

NordVPN setup on linux

maddingl : 5-6 minutes

I have written a little script that downloads the config files, renames them and enables automatic authentication. Insert your NordVPN login credentials in the generate authentication file part.

```
#!/bin/bash
# run as root!!!

# install openvpn. I'm running arch, this might be different on your system.
pacman -S openvpn

# go to openvpn config folder
cd /etc/openvpn

# download config files, extract and clean up
wget https://downloads.nordcdn.com/configs/archives/servers/ovpn.zip
unzip ovpn.zip
rm ovpn.zip

# rename tcp config files and put them in /etc/openvpn/client
cd ovpn_tcp
for file in *; do mv "${file}" "${file}/.nordvpn.com.tcp.ovpn/}tcp.conf"; done
cp * ../client

# rename udp config files and put them in /etc/openvpn/client
cd ../ovpn_udp
for file in *; do mv "${file}" "${file}/.nordvpn.com.udp.ovpn/}udp.conf"; done
cp * ../client

# generate authentication file
cd ../client
printf "<your email>\n<your password>" > auth.txt

# make all configs use authentication file
find . -name '*.conf' -exec sed -i -e 's/auth-user-pass/auth-user-pass\ auth.txt/g' {} \;

# clean up
cd ..
rm -r ovpn_tcp/
rm -r ovpn_udp
```

You can now start and stop vpn-connections via e.g.

```
systemctl start openvpn-client@de415tcp.service
```

and

```
systemctl stop openvpn-client@de415tcp.service
```

To automate this, and to connect to the server recommended by NordVPN, I have written two scripts. Make them executable and put them somewhere in your \$PATH. Pass a country code (like us, de or uk) as command line argument to start-vpn if you want to choose a specific country. It automatically chooses a tcp connection. You can change that to udp if you want.

start-vpn

```
#!/usr/bin/python
import sys

import requests
import os
import time

# you don't necessarily need the following. It's for monitoring via i3blocks.

def notify_i3blocks():
    os.system('pkill -RTMIN+12 i3blocks')

def fork_and_continue_notifying_in_background():
    newpid = os.fork()
    if newpid == 0: # if this is the child process
        for i in range(60):
            notify_i3blocks()
            time.sleep(1)

if __name__ == '__main__':

    notify_i3blocks()

    # below is what you do need.

    suffix = ''
    if len(sys.argv) > 1:
        countries = requests.get('https://nordvpn.com/wp-admin/admin-ajax.php?action=servers_countries').json()
        for country in countries:
            if country["code"].lower() == sys.argv[1].lower():
                suffix = '&filters={"country_id":"' + str(country["id"]) + '}'

    result = requests.get('https://nordvpn.com/wp-admin/admin-ajax.php?action=servers_recommendations' + suffix)
    profile = result.json()[0]['subdomain'] + 'tcp'

    command = 'systemctl start openvpn-client@' + profile + '.service'

    os.system(command)

    # the following is for i3blocks again.

    fork_and_continue_notifying_in_background()
```

stop-vpn

```
#!/bin/bash

function service {
    systemctl |
    grep openvpn |
    grep running |
    head -n1 |
    awk '{print $1;}'
}

while [[ $(service) ]]; do
    systemctl stop $(service)
done

# notify i3blocks
pkill -RTMIN+12 i3blocks
```

For convenience, I have two aliases in my ~/.bashrc:

```
alias start-vpn='sudo start-vpn'
alias stop-vpn='sudo stop-vpn'
```

if you do want to monitor it via i3blocks, put this in your i3blocks config:

```
[vpn]
interval=once
signal=12
```

and this in your i3blocks-scripts-directory (with name vpn):

```
#!/bin/bash

function name {
    systemctl |
    grep openvpn |
    grep running |
    head -n1 |
    awk '{print $1;}' |
    cut -d @ -f 2 |
    cut -d . -f 1
}

starting=$(pgrep -f start-vpn) # this might not be the most accurate, but it works
for me. Improvement suggestions are welcomed.

if [[ $(name) ]]; then
    echo $(name)
    echo && echo "#00FF00"
else
    if [[ ${starting} ]]; then
        echo starting vpn...
        echo && echo "#FFFF00"
    else
        echo no vpn
    fi
fi
```

```
        echo && echo "#FF0000"
    fi
fi
```

In order to automatically start and stop vpn when a network interface goes up/down, put the following in `/etc/NetworkManager/dispatcher.d/10-openvpn`. To activate the feature you need to enable and start the `NetworkManager-dispatcher.service`. More info [here](#).

At my university, I connect to eduroam, which does not allow vpn. That's why I exclude that.

`/etc/NetworkManager/dispatcher.d/10-openvpn`

```
#!/bin/bash

case "$2" in
    up)
        if ! nmcli -t connection | grep eduroam | grep wlp3s0 ; then
            start-vpn
        fi
        ;;
    down)
        stop-vpn
        ;;
esac
```

I hope this helps other people who want to use NordVPN on linux. Again, feel free to comment and suggest improvements. In particular, I am not sure how much of a security risk it is to have the NordVPN-password written out in plain text in a file.