OpenVPN Server and Client Installation and Configuration on Debian 7

By Michael David Under: Open Source On: April 4, 2014

This article details how to obtain IPv6 connectivity on OpenVPN using Debian Linux. The process has been tested on Debian 7 on a KVM VPS with IPv6 connectivity as the server, and a Debian 7 desktop. The commands are to be run as root.

Install OpenVPN in Linux

Install OpenVPN in Debian

What is OpenVPN?

OpenVPN is a VPN program that uses SSL/TLS to create secure, encrypted VPN connections, to route your Internet traffic, thus preventing snooping. Open VPN is highly capable of transparently traversing through firewalls. In fact, if the situation requires it, you can run it on the same TCP port as HTTPS (443), making the traffic indistinguishable and thus virtually impossible to block.

OpenVPN can use a variety of methods such as pre-shared secret keys, certificates, or usernames/passwords, to let clients authenticate to the server. OpenVPN uses the OpenSSL protocol and implements many security and control features such as challenge response authentication, single sign-on capability, load balancing and failover features and multi daemon support.

Why use OpenVPN?

Think secure communications – think OpenVPN. If you do not want anyone snooping on your internet traffic, use OpenVPN to route all your traffic through a highly encrypted, secure tunnel.

This is especially important when connecting to public WIFI networks in airports and other places. You can never be sure as to who is snooping on your traffic. You can channel your traffic through your own OpenVPN server to prevent snooping.

If you are in any of the countries that routinely monitor all your traffic and block websites at will, you can use OpenVPN over TCP port 443, to make it indistinguishable from HTTPS traffic. You can even combine OpenVPN with other security strategies like tunnelling your OpenVPN traffic over an SSL tunnel, to beat Deep Packet Inspection techniques that might be able to identify OpenVPN signatures.

System Requirements

OpenVPN requires very minimal requirements to run. A system with 64 MB RAM and 1 GB HDD space is enough to run OpenVPN. OpenVPN runs on almost all the mainstream Operating Systems.

Installation and Configuration of OpenVPN on Debian 7

Install OpenVPN on Master Server

Run the following command to install OpenVPN.

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

By default, the easy-rsa scripts are installed under ‘/usr/share/easy-rsa/‘ directory. So, we need to copy these scripts to desired location i.e. /root/easy-rsa.

mkdir /root/easy-rsa
cp -prv /usr/share/doc/openvpn/examples/easy-rsa/2.0 /root/easy-rsa

Generate CA Certificate and CA Key

Open file ‘vars‘ and make the following changes, but before making changes I suggest you to take backup of original file.

cp vars{,.orig}

Using your text editor, set up the default values for easy-rsa. For example.

KEY_SIZE=4096
KEY_COUNTRY="IN"
KEY_PROVINCE="UP"
KEY_CITY="Noida"
KEY_ORG="Home"
KEY_EMAIL="user@example.net"

Here, I am using a 4096 bit key. You can use a 1024, 2048, 4096 or 8192 bit key as desired.

Export the default values by running the command.

source ./vars

Clean up any certificates that were generated previously.

./clean-all

Next, run the following command to generate CA certificate and CA key.

./build-ca

Generate the server certificate by running the command. Substitute the ‘server name’ with your server-name.

./build-key-server server-name

Generate the Diffie Hellman PEM certificate.

./build-dh

Generate the client certificate. Substitute the ‘client name’ with your client-name.

./build-key client-name

Generate the HMAC code.

openvpn --genkey --secret /root/easy-rsa/keys/ta.key

Copy the certificates to the client and server machines as follows.

Ensure that the ca.crt is present on both the client and the server.
The ca.key key should be on the client.
The server requires server.crt, dh4096.pem, server.key and ta.key.
client.crt, client.key and ta.key should be on the client.

To set up the keys and certificates on the server, run the commands.

mkdir -p /etc/openvpn/certs
cp -pv /root/easy-rsa/keys/{ca.{crt,key},server-name.{crt,key},ta.key,dh4096.pem}
/etc/openvpn/certs/

Configuring OpenVPN Server

Now you need to configure OpenVPN server. Open file '/etc/openvpn/server.conf'. Please make changes as described below.

script security 3 system
port 1194
proto udp
dev tap

ca /etc/openvpn/certs/ca.crt
cert /etc/openvpn/certs/server-name.crt
key /etc/openvpn/certs/server-name.key
dh /etc/openvpn/certs/dh4096.pem
tls-auth /etc/openvpn/certs/ta.key 0

server 192.168.88.0 255.255.255.0
ifconfig-pool-persist ipp.txt
push "redirect-gateway def1 bypass-dhcp"
push "dhcp-option DNS 8.8.8.8"
push "dhcp-option DNS 8.8.4.4"

keepalive 1800 4000

cipher DES-EDE3-CBC # Triple-DES
comp-lzo

max-clients 10

user nobody
group nogroup

persist-key
persist-tun

```
#log openvpn.log
#status openvpn-status.log
verb 5
mute 20
```

Enable IP forwarding on the server.

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

Run the following command to set up OpenVPN to start on boot.

```
# update-rc.d -f openvpn defaults
```

Start OpenVPN service.

```
# service openvpn restart
```

Install OpenVPN on Client

Run the following command to install OpenVPN on the client machine.

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

Using a text editor, setup the OpenVPN client configuration in ‘/etc/openvpn/client.conf‘, on the client. An example configuration is as follows:

```
script security 3 system
client
remote vpn_server_ip
ca /etc/openvpn/certs/ca.crt
cert /etc/openvpn/certs/client.crt
key /etc/openvpn/certs/client.key
cipher DES-EDE3-CBC
comp-lzo yes
dev tap
proto udp
tls-auth /etc/openvpn/certs/ta.key 1
nobind
auth-nocache
persist-key
persist-tun
user nobody
group nogroup
```

Run the following command to set up OpenVPN to start on boot.

```
# update-rc.d -f openvpn defaults
```

Start OpenVPN service on the client.

service openvpn restart

Once you are satisfied that OpenVPN is running well on IPv4, here is how to get IPv6 working over OpenVPN.
Getting IPv6 working with OpenVPN on Server

Add the following lines to the end of the server configuration ‘/etc/openvpn/server.conf‘ file.

client-connect /etc/openvpn/client-connect.sh
client-disconnect /etc/openvpn/client-disconnect.sh

These two scripts build/destroy the IPv6 tunnel each time a client connects/disconnects.

Here is the content of client-connect.sh.

```
#!/bin/bash
BASERANGE="2a00:dd80:003d:000c"
ifconfig $dev up
ifconfig $dev add ${BASERANGE}:1001::1/64
ip -6 neigh add proxy 2a00:dd80:003d:000c:1001::2 dev eth0
exit 0
```

My host assigns me IPV6 addresses from the 2a00:dd80:003d:000c::/64 block. Hence, I use 2a00:dd80:003d:000c as the BASERANGE. Modify this value as per what your host has assigned you.

Each time a client connects to OpenVPN, this script assigns the address 2a00:dd80:003d:000c:1001::1 as the IPV6 address of the tap0 interface of the server.

The last line sets up Neighbour Discovery for our tunnel. I have added the IPv6 address of the client side tap0 connection as the proxy address.

Here is the content of client-disconnect.sh.

```
#!/bin/bash
BASERANGE="2a00:dd80:003d:000c"
/sbin/ip -6 addr del ${BASERANGE}::1/64 dev $dev
exit 0
```

This just deletes the IPv6 tunnel address of the server, when the client disconnects. Modify the value of BASERANGE as appropriate.

Make the scripts executable.

chmod 700 /etc/openvpn/client-connect.sh
chmod 700 /etc/openvpn/client-disconnect.sh

Add the following entries to ‘/etc/rc.local’ (You can also modify the appropriate sysctls in

/etc/sysctl.conf).

```
echo 1 >/proc/sys/net/ipv6/conf/all/proxy_ndp
echo 1 > /proc/sys/net/ipv4/ip_forward
echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
/etc/init.d/firewall stop && /etc/init.d/firewall start
```

These entries activate Neighbor Discovery and Forwarding. I have also added a firewall.

Create ‘/etc/init.d/firewall‘ and put in the following content.

```
#!/bin/sh
# description: Firewall
IPT=/sbin/iptables
IPT6=/sbin/ip6tables
case "$1" in
start)
$IPT -F INPUT
$IPT -A INPUT -i eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPT -A INPUT -i eth0 -p tcp --dport 22 -j ACCEPT
$IPT -A INPUT -i eth0 -p icmp -j ACCEPT
$IPT -A INPUT -i eth0 -p udp --dport 1194 -j ACCEPT
$IPT -A INPUT -i tap+ -j ACCEPT
$IPT -A FORWARD -i tap+ -j ACCEPT
$IPT -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPT -t nat -F POSTROUTING
$IPT -t nat -A POSTROUTING -s 10.8.0.0/24 -o eth0 -j MASQUERADE
$IPT -A INPUT -i eth0 -j DROP
$IPT6 -F INPUT
$IPT6 -A INPUT -i eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPT6 -A INPUT -i eth0 -p tcp --dport 22 -j ACCEPT
$IPT6 -A INPUT -i eth0 -p icmpv6 -j ACCEPT
$IPT6 -A FORWARD -s 2a00:dd80:003d:000c::/64 -i tap0 -o eth0 -j ACCEPT
$IPT6 -A INPUT -i eth0 -j DROP
exit 0
;;
stop)
$IPT -F
$IPT6 -F
exit 0
;;
*)
echo "Usage: /etc/init.d/firewall {start|stop}"
exit 1
;;
esac
```

Run ‘/etc/rc.local‘ and start the firewall.

sh /etc/rc.local

This completes the server side modifications.
Getting IPv6 working with OpenVPN on Client

Add the following as the last lines of your client configuration file ‘/etc/openvpn/client.conf‘.

```
# create the ipv6 tunnel
up /etc/openvpn/up.sh
down /etc/openvpn/down.sh
# need this so when the client disconnects it tells the server
explicit-exit-notify
```

The up and down scripts build/destroy the IPV6 client end points of the client tap0 connection each time a client connects/disconnects to or from the OpenVPN server.

Here is the content of up.sh.

```
#!/bin/bash
IPV6BASE="2a00:dd80:3d:c"
ifconfig $dev up
ifconfig $dev add ${IPV6BASE}:1001::2/64
ip -6 route add default via ${IPV6BASE}:1001::1
exit 0
```

The script assigns the IPV6 address 2a00:dd80:3d:c:1001::2 as the client IPV6 address and sets the default IPV6 route through the server.

Modify IPV6BASE to be the same as BASERANGE in the server configuration.

Here is the content of down.sh.

```
#!/bin/bash
IPV6BASE="2a00:dd80:3d:c"
/sbin/ip -6 addr del ${IPV6BASE}::2/64 dev $dev
/sbin/ip link set dev $dev down
/sbin/ip route del ::/0 via ${IPV6BASE}::1
exit 0
```

This just deletes the IPV6 address of the client and tears down the IPV6 route when the client disconnects from the server.

Modify IPV6BASE to be the same as BASERANGE in the server configuration and make script executable.

```
# chmod 700 /etc/openvpn/up.sh
# chmod 700 /etc/openvpn/down.sh
```

Optionally, modify ‘/etc/resolv.conf’ and add Google’s IPV6 nameservers for DNS resolution.

nameserver 2001:4860:4860::8888
nameserver 2001:4860:4860::8844

Restart openvpn on the server and then connect to it from the client. You should be connected. Visit test-ipv6.com to see that your IPV6 connectivity over OpenVPN is working.
Reference Links

OpenVPN Homepage

Source: stavrovski