

Saját szkript a Wireguard VPN-kapcsolatod kezelésére

PCLinuxOS Magazine - 2026. július

Írta: Cuig

Egyáltalán miért írok ilyen szkriptet?

Mivel szeretem egyszerű, nem erőszakolt módon intézni a dolgokat, és sok egyéb miatt más elfogadható opcióim nem volt.

A többi ok között van, hogy az általam választott Proton, VPN-szolgáltató csak a systemd-t futtató rendszerekhez adja az alkalmazásait. Nagyon rövid az általuk „támogatott” disztrók listája.

Egy szép grafikus felületű (GUI) alternatíva a wireguard (figyelj a betűzésre), de nálam problémás volt a telepítése, mivel egy VPN lekapcsolásakor internetkapcsolat nélkül hagy!

Szóval ezért van ez a szkript. De minek a VPN egyáltalán?

Miért is akarok VPN-t használni? Valójában több oka van, amik többségére a magazin más oldalain már kitértem, de leginkább a kormányom tevékenysége miatt.

- Sok külföldi oldal (például az rt.com) blokkolva van az országomban azért, hogy CSAK azt olvashassam, amit a kormányom engedélyez.
- Ez számomra semmilyen körülmények között sem elfogadható.
- Emiatt, és a VPN-adta kapott privát tér az, amiért ilyet használok.
- Időszakonként más földrajzi helyek (ingyenes) közszolgálati TV-it is nézem, ahol VPN-t kell használni.

A wireguard-protokollt használom, mivel az a legfrissebb, biztonságos és az általában elérhetőek közül a leggyorsabb. A szkriptben SUDO szerepel,

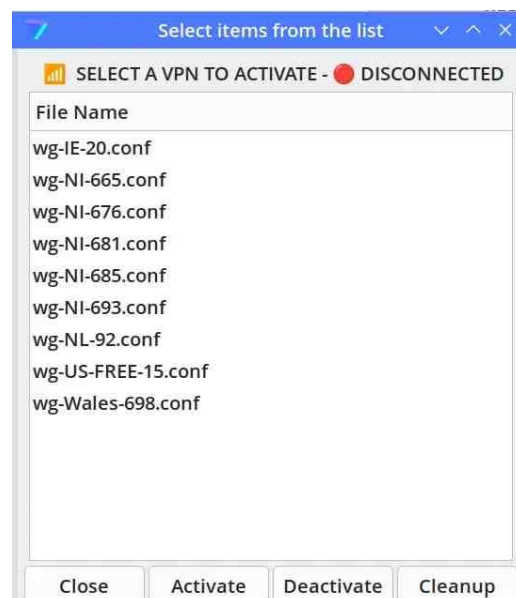
mert a wireguard.conf fájlok az /etc/wireguard-ban vannak, aminek a root a tulajdonosa, illetve pár parancshoz is root-jog kell. Ha wireguard fájlok engedélye rossz, akkor hibát dob – feltehetően azért, mert ez csökkentené a biztonságot.

A GUI-ban Zenity-t használtam, mivel Yad-ban (az első próbáltam) nem tudtam kitalálni hogyan kell megcsinálni azt, amit akartam.

Figyelj arra, hogy a szkript több elemének beállításához SUDO kell. További függőségei: Zenity, SUDO, wireguard-tools és esetemben ikonként a papirus-icon-theme.

A szkript

A szkript kiválasztja az elsődleges hálózati kapcsolatot és csak egyetlen aktív VPN-kapcsolatot engedélyez egy időben.



VPN-kapcsolat nélküli indításkor ezt mutatja (kép). Felsorolja a /etc/wireguard könyvtárban lévő „.conf” kiterjesztésű fájlokat. Ezek a fájlok, amiket a VPN-szolgáltatótól kapsz. A te fájljaid az itt bemutatottaktól biztos különböznek.

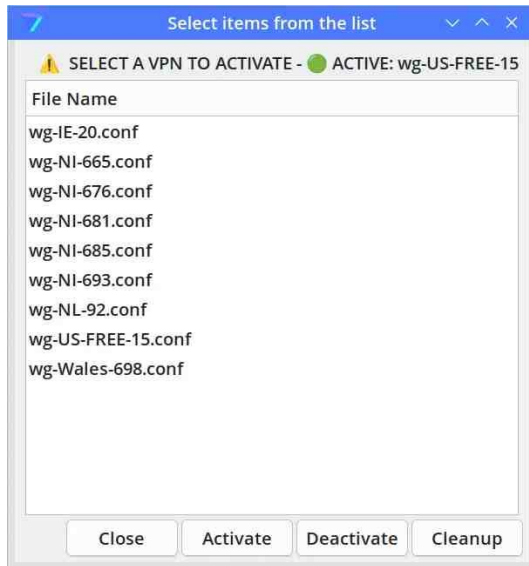
Ahogy az fent található “Disconnected”-ből (lekapcsolva) látható, nincs aktív VPN, és kéri, hogy válassz egyet az aktiváláshoz. Miután kiválasztottad, egyszerűen kattints az „Activate” gombra, és a VPN aktiválódik.

Saját szkript a Wireguard VPN-kapcsolatod kezelésére

Egy megjelenő ablak erősíti meg, hogy melyik VPN van aktiválva. Bezárásához válaszd az OK-t.



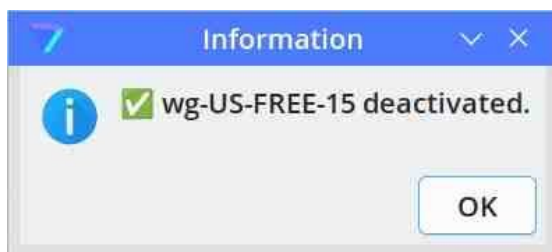
A szkript következő indításakor az aktív VPN a cím alatt jelenik meg.



Ebből az ablakból kapcsolhatod ki a VPN-t, visszatérve a normál IP-címre, vagy kiválaszthatsz egy másik VPN-fájlt, és az Aktiválás gombbal indíthatod.

Ha egy új fájlt választasz és „Activate”-re kattintasz, akkor a meglévő VPN-t lekapcsolja, és aktiválja az újonnan kiválasztottat, biztosítva, hogy bármikor csak egy VPN legyen aktív. Egy kis ablak tájékoztat a választott új VPN nevééről.

A Deactivate gomb kiválasztásával a következő ablak jelenik meg.



A fő ablakban a „Close” (bezár) gomb egyértelmű, de a „Cleanup” (tisztít) gombhoz magyarázat dukál. Ha valami félresikerül, mint PC-fagyás, vagy erőltetett kilépés, a „Cleanup” gomb helyre kell, hogy állítsa a normális helyzetet. Azért került bele, hogy megoldjon néhány hibát, vagy működési zavart. Alapvetően az alapértékekre állítja a kapcsolatodat. Akkor történhet, amikor az asztal más alkalmazások, vagy folyamat miatt lefagy, vagy elszáll és a VPN-hálózatod zavarodott helyzetben ragad. Ebben az esetben ez segíthet.

Proton VPN-t használok, így az Ő ikonjukat használom a szkriptben. Az ikon a szkript szerkesztésével lecserélhető – a beírást kell megváltoztatni.

Nem sok egyebet lehet még mondani. Aktiválja, kapcsolja és deaktiválja a VPN-kapcsolatot egyszerű felületen. Semmi extra, csak funkcionalitás.

A kényelem kedvéért mellékeltem egy asztali fájlt. A működő szkriptet \$HOME/bin/ProtonWG-be helyeztem, így ha máshová tennéd, akkor az asztali fájl elérési útját szerkeszteni kell.

Először itt van a.desktop fájl, my-vpn-manager.desktop:

```
#!/usr/bin/env xdg-open
[Desktop Entry]
Categories=Internet;Network;
Comment[en_GB]=VPN Manager
Comment=VPN Manager
Exec=$HOME/bin/ProtonWG/my-vpn-manager
GenericName[en_GB]=VPN Manager
GenericName=VPN Manager
Icon=/usr/share/icons/Papirus/48x48/apps/proton-vpn-logo.svg
MimeType=
Name[en_GB]=my-vpn-manager
Name=my-vpn-manager
Path=
StartupNotify=true
Terminal=false
TerminalOptions=
```

Saját szkript a Wireguard VPN-kapcsolatod kezelésére

```
Type=Application
X-KDE-SubstituteUID=false
X-KDE-Username=
```

A következő a működő szkript, **my-vpn-manager**.

```
#!/bin/bash
# =====
# VPN Manager Script v1.6 2026-04-13
# Features: Switching VPN, Stale VPN cleanup, File detection of ".conf"
# only
# OS: PCLinuxOS 2026
# Dependencies: Sudo, wireguard-tools, Zenity, papirus-icon-theme
# =====

set -x # For debug output in a terminal
cd /etc/wireguard
Icon=/usr/share/icons/Papirus/48x48/apps/proton-vpn-logo.svg

# --- Helper: Detect Active VPN ---
get_active_vpn() {
    # Must use sudo to read kernel state. We need the 2nd field.
    local iface=$(sudo wg show 2>/dev/null | awk 'NR==1 {print $2}')
    echo "$iface"
}

# --- Helper: Find Primary Network Interface ---
get_primary_iface() {
    # Try to find the interface with the default route
    local iface=$(ip route | grep default | awk '{print $5}' | head -n1)

    # Fallback: First non-loopback interface
    if [[ -z "$iface" ]]; then
        iface=$(ip link show 2>/dev/null | awk -F': ' '/^[0-9]+: (eth|wl|
en)/ {print $2}' | head -n1)
    fi
}

# Last resort: eth0
if [[ -z "$iface" ]]; then
```

```
    iface="eth0"
fi
echo "$iface"
}

# --- Helper: Cleanup ALL Stale Connections ---
do_cleanup() {
    echo ">>> Running Cleanup..."

    # Extract ONLY the interface names (first column of first line for each
    # block)
    # Use 'awk' to find lines starting with "interface:" and print the 2nd
    # field.
    local ifaces=$(sudo wg show 2>/dev/null | awk '/^interface:/ {print
$2}')

    if [[ -z "$ifaces" ]]; then
        zenity --window-icon=$Icon --info --text="Info No active connections
found."
        return
    fi

    local count=0
    for i in $ifaces; do
        echo "Stopping: $i"
        sudo wg-quick down "$i" 2>/dev/null || true
        ((count++))
    done

    local PRIMARY=$(get_primary_iface)
    echo "Resetting network interface: $PRIMARY"
    if [ -f /usr/libexec/nm-ifdown ] && [ -f /usr/libexec/nm-ifup ]; then
        sudo /usr/libexec/nm-ifdown "$PRIMARY"; sleep 1; sudo /usr/libexec/
nm-ifup "$PRIMARY"
    else
        sudo ip link set "$PRIMARY" down; sleep 1; sudo ip link set "$PRIMARY"
up
    fi
}
```

Saját szkript a Wireguard VPN-kapcsolatod kezelésére

```
zenity --window-icon=$Icon --info --text="Info Cleaned up $count --cancel-label="Close")
connection(s)."
sleep 1
}

# --- Helper: Hard Kill ---
do_hard_kill() {
    local iface=$1
    sudo wg-quick down "$iface"
    local PRIMARY=$(get_primary_iface)
    if [ -f /usr/libexec/nm-ifdown ] && [ -f /usr/libexec/nm-ifup ]; then
        sudo /usr/libexec/nm-ifdown "$PRIMARY"; sleep 1; sudo /usr/
libexec/nm-ifup "$PRIMARY"
    else
        sudo ip link set "$PRIMARY" down; sleep 1; sudo ip link set
"$PRIMARY" up
    fi
}

#
=====
# MAIN LOOP
#
=====
while true; do
    # 1. Check Status (Refreshed every loop)
    CURRENT=$(get_active_vpn)

    if [[ -n "$CURRENT" ]]; then
        TITLE=" Info SELECT A VPN TO ACTIVATE - Info ACTIVE: $CURRENT"
    else
        TITLE=" Info SELECT A VPN TO ACTIVATE - Info DISCONNECTED"
    fi

    # 2. Show Menu
    SELECTED=$(ls *.conf 2>/dev/null | zenity --window-icon=$Icon --
text="$TITLE" --list --height 600 --column="File Name" --ok-
label="Activate" --extra-button="Deactivate" --extra-button="Cleanup"

    EXIT_CODE=$?

    # 3. Cancel
    if [ $EXIT_CODE -eq 1 ] && [ -z "$SELECTED" ]; then
        exit 0
    fi

    # 4. Cleanup Button
    if [ "$SELECTED" = "Cleanup" ]; then
        do_cleanup
        continue
    fi

    # 5. Deactivate Button
    if [ "$SELECTED" = "Deactivate" ]; then
        if [[ -n "$CURRENT" ]]; then
            do_hard_kill "$CURRENT"
            zenity --window-icon=$Icon --info --text="Info $CURRENT
deactivated."
        else
            zenity --window-icon=$Icon --info --text="Info No VPN active."
        fi
        continue
    fi

    # 6. Handle Activate Button - new or switch VPN
    if [ $EXIT_CODE -eq 0 ]; then
        if [ -z "$SELECTED" ]; then
            zenity --window-icon=$Icon --warning --text="Info No file
selected."
            continue
        fi

        # If a VPN is active, stop it silently first
        if [[ -n "$CURRENT" ]]; then
            echo "Switching from $CURRENT to $SELECTED..."
        fi
    fi
}
fi
```

Saját szkript a Wireguard VPN-kapcsolatod kezelésére

```
sudo wg-quick down "$CURRENT"
# Wait for kernel to fully clear routes (Critical for stability)
sleep 2
fi
break
fi
done

# --- Final Activation ---
echo "ÖŸ€ Activating: $SELECTED"
FILE="{SELECTED%. *}"

# Safety Check: Ensure no other VPN is active before starting
sleep 1
FINAL_CHECK=$(get_active_vpn)

if [[ -n "$FINAL_CHECK" ]]; then
# If the same interface is still there, try one last time to stop it
if [[ "$FINAL_CHECK" == "$CURRENT" ]]; then
echo "Warning: Interface $FINAL_CHECK still active. Retrying
stop..."
sudo wg-quick down "$FINAL_CHECK"
sleep 1
FINAL_CHECK=$(get_active_vpn)
fi

if [[ -n "$FINAL_CHECK" ]]; then
zenity --window-icon=$Icon --error --text="Info Failed: Another
VPN ($FINAL_CHECK) is still active."
exit 1
fi
fi

sudo wg-quick up "$FILE"

if [ $? -eq 0 ]; then
zenity --window-icon=$Icon --info --text="Info VPN $FILE activated
successfully."
```

```
else
zenity --window-icon=$Icon --error --text="Info Failed to activate
$FILE."
fi
```

Letöltheted a bash szkriptet a magazin szerveréről, [innen](#). Tedd a fájlt abba a könyvtárba, ahol általában a bash szkripteket tárolod. Ügyelj arra, hogy a fájlnevet `wireguard-vpn-manager.sh.txt`-ről `wireguard-vpn-manager.sh`-re változtasd, és a fájlt tedd végrehajthatóvá. A bash script 5.2 KiB méretű, ezért elég gyorsan le kell töltnie.

Megjegyzés a SUDO-hoz

Használatának volt egy aspektusa, amiről megfeledeztem, hogy a szükséges sudo jogosultságokat hozzá kell rendelnem a felhasználóhoz. Ez a `/etc/sudoers` fájl egyszerű szerkesztésével történik.

Mint tudjuk, a teljes jogosultság megszerzéséhez su kell a root-nak ami a fájlban ezzel a bejegyzéssel érhető el: **root ALL=(ALL:ALL) ALL**

Tehát ha azt akarjuk, hogy felhasználónk hasonló jogosultságokkal rendelkezzen, adjuk hozzá ezt a bejegyzést: **user ALL=(ALL:ALL) ALL**

Mivel a sudo-t egy szkripten belül használjuk, nem tudni, mikor kér jelszót, ezért megadjuk, hogy a sudo ne kérjen jelszót az adott felhasználó estén: **user ALL=(ALL:ALL) NOPASSWD: ALL**

A hatása az, hogy a sudo soha nem kér jelszót a megadott „felhasználótól”. A többi felhasználónak továbbra is meg kell adnia jelszavát.

A jelszót nem kérő parancsok pontosabb meghatározásához a sudoers bejegyzés a következő lehet **user ALL=(ALL) NOPASSWD: /usr/bin/wg, /usr/bin/wg-quick, /sbin/ip, /usr/bin/nmcli, /usr/libexec/nm-ifup, /usr/libexec/nm-ifdown, /sbin/dhclient**

Ez az adott a felhasználónak lehetővé teszi a parancsok sudo-s használatát anélkül, hogy beírná a jelszót és így működni kell a szkriptben. Az összes

Saját szkript a Wireguard VPN-kapcsolatod kezelésére

más parancshoz, ami megemelt jogosultságot igényel, meg kell adni a jelszót. Az általános bejegyzés, ami kell, hogy a meghatározottakon túli parancsokhoz SUDO használható legyen **user ALL=(ALL:ALL) NOPASSWD: ALL**

Úgy gondolom, hogy a parancsokat megadó fenti sornak elegendőnek kell lennie a közzétett szkriptek futtatásához.

További frissítés!

Az eredeti (fenti) bash szkript közzététele után Cuig újraírta a szkriptet Python szkriptként. A Python-szkript további előnye, hogy az értesítési területen lévő ikonon keresztül dinamikusan mutatja, hogy a Wireguard VPN-kapcsolat jelenleg be- vagy kikapcsolva van-e.

A Python szkript [itt](#) PCLINUXOS fórumában található meg. A Python szkript egy kicsit nagyobb (körülbelül 64 KiB, és a szkript mérete miatt két bejegyzésre oszlik). A szkript Python verzióját letöltheted a magazin szerveréről is, [innen](#). Csakúgy, mint a magazin webhelyéről letöltött bash szkripteknél, tárold a fájlt abban a könyvtárban, ahol általában a szkripteket tartod, távolítsd el a „.txt” fájlkiterjesztést, és tedd végrehajthatóvá. A futtatáshoz a parancs „python3 wireguard-vpn-manager.py” lesz.



The PCLinuxOS Magazine Special Editions!

A collage of six PCLinuxOS Magazine Special Editions covers. The covers include: 'Windows Migration Guide' (September 2013), 'Enlightenment' (May 2011), 'The KDE 4 SC Special Edition' (Fall 2010), 'GTK: Lightweight Desktops: Xfce & LXDE Special Edition' (November 2010), 'Command Line Interface Intro' (October 2010), and 'Openbox Special Edition' (March 2012).

Get Your Free Copies Today!