National Supercomputing Centre (NSCC) Singapore
1 Fusionopolis Way, Connexis South,
#17-01, Singapore 138632
E: contact@nscc.sg
Reg no: 199702110H

Quick Guide | 1 Oct 2026

Check Point VPN Access on Linux with snx-rs

Installing and configuring the snx-rs client for vpn2.nscc.sg on Ubuntu 24.04 LTS and CachyOS

Document ID	NSCC-QG-VPN-001	**Version**	1.0
Owner	NSCC System Team	**Last revised**	1 Oct 2026
System	Check Point remote access VPN (vpn2.nscc.sg)	**Classification**	Public

Disclaimer

snx-rs is an open-source client developed by a third party. It is not developed, supported or endorsed by Check Point or NSCC, and is provided for users who choose to use it at their own risk. NSCC does not provide support for this client. The supported clients are Check Point Endpoint Security VPN for Windows and macOS.

Any links to third-party software available in this guide are provided "as is" without warranty of any kind, either expressed or implied, and such software is to be used at your own risk. The use of the third-party software links in this guide is done at your own discretion and risk and with agreement that you will be solely responsible for any damage to your computer system or loss of data that results from such activities. You are solely responsible for adequate protection and backup of the data and equipment used in connection with any of the software linked to this guide, and we will not be liable for any damages that you may suffer in connection with downloading, installing, using, modifying or distributing such software. No advice or information, whether oral or written, obtained by you from us or from this guide shall create any warranty for the software. Additionally, we make no warranty that:

- The third-party software will meet your requirements.
- The third-party software will be uninterrupted, timely, secure or error-free.
- The results from the use of the third-party software will be effective, accurate or reliable.
- The quality of the third-party software will meet your expectations.
- If errors or problems occur in connection with a download of the third-party software obtained from the links in this document, they will be corrected.

By continuing with this guide, you confirm that you have read, understood and agree to this disclaimer. If you do not agree, do not proceed with the rest of this guide, and use the Check Point Endpoint Security VPN client for Windows or macOS instead.

1 Overview

This guide describes how to install and configure snx-rs, an open-source client for Check Point VPN tunnels, so that a Linux workstation can connect to the NSCC Check Point remote access VPN gateway `vpn2.nscc.sg`. It covers Ubuntu 24.04 LTS and CachyOS (an Arch Linux-based distribution).

Check Point does not provide a supported Endpoint Security VPN client for Linux. snx-rs implements the Check Point client protocol, including the CISCO_DUO_LDAP login option (LDAP password plus Cisco Duo second factor) used on vpn2, and connects over the same path as the Windows client.

1.1 Conventions

Lines starting with $ are commands to run; the $ is the shell prompt and is not typed. Indented lines without a prompt continue the command above.

Shaded boxes show the contents of a file exactly as it should appear.

2 Prerequisites

- Ubuntu 24.04 LTS Desktop, or CachyOS with a desktop environment. KDE Plasma is recommended on CachyOS.
- A desktop environment with a system tray (StatusNotifierItem support). Ubuntu and KDE Plasma provide this by default.
- An NSCC LDAP account that is authorised for remote access VPN and enrolled in Cisco Duo.
- Outbound access to `vpn2.nscc.sg` on TCP 443, plus internet access for package installation.
- sudo rights on the workstation.

Confirm the gateway is reachable before you start. The command must report that the connection succeeded.

```
$ nc -zv vpn2.nscc.sg 443
```

3 Installation on Ubuntu 24.04 LTS

snx-rs publishes a signed APT repository. The package includes the command-line client, the background service and the graphical tray application.

1. Update the package lists.

   ```
   $ sudo apt update
   ```

2. Add the snx-rs APT repository.

   ```
   $ sudo curl -fsSL -o /etc/apt/sources.list.d/snx-rs.sources \
         https://ancwrd1.github.io/snx-rs/snx-rs.sources
   $ sudo apt update
   ```

> **Note:** If `apt update` reports `NO_PUBKEY` for the snx-rs repository, add the signing key with `sudo curl -fsSL -o /usr/share/keyrings/snx-rs.gpg https://ancwrd1.github.io/snx-rs/KEY.gpg` and run `sudo apt update` again.

3. Install snx-rs.

   ```
   $ sudo apt install -y snx-rs
   ```

4. Enable and start the snx-rs background service.

```
$ sudo systemctl enable --now snx-rs
$ systemctl status snx-rs
```

The service must show `active (running)`.

5. Download the GlobalSign intermediate certificate and convert it to PEM format.

```
$ curl -fsSL -o /tmp/gs-intermediate.der \
      http://secure.globalsign.com/cacert/gsrsaovsslca2018.crt
$ openssl x509 -inform DER -in /tmp/gs-intermediate.der -out /tmp/gs-intermediate.crt
$ openssl x509 -in /tmp/gs-intermediate.crt -noout -subject
```

The subject must read `CN = GlobalSign RSA OV SSL CA 2018`.

6. Add the intermediate certificate to the system trust store.

```
$ sudo cp /tmp/gs-intermediate.crt \
      /usr/local/share/ca-certificates/globalsign-rsa-ov-ssl-ca-2018.crt
$ sudo update-ca-certificates
```

The output reports `1 added`. See section 8 for why this step is needed.

7. Restart the snx-rs service so that it loads the updated trust store.

```
$ sudo systemctl restart snx-rs
```

4 Configuring the vpn2 Profile

This procedure is the same on both distributions. First confirm that the workstation now trusts the vpn2 certificate chain.

```
$ openssl s_client -connect vpn2.nscc.sg:443 -servername vpn2.nscc.sg </dev/null 2>&1 \
      | grep "Verify return code"
```

The expected output is:

```
Verify return code: 0 (ok)
```

Then create the connection profile in the snx-rs graphical interface.

1. Open the application menu and launch SNX-RS VPN client. Its icon appears in the system tray.



Figure 1. SNX-RS VPN client in the application menu

2. Click the snx-rs tray icon and choose Settings.

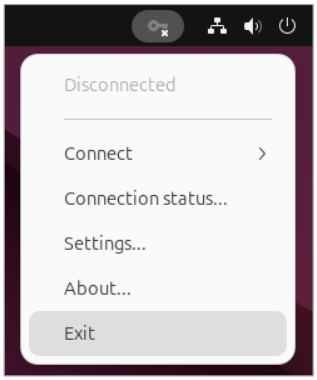


Figure 2. snx-rs tray menu on Ubuntu

3. Under Connection profiles, click + and name the profile VPN2.
4. On the General tab, enter `vpn2.nscc.sg` as the VPN server address.
5. Click the refresh button (circular arrow) at the right of the address field to fetch the login options from the gateway. The Authentication method list is filled in when the fetch succeeds.

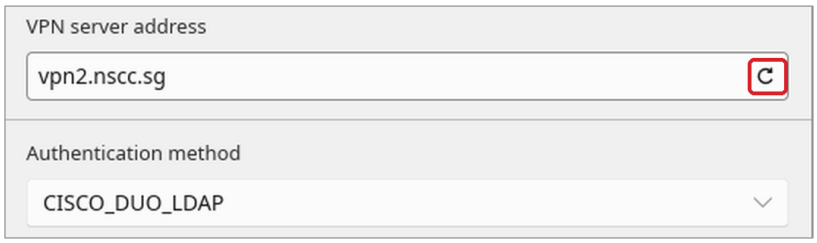


Figure 3. Refresh button (highlighted) that fetches the login options from vpn2

6. In Authentication method, choose CISCO_DUO_LDAP.
7. Set Tunnel type to SSL and leave Machine certificate authentication switched off.
8. Enter your LDAP username in Username. Enter your LDAP password in Password, or leave it empty so that snx-rs prompts for it at connect time. A saved password is stored in the OS keychain.
9. Click Apply, then OK.

Table 1. vpn2 profile settings

Setting	Value
VPN server address	vpn2.nscc.sg
Authentication method	CISCO_DUO_LDAP
Tunnel type	SSL
Username	Your LDAP username
Password	Your LDAP password, or empty to be prompted at connect time
Machine certificate authentication	Off

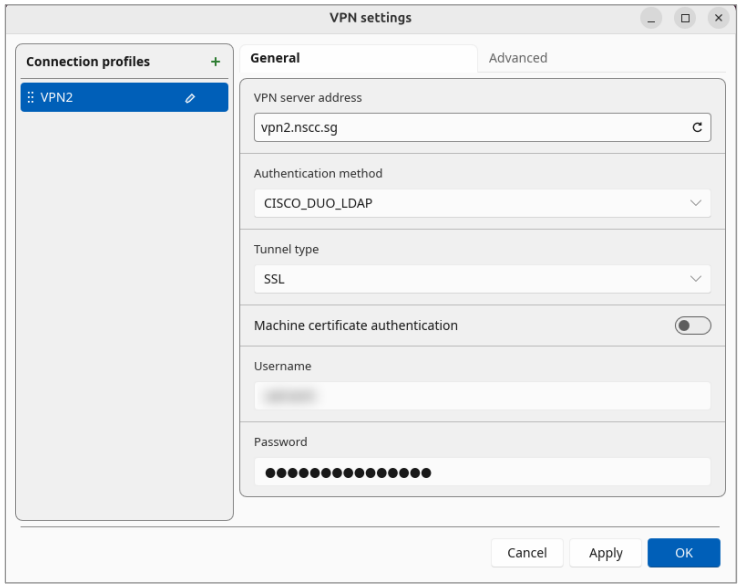


Figure 4. snx-rs VPN settings for vpn2 (SSL tunnel, CISCO_DUO_LDAP)

5 Connecting and Disconnecting

1. Click the snx-rs tray icon, choose Connect, then choose the VPN2 profile (see Figure 2).
2. Enter your LDAP password when prompted, if it is not saved in the profile.
3. Approve the Cisco Duo push on your phone, or enter a Duo passcode if prompted.
4. Wait for the tray icon to show the connected state.

To disconnect, click the snx-rs tray icon and choose Disconnect.

6 Verification

Click the snx-rs tray icon and choose Connection status. A working connection shows the following values.

Table 2. Connection information for a working vpn2 session

Field	Expected value
Status	Connected
Login type	vpn_DUO_LDAP
Tunnel type	SSL
IP address	An Office Mode address assigned by vpn2
DNS servers	NSCC internal DNS servers, assigned by vpn2
Interface	snx-tun
DNS configured / Routing configured	true / true
Default route	false (split tunnel: only NSCC networks use the VPN)

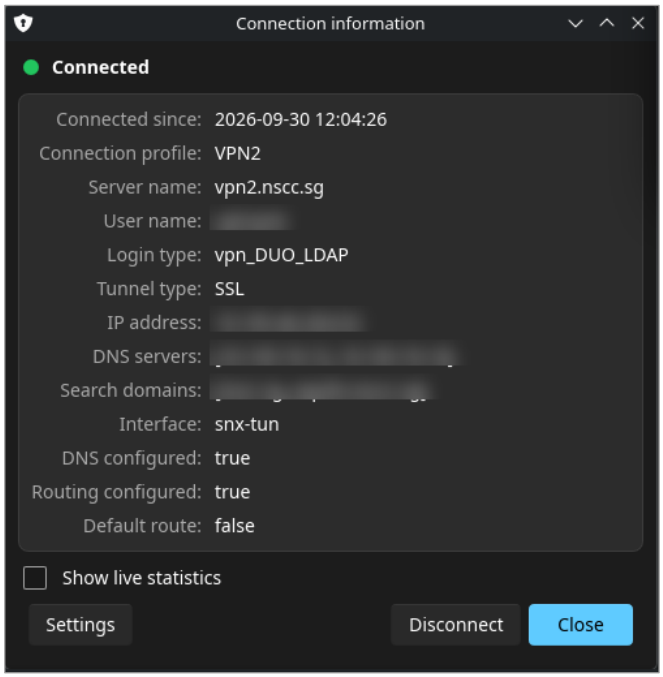


Figure 5. snx-rs connection information for a working vpn2 session

From a terminal, confirm that the tunnel interface has an address and that NSCC routes point to it.

```
$ ip -brief addr show snx-tun
$ ip route | grep snx-tun
```

Finally, connect to an internal host that is reachable only through vpn2, for example an Aspire2B login node. If the connection fails, follow the service log while reconnecting.

```
$ ssh <username>@aspire2b.nscc.sg
$ sudo journalctl -u snx-rs -f
```

7 Installation on CachyOS

On CachyOS, snx-rs is installed from the Arch User Repository (AUR) and is compiled from source. The build downloads and compiles several hundred Rust crates and takes approximately 5 to 15 minutes depending on CPU.

1. Update the system fully. Do not install packages after a partial upgrade.

   ```
   $ sudo pacman -Syu
   ```

2. Check whether an AUR helper is installed.

   ```
   $ which paru yay
   ```

3. If neither is installed, install paru from the CachyOS repositories.

   ```
   $ sudo pacman -S paru
   ```

> **Note:** Choose `aur/snx-rs`. Do not use `snx-rs-headless` on its own, which has no graphical interface, or `snx-rs-git`, which is an outdated package.

```
~
❯ paru -Ss snx-rs
aur/snx-rs 6.4.1-1 [+9 ~0.05]
    Rust client for Checkpoint VPN tunnels
aur/snx-rs-headless 6.4.1-1 [+9 ~0.05]
    Rust client for Checkpoint VPN tunnels
aur/snx-rs-git 2.2.1.r350.g0cf63a9-5 [+3 ~0.00]
    Open source VPN client for Checkpoint security gateways (git version)
```

Figure 6. paru search results for snx-rs; install aur/snx-rs

4. Build and install snx-rs. Run this as your normal user, not with sudo.

   ```
   $ paru -S snx-rs
   ```

Answer the prompts as follows:

- When asked for a provider for cargo, choose `rust` from the `cachyos-extra-v3` repository (the default, option 1). Do not choose `rustup`.

```
❯ paru -S snx-rs
:: Resolving dependencies...

:: There are 4 providers available for cargo:
:: Repository cachyos-extra-v3:
    1) rust  2) rustup
:: Repository extra:
    3) rust  4) rustup
Enter a number (default=1): 1
:: Calculating conflicts...
:: Calculating inner conflicts...

Repo (10)                      Old Version  New Version          Make Only
cachyos-extra-v3/enchant                    2.8.21-1.1           No
cachyos-extra-v3/harfbuzz-icu               14.5.0-1.1           No
cachyos-extra-v3/hyphen                     2.8.9-1.1            No
cachyos-extra-v3/libavif                    1.4.2-1.1            No
cachyos-extra-v3/libmanette                 0.2.13-2.1           No
cachyos-extra-v3/libyuv                     r2921+644251f25-1.1  No
cachyos-extra-v3/rust                       1:1.98.1-1.1         Yes
cachyos-extra-v3/woff2                      1.0.2-6.1            No
cachyos-extra-v3/xdg-dbus-proxy             0.1.9-1.1            No
extra/webkitgtk-6.0                         2.52.6-1             No

Aur (2)                        Old Version  New Version          Make Only
aur/snx-rs-headless                         6.4.1-1              No
aur/snx-rs                                  6.4.1-1              No

:: Proceed to review? [Y/n]: y
```

Figure 7. paru dependency summary after choosing rust as the cargo provider

- Answer y at Proceed to review. In the PKGBUILD, confirm that the source points to `github.com/ancwrd1/snx-rs`, press q to leave the viewer, then answer Y at Accept changes? to start the build.

```
Paging with less. Press 'q' to quit or 'h' for help.

:: snx-rs:
  PKGBUILD:
    # Maintainer: Zdeněk Biberle <zdenek at biberle dot net>
    pkgbase=snx-rs
    pkgname=(snx-rs-headless snx-rs)
    pkgver=6.4.1
    pkgrel=1
    pkgdesc="Rust client for Checkpoint VPN tunnels"
    arch=(x86_64)
    url=https://github.com/ancwrd1/snx-rs
    license=(AGPL-3.0-only)
    makedepends=(cargo gtk4 webkitgtk-6.0)
    source=("$pkgbase-$pkgver.tar.gz::https://github.com/ancwrd1/$pkgbase/archive/refs/tags/v$pkgver.tar.gz")
    sha256sums=('0af86dc45830d724b84bd0fdb077084a4c00cfcf4e8a76e3fe78daed30a68658')

    prepare() {
      cd "$pkgbase-$pkgver"
      export RUSTUP_TOOLCHAIN=stable
      cargo fetch --locked --target "$CARCH-unknown-linux-gnu"
    }

    build() {
      cd "$pkgbase-$pkgver"
      export RUSTUP_TOOLCHAIN=stable
      export CARGO_TARGET_DIR=target
      cargo build --frozen --release --features mobile-access
    }

    check() {
      cd "$pkgbase-$pkgver"
      export RUSTUP_TOOLCHAIN=stable
      cargo test --frozen --features mobile-access -- --skip platform::linux::tests::test_xfrm_check
    }

    package_snx-rs-headless() {
      depends=(glibc libgcc openssl sqlite)
      conflicts=("snx-rs<${pkgver}-${pkgrel}")
      replaces=("snx-rs<${pkgver}-${pkgrel}")

      cd "$pkgbase-$pkgver"
      install -Dm0755 -t "$pkgdir/usr/bin/" target/release/{snx-rs,snxctl}
      install -Dm0644 -t "$pkgdir/usr/lib/systemd/system/" package/snx-rs.service
:: Accept changes? [Y/n]: Y
```

Figure 8. snx-rs PKGBUILD review in paru

The installation also pulls in `snx-rs-headless` (background service and command-line client) and `webkitgtk-6.0` (embedded browser for web-based logins). This is expected.

> **Note:** During the build, the terminal shows a long list of Downloading and Compiling lines. These are the Rust libraries (crates) that snx-rs is built from. They are compiled into snx-rs and are not installed as separate system packages. This is normal and takes several minutes. Every snx-rs update from the AUR is compiled the same way. Build files are kept in `~/.cache/paru` and can be removed afterwards with `paru -Sc`.

5. Enable and start the snx-rs background service.

```
$ sudo systemctl enable --now snx-rs
$ systemctl status snx-rs
```

The service must show `active (running)`.

6. Download the GlobalSign intermediate certificate and convert it to PEM format.

```
$ curl -fsSL -o /tmp/gs-intermediate.der \
      http://secure.globalsign.com/cacert/gsrsaovsslca2018.crt
$ openssl x509 -inform DER -in /tmp/gs-intermediate.der -out /tmp/gs-intermediate.crt
```

7. Add the intermediate certificate to the system trust store.

```
$ sudo cp /tmp/gs-intermediate.crt \
      /etc/ca-certificates/trust-source/anchors/globalsign-rsa-ov-ssl-ca-2018.crt
$ sudo update-ca-trust
```

8. Restart the snx-rs service so that it loads the updated trust store.

```
$ sudo systemctl restart snx-rs
```

Then create the vpn2 connection profile, connect and verify the connection as described in sections 4 to 6.

8 Troubleshooting

Table 3. Common problems

Symptom	Cause	Action
"TLS certificate validation failed" when fetching login options	The GlobalSign intermediate certificate is missing from the system trust store.	Install the intermediate certificate (sections 3 and 7), then restart the snx-rs service.
Authentication method list stays empty	Certificate validation failed, or TCP 443 to vpn2 is blocked.	Run `nc -zv vpn2.nscc.sg 443` and the `openssl s_client` check in section 4.
Tray menu opens empty (Ubuntu)	The GNOME AppIndicator extension has not loaded the snx-rs menu.	Restart the tray application with `pkill -f snx-rs-gui` and then `snx-rs-gui &`, or log out and back in.
No snx-rs tray icon	The desktop environment has no system tray support.	Run `snx-rs-gui --no-tray` to open a status window. On GNOME, enable the AppIndicator extension.
Tunnel does not come up with Tunnel type IPsec	vpn2 is used with the SSL tunnel type.	Change Tunnel type to SSL in the profile and reconnect.
`paru: command not found` on CachyOS	No AUR helper is installed.	Run `sudo pacman -S paru`, or use `yay` if it is present.

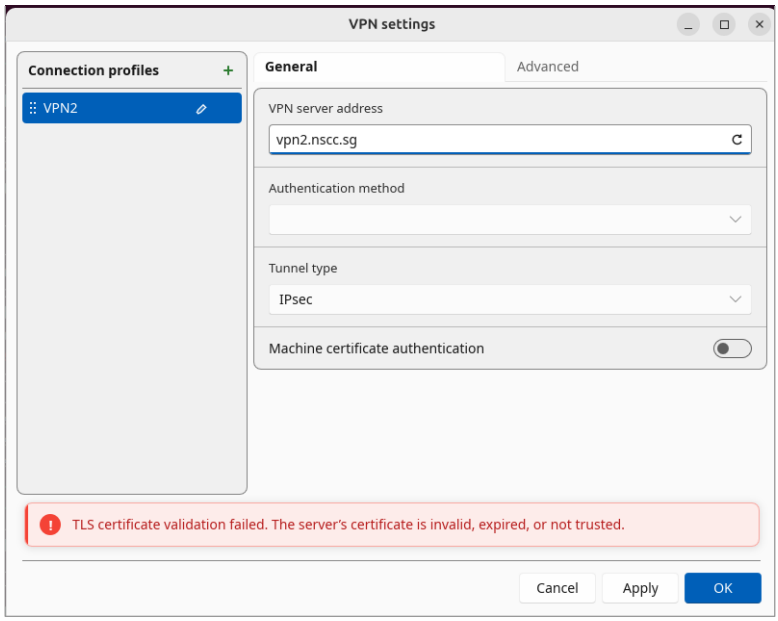


Figure 9. TLS certificate validation error shown when the GlobalSign intermediate certificate is missing

9 Rollback

To remove snx-rs and the added certificate on Ubuntu 24.04 LTS:

```
$ sudo systemctl disable --now snx-rs
$ sudo apt remove -y snx-rs
$ sudo rm /etc/apt/sources.list.d/snx-rs.sources
$ sudo rm /usr/local/share/ca-certificates/globalsign-rsa-ov-ssl-ca-2018.crt
$ sudo update-ca-certificates --fresh
```

To remove snx-rs and the added certificate on CachyOS:

```
$ sudo systemctl disable --now snx-rs
$ paru -Rns snx-rs snx-rs-headless
$ sudo rm /etc/ca-certificates/trust-source/anchors/globalsign-rsa-ov-ssl-ca-2018.crt
$ sudo update-ca-trust
```

10 References

- snx-rs project: https://github.com/ancwrd1/snx-rs
- snx-rs package repository: https://ancwrd1.github.io/snx-rs/
- snx-rs quick start: https://github.com/ancwrd1/snx-rs/blob/main/docs/quick-start.md
- GlobalSign RSA OV SSL CA 2018 intermediate: http://secure.globalsign.com/cacert/gsrsaovsslca2018.crt