How to Setup PureVPN on Raspberry Pi

Finding it difficult to configure PureVPN on your Raspberry Pi? Simply, follow this guide and learn how you can set up the VPN on your device with a few clicks only:

Raspberry Pi (PPTP)

1 First of all you will have to install PPTP client that can be done using the following command:

sudo apt-get install pptp-linux

2 Next, Create a file in /etc/ppp/peers with any name of your choice and follow this configuration

pty “pptp $VPNHOSTNAME –nolaunchpppd –debug”
name $USERNAME
password $PASSWORD
remotename PPTP
require-mppe-128
require-mschap-v2
refuse-eap
refuse-pap
refuse-chap
refuse-mschap
noauth
debug
persist
maxfail 0
defaultroute
replacedefaultroute
usepeerdns

3 Without a delay, run the following command:

sudo pon $FILENAME

Congratulations! You can now use PureVPN on Raspberry Pi.

How to Share Logs?

In case, you find issues in running the setup then share your logs with us for further assistance. You can do that by running the following command:

pon $FILENAME debug dump logfd 2 nodetach

How to Run PureVPN automatically on Startup?

In case you want PureVPN to run PureVPN automatically on startup, you will have to create the following script in /etc/init.d/pptp.

```
#! /bin/sh

case “$1” in
  start)
 sleep 10
 pon $/etc/ppp/peers/FILENAME
    echo “PPTP Started”
    ;;
  stop)
    poff $/etc/ppp/peers/FILENAME
    echo “PPTP Stopped”
    ;;
  *)
    echo “Usage: /etc/init.d/pptp {start|stop}”
    exit 1
    ;;
 esac

exit 0
```

Now Run:

update-rc.d [filename of script] defaults

There you go! PureVPN is now automatically connected on startup.

Congratulations; you have setup PureVPN on your Raspberry Pi!

Raspberry Pi (OpenVPN)

Install the DNS forwarder

A DNS forwarder accepts DNS requests from clients and forwards them to real name servers, for example; 8.8.8.8 for Google. This, in turn, prevents DNS leaks when clients are utilizing the Pi as a router or gateway. We will use **dnsmasq** as our DNS forwarder for this project:

sudo apt-get install -y dnsmasq

Edit /etc/dnsmasq.conf:

sudonano /etc/dnsmasq.conf

Uncomment the ‘domain-needed’ and ‘bogus-priv’ settings:

Never forward plain names (without a dot or domain part)domain-needed# Never forward addresses in the non-routed address spaces.bogus-priv

Uncomment the ‘interface’ setting, set it to eth0:

If you want dnsmasq to listen for DHCP and DNS requests only on# specified interfaces (and the loopback) give the name of the# interface (eg eth0) here.# Repeat the line for more than one interface.interface=eth0

Save the file. To continue with the changes, restart the dnsmasq service:

sudo service dnsmasq restart

Install OpenVPN:

sudo apt-get install -y openvpn

Download the OpenVPN configuration files and extract them to pi user’s home folder/home/pi/openvpn

Download the files from here:

Copy the Certificate and WDC.key to /etc/openvpn:

sudo cp openvpn/WDC.key openvpn/ca.crt /etc/openvpn

Create the file /etc/openvpn/auth.txt. This file will contain your PureVPN login credentials:

sudo nano /etc/openvpn/auth.txt

Sample contents:

Purevpn0s123456
password

The first line is your user name, the second line is your password.

Make the file read/writeable only by root:

sudo chmod 600 /etc/openvpn/auth.txt

Create the file /etc/openvpn/server.conf:

sudo nano /etc/openvpn/server.conf

The file contents should be as follows:

client

dev tun

remote ukm1-ovpn.purevpn.net 53

proto udp

nobind

persist-key

persist-tun

tls-auth Wdc.key 1

ca ca.crt

cipher AES-256-CBC

comp-lzo

verb 1

mute 20

float

route-method exe

route-delay 2

auth-user-pass auth.txt

auth-retry interact

explicit-exit-notify 2

ifconfig-nowarn

auth-nocache

Change ownership of the openvpn configuration file:

sudo chown www-data:www-data /etc/openvpn/server.conf

Now restart the openvpn service:

sudo service openvpn restart

If you want to follow the connection process in the logs, open another ssh session and execute the following command:

sudo tail -f /var/log/syslog

Any openvpn connection errors will be shown.

Configure logrotate

Usually log files eat up significant space on a disk, to prevent it, we'll use **logrotate**. This service comes pre-installed in the Raspbian build. Edit the logrotate configuration file:

sudo nano /etc/logrotate.conf

change the rotate frequency to daily, and the backlogs to 4. Here is what the top section of the file should look like:

see “man logrotate” for details# rotate log files dailydaily# keep 4 days worth of backlogsrotate 4

Configure iptables

The firewall used by Raspbian is configured using iptables. It allows or blocks connections based on a number of rules. To allow the forwarding of http and dns connections, we will start with a working set of iptables rules. A script to load these rules is included. If you have not downloaded the VPN Client Gateway project files to your Pi, you should do so now:

wget download

Extract the files:

unzip master.zip

Run the firewall script to load the iptables rules. You’ll find the script in the folder vpn_client_gateway-master/fw:

vpn_client_gateway-master/fw/fw-config

To save these rules so that they reload at boot time, we’ll use the **iptables-persistent** utility.

Install iptables-persistent:

sudo apt-get install -y iptables-persistent

While the installation is being done, the program will ask you to save the current iptables rules (ipV4 and ipV6). You should acknowledge both prompts and the iptables rules will be saved and re-loaded at boot time.

Enable IP forwarding and disable IPV6

sudo nano /etc/sysctl.conf

Uncomment the following setting:

net.ipv4.ip_forward = 1

Add the following line:

net.ipv6.conf.all.disable_ipv6=1

To enable the changes, run the following command:

sudo sysctl -p /etc/sysctl.conf