Network

For network related topics : WiFi, WireGuard VPN, DNS...

- WiFi : Eduroam (WPA2 enterprise)
- WireGuard
  - Installing WireGuard
  - [EXAMPLE]: Using WireGuard with MullvadVPN
  - Where's the configuration file saved?
- Custom DNS
  - Stop connman from managing DNS
  - Use dnsmasq for DNS
  - Install & configure unbound DNS server

WiFi : Eduroam (WPA2 enterprise)

As detailed in entries on the forum .... (https://forum.sailfishos.org/t/eduroam-wpa2-enterprise)

Configuration

Encryption: WPA-EAP(TTLS)
EAP method: TTLS
Inne authentication: PAP
CA Certificate: No verfication
Identity: your email address
Password: your password

Auth may require: PEAP/MSCHAPv2

Manual Conman config

```
devel-su
vi /home/nemo/.local/share/system/privileged/connman/wifi_eduroam.config
paste it
[service_eduroam]
Type=wifi
Name=eduroam
EAP=ttls
CACertFile=/etc/ssl/certs/exampleCA.crt #path to your CA certificate (if you really need certificate)
Phase2=PAP
Identity=your email address
Passphrase=your password
```

Convert .p12 file to pems and config

```
#!/bin/sh
# easyroam.sh cert - install pkcs12 certificate as Easyroam NetworkManager Profile
```

```
helpString="Usage $0 <certificate>"
if [ $# -lt 1 ]; then
	echo "$helpString" >&2
	exit 1
fi

case "$1" in
-h|--help)
	echo "$helpString" >&2
	exit;;
esac

ClientCertificate="$1"
connection="Easyroam"

[ -f /etc/os-release ] &&  . /etc/os-release

check_nmcli() {
	# check for nmcli
	if ! type nmcli >/dev/null 2>&1; then
		echo "ERROR: nmcli not found!" >&2
		echo "This wizard assumes that your network connections are managed by NetworkManager." >&2
		exit 1
	fi
}

check_gdbus() {
	if ! type gdbus >/dev/null 2>&1; then
		echo "ERROR: gdbus not found!" >&2
		echo "This wizard assumes that your network connections are managed by NetworkManager." >&2
		exit 1
	fi
}

cleanup_networkmanager() {
	# Remove existing connections
	for conn in $connection eduroam; do
		for uuid in $(nmcli connection show | awk '$1==c{ print $2 }' c="$conn"); do
			nmcli connection delete uuid "$uuid"
		done
```

```
        done
}

add_networkmanager() {
        # Create new connection
        nmcli connection add \
                type wifi \
                con-name "$connection" \
                ssid "$SSID" \
                -- \
                wifi-sec.key-mgmt wpa-eap \
                802-1x.eap tls \
                802-1x.identity "$OuterIdentity" \
                802-1x.ca-cert "$root_ca_file" \
                802-1x.client-cert "$client_cert_file" \
                802-1x.private-key-password "$Passphrase" \
                802-1x.private-key "$client_key_file"
}

add_connman() {
        devel-su gdbus call --system --dest net.connman  --object-path / --method
net.connman.Manager.CreateService \
        "" \
        "" \
        "" \
        "[('AutoConnect', 'true'), ('CACert', '$(cat "$root_ca_file")'),('ClientCertFile', '$client_cert_file'),
        ('PrivateKeyFile', '$client_key_file'), ('PrivateKeyPassphrase', '$Passphrase'),
        ('EAP', 'tls'), ('Hidden', 'false'), ('Identity', '$OuterIdentity'), ('Name', 'eduroam'),
        ('Phase2', 'PAP'), ('Security', 'ieee8021x')]"

}

if [ "$ID" = "sailfishos" ]; then
        check_gdbus
else
        check_nmcli
fi

# check prerequisites
for d in openssl awk; do
```

```
	type "$d" >/dev/null 2>&1 && continue
	echo "ERROR: $d not found!" >&2
	echo >&2
	echo "You may fix this using:" >&2
	type apt	>/dev/null 2>&1 && echo "sudo apt install -y $d" >&2
	type dnf	>/dev/null 2>&1 && echo "sudo dnf install -y $d" >&2
	type zypper	>/dev/null 2>&1 && echo "sudo zypper install $d" >&2
	type pacman	>/dev/null 2>&1 && echo "sudo pacman -Syu $d" >&2
	type pkcon	>/dev/null 2>&1 && echo "devel-su pkconf install $d" >&2
	type xbps-install >/dev/null 2>&1 && echo "sudo xbps-install -Su $d" >&2
	echo >&2
	exit 2
done

conf_dir="$HOME/.easyroam"
client_cert_file="$conf_dir/easyroam_client_cert.pem"
client_key_file="$conf_dir/easyroam_client_key.pem"
root_ca_file="$conf_dir/easyroam_root_ca.pem"

[ -d "$conf_dir" ] || mkdir -p "$conf_dir"

openssl_extra=
version=$(openssl version | awk -F "[ .]" '{print $2}')
[ "${version:-2}" -ge 3 ] && openssl_extra="-legacy"

SSID=eduroam
OuterIdentity="$(openssl pkcs12 $openssl_extra -info -passin "pass:" -in "$ClientCertificate" -nodes 2>/dev/null
| awk '/subject=CN/{print $3}' | sed -e 's/,$//g')"
Passphrase=$(openssl rand -base64 24)

printf "Extracting client cert ... "
openssl pkcs12 $openssl_extra -in "$ClientCertificate" -passin "pass:" -nokeys -out "$client_cert_file"
printf "success\n"

printf "Extracting client key ... "
openssl pkcs12 $openssl_extra -in "$ClientCertificate" -passin "pass:" -passout "pass:$Passphrase" -nodes -
nocerts | \
	openssl rsa -passout "pass:$Passphrase" -aes128 -out "$client_key_file"

printf "Extracting CA cert ... "
```

```
openssl pkcs12 $openssl_extra -passin "pass:" -passout "pass:" -nokeys -in "$ClientCertificate" -cacerts -out
"$root_ca_file"
printf "success\n"

if [ "$ID" = "sailfishos" ]; then
        add_connman
else
        cleanup_networkmanager
        add_networkmanager
fi
```

WireGuard

How to use WireGuard VPNs on SailfishOS. Tutorials for various VPN providers

Installing WireGuard

Install necessary tools

With Sailfish OS 5.0 (2026)

Since there is no official documentation so far, there is a short how-to make WireGuard working.

```
devel-su pkcon remove wireguard-go wireguard-tools  # for those whom previously installed those two packages in a previous setup or attempt
devel-su pkcon install jolla-settings-networking-plugin-vpn-wireguard  # adds Wireguard among the proposed VPN in the settings
devel-su systemctl restart connman  # restart the connexion manager to load the new Wireguard plugin
```

Then, especially for those that have been using WireGuard already, you have to forget WireGuard configs and set it from scratch or import again.

Once your file is imported, you are presented a WireGuard-VPN-connexion configuration interface and it is advised to activate two settings:

- Advanced > Remember authentication information
- Advanced > Enable IPv6 data leak protection

Thanks to @kan_ibal https://forum.sailfishos.org/t/wireguard-in-sailfishos-5-0/22346 and @dopi04 https://forum.sailfishos.org/t/wireguard-in-sailfishos-5-0/22346/26

With previous versions (2024)

SailfishOS doesn't provide WireGuard functionality out-of-the-box, so we first need to install a few third-party programs from OpenRepos. You can get all of these either by downloading the RPMs manually from openrepos.net or by using Storeman.

Wireguard userspace implementation

Install the package `wireguard-go` : Download from OpenRepos

Wireguard userspace tools

Install the package `wireguard-tools` : Download from OpenRepos

Connman plugin for integrating Wireguard into Sailfish network manager

Install the package "WireGuard for Sailfish (connman plugin)" (`connman-plugin-vpn-wireguard`):

Download from OpenRepos

VPN plugin for Sailfish OS settings app

Install the package "WireGuard for Sailfish (Settings UI)" (`jolla-settings-networking-plugin-vpn-wireguard`):

Download from OpenRepos

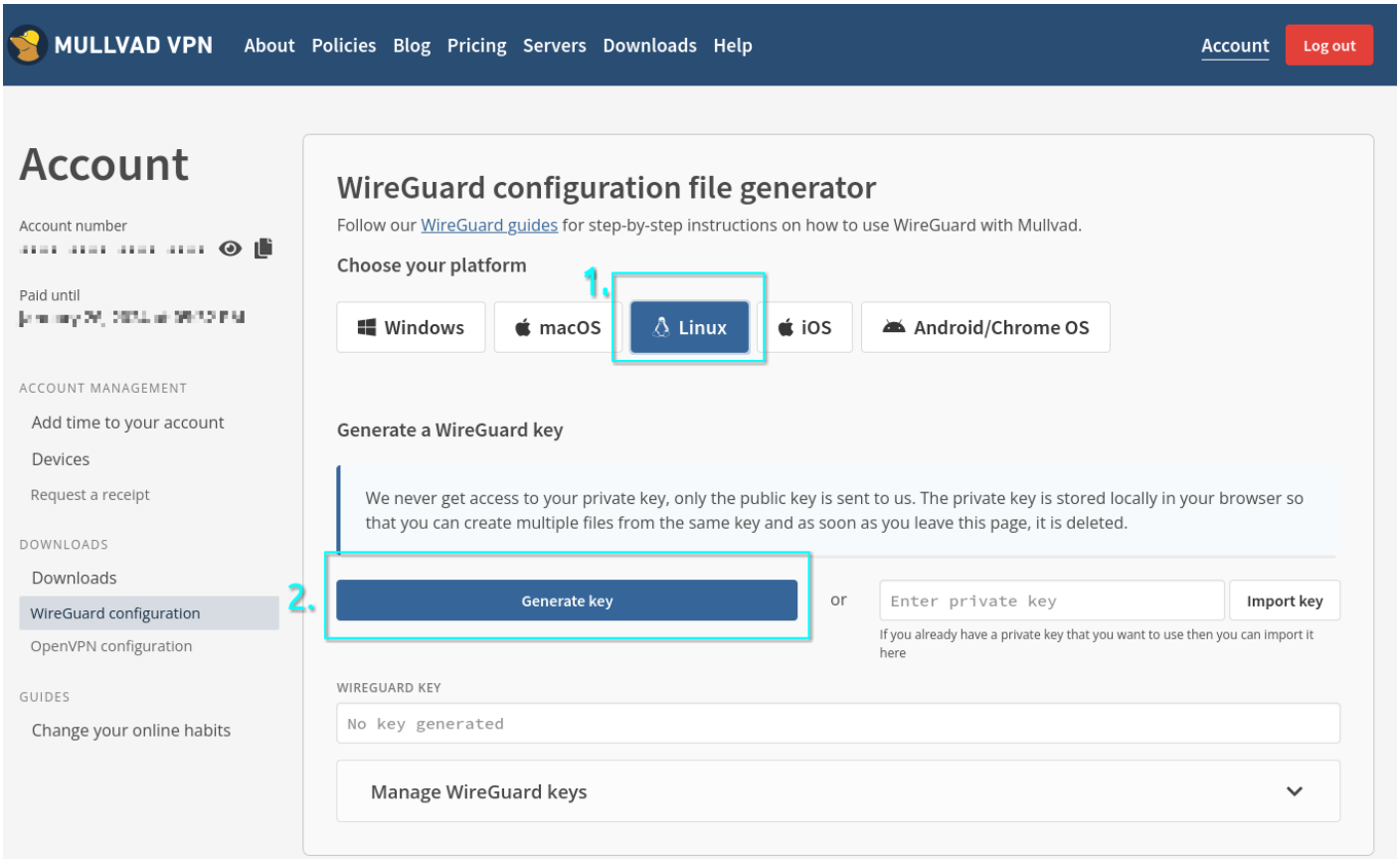
This plugin is optional but makes using WireGuard much easier.

[EXAMPLE]: Using WireGuard with MullvadVPN

Step 1: Obtain a WireGuard config file from Mullvad

To get a config file, head over to https://mullvad.net/de/account/wireguard-config (you will need to login with your account number).
1. For the "platform" select "Linux".

2.Next, click on "Generate key" (or something like that in your language).

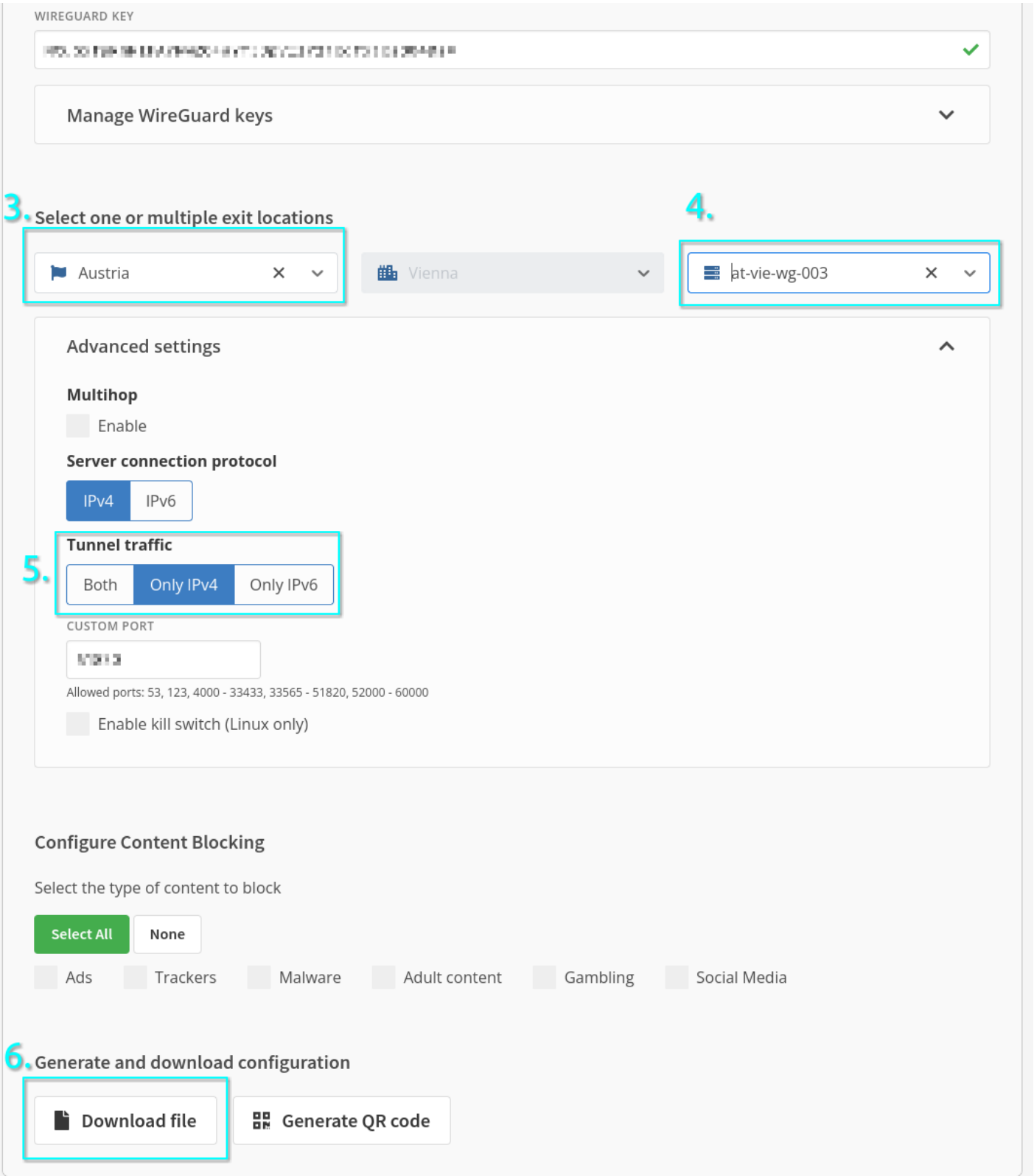


3. Select a country

4. Select a city for your country

5. For the "Tunnel traffic", choose "Only IPv4"

6. Click on "Download file". You will get a `.conf`-file.



7. Copy the file to the storage of your SailfishOS-device

Step 2: Importing the file on Sailfish

1. Open the settings-app and navigate to "VPN"
2. In the pull-down menu, choose "Add new VPN"
3. Choose "WireGuard" for the VPN-Type (typically located at the bottom of the list)
4. On the next page, choose to import a "wg-quick.conf" file.
5. A file-chooser-dialog will open. Choose the `.conf`-file we downloaded earlier.
6. The setup will tell you that the import was successful. Now you just need to give the VPN a name.
7. I recommend to choose "Remember login credentials" (or something like that in your language) under the "Advanced"-settings
8. You can now select the VPN in the list to connect to it.
   NOTE: If SailfishOS asks for credentials to login, the username is your account number and the password is just the letter "m"

Needing help?

If you run into troubles, feel free to ask for help here

WireGuard

Where's the configuration file saved?

Wireguard on Sailfish OS: where's the configuration file saved?

May 05, 2025 — Nico Cartron

As usual, posting it on my blog for me, but it will probably be useful to others: I was (re)configuring Wireguard on my X10iii the other day, cause it stopped working after upgrading to Sailfish OS 5.

I could not find where the configuration was saved, but Peter G. on the Sailfish OS Telegram channel pointed me to:

```
/home/defaultuser/.local/share/system/privileged/connman-vpn
```

and indeed it's there:

```
[root@Xperia10III connman-vpn]# ls -l
total 12
drwx------  2 root  root  4096 Mar 14 09:02 provider_XXX_sailfishos_org
drwx------  2 root  root  4096 Mar 14 09:02 vpn_XXX_sailfishos_org
```

the `provider_XXX_sailfishos_org/settings` file contains everything you've defined in the SFOS GUI (IP addresses have been hidden by me):

```
[XXX_sailfishos_org]
Name=Soucelles
Type=wireguard
Host=X.X.X.X
VPN.Domain=sailfishos.org
WireGuard.Address=D.C.B.A/32
```

```
WireGuard.DNS=A.B.C.D
WireGuard.PrivateKey=<YOUR CLIENT PRIVATE CLIENT>
WireGuard.PublicKey=<YOUR SERVER PUBLIC KEY>
WireGuard.AllowedIPs=<REMOTE NETWORK>
WireGuard.EndpointPort=<WIREGUARD PORT>
WireGuard.PersistentKeepalive=15
WireGuard.DisableIPv6=false
```

and the `vpn_XXX_sailfishos_org/settings` file itself contains:

```
[vpn_XXX_sailfishos_org]
Name=Soucelles
SplitRouting=false
AutoConnect=true
Modified=2025-03-14T09:02:04Z
IPv4.method=fixed
IPv4.netmask_prefixlen=32
IPv4.local_address=D.C.B.A
IPv4.gateway=X.X.X.X
IPv6.method=off
IPv6.privacy=disabled
```

Custom DNS

Per default, the DNS settings get set automatically and there is no way to change them (v4.0.5.18). This guide shows you how to replace connman's DNS with dnsmasq and setting your own DNS config.

Stop connman from managing DNS

> If you don't have any DNS configured in /etc/resolv.conf anymore, you can't resolv any domains on the internet. It's smart to install a replacement for connman's DNS in advance. Otherwise you can temporarily solve that by executing echo "9.9.9.9" | devel-su tee /etc/resolv.conf

You need to add the `--nodnsproxy` proxy flag to connmand. This can be done by runing `devel-su systemctl edit connman` and pasting this into there:

```
[Service]
RuntimeDirectory=connman
```

Afterwards run `devel-su systemctl restart connman`.

Use dnsmasq for DNS

> As the name already implies, dnsmasq is not a full-fledged DNS server. If you want a real, recursive DNS resolver, consider using unbound instead.

Install dnsmasq

dnsmasq is available on SailfishOS chum. You 1st need to add the 3rd-party-repo, afterwards running `devel-su pkcon refresh && pkcon install dnsmasq` will install it.

Disable your current DNS handler

On a fresh installation, DNS is handled by connman. You can remove it with these instructions.

Permanently run dnsmasq on boot

ExecuteÂ `devel-su systemctl enable --now dnsmasq` to enable it permanently & start it now.

Configure dnsmasq

dnsmasq's configuration lies under `/etc/dnsmasq.conf`. You can get more information about its config options from dnsmasq's manpage and ArchWiki.

Install & configure unbound DNS server

Unbound is a full-fledge recursive DNS resolver. You should consider to not fetch from the root servers but forward your requests to another Server via DoT. Bind unbound to 127.0.0.1 to avoid access from outside.

More information can be found in ArchWiki.