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
```

Revision #2
Created 6 January 2026 09:42:54 by poetaster
Updated 23 February 2026 14:48:15 by Siltaar