redirect-gateway def1 bypass-dhcp"** line, which tells the client to redirect all traffic through our OpenVPN.

```
push "redirect-gateway def1 bypass-dhcp"
```

Next we need to provide DNS servers to the client, as it will not be able to use the default DNS servers provided by your Internet service provider. We're going to use Google's public DNS servers, 8.8.8.8 and8.8.4.4.

For this, uncomment the **push “dhcp-option DNS** lines and updating the IP addresses.

```
push ″dhcp-option DNS 8.8.8.8″
push ″dhcp-option DNS 8.8.4.4″
```

Change user and group to nobody

```
user nobody
group nobody
```

Save and exit the OpenVPN server configuration file.

Step 4 — Generating Keys and Certificates

Now**,** we’ll need to generate our keys and certificates. Easy RSA installs some scripts to generate these keys and certificates.

Create a directory for the keys by the following command

```
mkdir -p /etc/openvpn/easy-rsa/keys
```

We also need to copy the key and certificate generation scripts into the directory.

```
cp -rf /usr/share/easy-rsa/2.0/* /etc/openvpn/easy-rsa
```

Now, we’re going to edit the default values in the script. So we don’t have to type our information in each time. Open the file in vi editor.

```
vi /etc/openvpn/easy-rsa/vars
```

Change values that start with **KEY_**. Update the following values to be accurate for your organization.

Some of the important value that should be change carefully are,

- KEY_NAME: You should enter server here; you could enter something else, but then you would also have to update the configuration files that reference and
- KEY_CN: Enter the domain or subdomain that resolves to your server

Refer the sample file below,

```
. . .
# These are the default values for fields
# which will be placed in the certificate.
# Don’t leave any of these fields blank.
export KEY_COUNTRY=″US″
export KEY_PROVINCE=″NY″
export KEY_CITY=″New York″
export KEY_ORG=″unixmen″
export KEY_EMAIL=″jijojamestj@gmail.com″
export KEY_OU=″Community″
# X509 Subject Field
export KEY_NAME=″server″
. . .
export KEY_CN=openvpn.unixmen.com
. . .
```

OpenSSL configuration may not load due to the version being undetectable. To avoid this remove the version number from the openSSl file name.

```
cp /etc/openvpn/easy-rsa/openssl-1.0.0.cnf /etc/openvpn/easy-rsa/openssl.cnf
```

Next, We are going to generate the keys and certificates. Move to easy-rsa directory and ***source*** in our new variables.

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

```
source ./vars
```

Then, we will clean up any keys and certificates which may already be in this folder and generate our certificate authority.

```
./clean-all
```

When you build the certificate authority, you will be asked to enter all the information we put into the vars file, but you will see that your options are already set as the defaults. So, you can just press ENTER for each one.

```
./build-ca
```

Next, We will generate the key and certificate for the server. Please press ENTER for each question as for the above step

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

Now we will generate Diffie-Hellman key exchange file. This command will take few to complete:

```
./build-dh
```

So, we completed the server keys and certificates generation process. Copy them all into our OpenVPN directory.

```
cd /etc/openvpn/easy-rsa/keys
```

```
cp dh2048.pem ca.crt server.crt server.key /etc/openvpn
```

For authenticate our clients will also need certificates. These keys and certificates will be shared with your clients, and it’s best to generate separate keys and certificates for each client you intend on connecting.

Make sure that if you do this you give them descriptive names, but for now we’re going to have one client so we’ll just call it client.

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

```
./build-key client
```

```
  That’s it for keys and certificates.
```

Step 5 — Routing

Install the iptables and disablefirewalld by execute the following commands

```
yum install iptables-services -y
```

```
systemctl mask firewalld
systemctl enable iptables
```

```
systemctl stop firewalld
```

```
systemctl start iptables
```

```
iptables --flush
```

Next, We need to add a rule to iptables to forward our routing to our OpenVPN subnet, and save this rule.

```
iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o eth0 -j MASQUERADE
```

```
iptables-save > /etc/sysconfig/iptables
```

Next, enable IP forwarding in sysctl. Open sysctl.conf in vi editor.

```
vi /etc/sysctl.conf
```

Add the following line at the top of the file:

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

For the IP forwarding will take effect. We need to restart the network service. Issue the following command

```
systemctl restart network.service
```

Step 6 — Starting OpenVPN

Now, we completed the installation and ready start the openVPN service. add it to systemctl using the command

```
systemctl -f enable  openvpn@server.service
```

Start OpenVPN:

```
systemctl start  openvpn@server.service
```

So we have successfully completed all the server-side configuration done for OpenVPN.

Next Let’s see how to connect a client to the server.

Step 6 — Configuring a Client

To connect you will definitely need a copy of the ca certificate from the server, along with the client key and certificate.

Locate the following files on the **server**. In this article we used ‘client’ as the descriptive name for the client keys.

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

```
/etc/openvpn/easy-rsa/keys/client.crt
```

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

Copy these three files to your client machine. For this, Open the file in the server and copy the content of the file into a new file in the client system an save, or use **SFTP**.

We’re going to create a file called client.ovpn. This is a configuration file for an OpenVPN client, telling it how to connect to the server.

- You’ll need to change the first line to reflect the name you gave the client in your key and certificate; in our case, this is just client
- You also need to update the IP address from your_server_ip to the IP address of your server; port1194 can stay the same
- Make sure the paths to your key and certificate files are correct

```
client
dev tun
proto udp
remote your_server_ip 1194
resolv-retry infinite
nobind
persist-key
persist-tun
comp-lzo
```

```
verb 3
ca /path/to/ca.crt
cert /path/to/client.crt
key /path/to/client.key
```

This file can now be used by any OpenVPN client to connect to your server.

Assume the client machine has Windows OS.

You will need the official OpenVPN Community Edition binaries which come with a GUI. Then, place your .ovpn configuration file into the proper directory, **ex. C:\Program Files\OpenVPN\config**, and click **Connect** in the GUI. OpenVPN GUI on Windows must be executed with administrative privileges.

That’s it!
Cheers..!