SvxReflector

Inhaltsverzeichnis

Installation

Svxreflector 1.0 ist in Debian 12 enthalten. Talkgroups werden erst ab Version 2.0 unterstützt. Aktuell ist inzwischen Version 3.0, diese ist allerdings nicht mit Version 2.0 kompatibel, dh. svxlink mit der Version 3.0 kann sich nicht mit einem svxreflector der Version 2.0 verbinden.

Die folgende Anleitung gilt für Debian und Derivate (etwa Raspberry Pi OS) und erzeugt neben svxlink (Repeater-Software) auch sxvreflector (Vernetzungs-Software).

Svxlink-Code aus Github clonen

```
cd /opt
apt -y install git
git clone https://github.com/sm0svx/svxlink
cd svxlink/
cat INSTALL.adoc
```

Fehlende Pakete installieren

(hier für Debian 12)

```
apt -y install build-essential cmake doxygen pkg-config \
libsigc++-2.0-dev libasound2-dev libspeex-dev libopus-dev libogg-dev \
libpopt-dev libgcrypt20-dev libgpiod-dev librtlsdr-dev libjsoncpp-dev \
tcl-dev libgsm1-dev libcurl4-openssl-dev groff libssl-dev
```

Build

entsprechend INSTALL.adoc:

```
  cd src
  mkdir build
  cd build
  # QT4 not in Debian 12  (only QT5), skip QT UI
  # cmake .. -DUSE_QT=NO
  # Debian-style variant with further options set
  cmake -DCMAKE_INSTALL_PREFIX=/usr -DSYSCONF_INSTALL_DIR=/etc -
DLOCAL_STATE_DIR=/var -DUSE_QT=OFF -DWITH_SYSTEMD=yes ..

  make
  make doc
  useradd svxlink
  # usermod -a svxlink -G gpio
  sudo usermod -a svxlink  -G audio
  sudo make install
  sudo ldconfig
```

Nun sollte sowohl svxlink, wie auch svxreflector verfügbar sein.

Migration von svxreflector 2.0 auf 3.0

Seit der Version 3.0 ist eine Public-Key-Infrastructure (PKI) aufbauend auf OpenSSL enthalten.

Dazu muss ein Zertifikat erstellt werden, die passiert automatisch wenn der Hostname (COMMON_NAME) in der Konfiguration (svxreflector.conf) enthalten ist:

```
# version 3.0
[SERVER_CERT]
COMMON_NAME=svx.oe3xnr.hamip.at
```

Der Common-Name kann über die IP-Adresse in der HamnetDB abgefragt werden. Zur Ausgabe der Hamnet-DB ist ".hamip.at" hinzuzufügen. Beim Start wird die erfolgreiche Erstellung des Zertifikats angezeigt:

```
Tue 04 Feb 2025 05:29:38 PM CET: SvxReflector v1.2.99.26@24.02-71-gcf2ce04b
Copyright (C) 2003-2025 Tobias Blomberg / SM0SVX
Tue 04 Feb 2025 05:29:38 PM CET:
Tue 04 Feb 2025 05:29:38 PM CET: SvxReflector comes with ABSOLUTELY NO
WARRANTY. This is free software, and you are
Tue 04 Feb 2025 05:29:38 PM CET: welcome to redistribute it in accordance
with the terms and conditions in the
Tue 04 Feb 2025 05:29:38 PM CET: GNU GPL (General Public License) version 2
or later.
Tue 04 Feb 2025 05:29:38 PM CET:
Tue 04 Feb 2025 05:29:38 PM CET: Using configuration file: /etc/svxlink
/svxreflector.conf
Tue 04 Feb 2025 05:29:38 PM CET: ----------- Root CA Certificate -----------
Tue 04 Feb 2025 05:29:38 PM CET: Serial No.       :
0x7F875321F6F7ACE0C8A4F2374D130F6DB71D5176
Tue 04 Feb 2025 05:29:38 PM CET: Issuer           : CN = SvxReflector Root CA
Tue 04 Feb 2025 05:29:38 PM CET: Subject          : CN = SvxReflector Root CA
Tue 04 Feb 2025 05:29:38 PM CET: Not Before       : Tue Feb  4 17:00:19 2025
Tue 04 Feb 2025 05:29:38 PM CET: Not After        : Sat Jan 29 17:00:19 2050
Tue 04 Feb 2025 05:29:38 PM CET: ------------------------------------------
Tue 04 Feb 2025 05:29:38 PM CET: ---------- Issuing CA Certificate ----------
Tue 04 Feb 2025 05:29:38 PM CET: Serial No.       :
0x47DD2DC96697A6DB5263A95688582C33D57B2F0B
Tue 04 Feb 2025 05:29:38 PM CET: Issuer           : CN = SvxReflector Root CA
Tue 04 Feb 2025 05:29:38 PM CET: Subject          : CN = SvxReflector Issuing
CA
Tue 04 Feb 2025 05:29:38 PM CET: Not Before       : Tue Feb  4 17:00:19 2025
Tue 04 Feb 2025 05:29:38 PM CET: Not After        : Fri Jan 30 17:00:19 2026
Tue 04 Feb 2025 05:29:38 PM CET: ------------------------------------------
Tue 04 Feb 2025 05:29:38 PM CET: Server private key file not found.
Generating '/var/lib/svxlink/pki/private/svx.oe3xnr.hamip.at.key'
Tue 04 Feb 2025 05:29:38 PM CET: Generating server certificate signing
request file '/var/lib/svxlink/pki/csrs/svx.oe3xnr.hamip.at.csr'
Tue 04 Feb 2025 05:29:38 PM CET: -------- Certificate Signing Request -------
Tue 04 Feb 2025 05:29:38 PM CET: Subject          : CN = svx.oe3xnr.hamip.at
Tue 04 Feb 2025 05:29:38 PM CET: Subject Alt Name : DNS:svx.oe3xnr.hamip.at
Tue 04 Feb 2025 05:29:38 PM CET: ------------------------------------------
Tue 04 Feb 2025 05:29:38 PM CET: Generating server certificate file '/var/lib
/svxlink/pki/certs/svx.oe3xnr.hamip.at.crt'
Tue 04 Feb 2025 05:29:38 PM CET: ------------ Server Certificate ------------
Tue 04 Feb 2025 05:29:38 PM CET: Serial No.       :
0x5DCFD4727D7A3C6D749173D0561F747CC7918456
Tue 04 Feb 2025 05:29:38 PM CET: Issuer           : CN = SvxReflector Issuing
CA
Tue 04 Feb 2025 05:29:38 PM CET: Subject          : CN = svx.oe3xnr.hamip.at
Tue 04 Feb 2025 05:29:38 PM CET: Not Before       : Tue Feb  4 17:29:38 2025
Tue 04 Feb 2025 05:29:38 PM CET: Not After        : Mon May  5 18:29:38 2025
Tue 04 Feb 2025 05:29:38 PM CET: Subject Alt Name : DNS:svx.oe3xnr.hamip.at
Tue 04 Feb 2025 05:29:38 PM CET: ------------------------------------------
```

Damit sich Clients verbinden können ist auch auf der Clientseite eine entsprechende Konfiguration notwendig. Im Wesentlichen passiert diese automatisch, doch es ist zu beachten, dass der Verbindungsaufbau über exakt jenen Hostnamen erfolgt, der auch am Server als COMMON_NAME eingetragen wurde. Steht am svxlink-Rechner DNS nicht zur Verfügung, dann kann über einen Eintrag in /etc/hosts die Übersetzung von Hostnamen zu IP erfolgen. Hier ein erfolgreicher Start aus Client-Sicht:

```
Tue Feb  4 17:43:20 2025: Connected to APRS server 109.72.122.50 on port 14580
Tue Feb  4 17:43:20 2025: ReflectorLogic: Connection established to
44.143.55.139:5300 (primary)
Tue Feb  4 17:43:20 2025: ReflectorLogic: Setting up encrypted communications
channel
Tue Feb  4 17:43:20 2025: ReflectorLogic: Encrypted connection established
Tue Feb  4 17:43:20 2025: ReflectorLogic: Sending requested Certificate
Signing Request to server
Tue Feb  4 17:43:20 2025: ReflectorLogic: Authentication OK
Tue Feb  4 17:43:20 2025: ReflectorLogic: Connected nodes: OE3XNR, OE3XER
Tue Feb  4 17:43:20 2025: ------ Opus encoder parameters ------
Tue Feb  4 17:43:20 2025: Frame size           = 320
Tue Feb  4 17:43:20 2025: Complexity           = 9
Tue Feb  4 17:43:20 2025: Bitrate              = 20000
Tue Feb  4 17:43:20 2025: VBR                  = YES
Tue Feb  4 17:43:20 2025: Constrained VBR      = YES
Tue Feb  4 17:43:20 2025: Maximum audio bw     = MEDIUMBAND
Tue Feb  4 17:43:20 2025: Audio bw             = FULLBAND
Tue Feb  4 17:43:20 2025: Signal type          = VOICE
Tue Feb  4 17:43:20 2025: Application type     = AUDIO
Tue Feb  4 17:43:20 2025: Inband FEC           = NO
Tue Feb  4 17:43:20 2025: Expected Packet Loss = 0%
Tue Feb  4 17:43:20 2025: DTX                  = NO
Tue Feb  4 17:43:20 2025: LSB depth            = 16
Tue Feb  4 17:43:20 2025: ---------------------------------------
Tue Feb  4 17:43:20 2025: ------ Opus decoder parameters ------
Tue Feb  4 17:43:20 2025: Gain       = 0dB
Tue Feb  4 17:43:20 2025: ---------------------------------------
Tue Feb  4 17:43:20 2025: ReflectorLogic: Using audio codec "OPUS"
Tue Feb  4 17:43:20 2025: ReflectorLogic: Using UDP cipher id-aes128-GCM
Tue Feb  4 17:43:20 2025: ReflectorLogic: Bidirectional UDP communication
verified
```

Der genaue Ablauf ist dabei folgender (Quelle: https://groups.io/g/svxlink/topic/certificate_authentication/107698563):

Reflector side

1. The reflector is started
2. Checks if there already is a valid root CA key and cert (private/svxreflector_root_ca.key, certs/svxreflector_root_ca.crt). If not, generate those files. The root CA certificate is self-signed.
3. Checks if there already is a valid issuing CA key, csr and cert (private/svxreflector_issuing_ca.key, csrs/svxreflector_issuing_ca.csr, certs/svxreflector_issuing_ca.crt). If not, generate those files. The issuing CA certificate is signed using the root CA.
4. Checks if there is a server key, csr and cert for the reflector (private/myreflector.org.key, csrs/myreflector.org.csr, certs/myreflector.org.crt). If not, generate those files and sign the server certificate using the issuing CA.
5. Checks if there is a ca-bundle.crt file. If not, copy the root CA certificate file to ca-bundle.crt.

6. When a CSR is received from a node it is stored in pending_csrs
7. When the sysop issue the sign command, sign a CSR in the pending_csrs directory. Move the CSR to the csrs directory and store the signed certificate in the certs directory. When the node connects or already is connected, send the certificate to the node.

Node side

1. The SvxLink node is started
2. Checks if there already is key- and csr-files (CALL.key, CALL.csr). If not, generates them.
3. Connects to the reflector
4. Checks if there already is a ca-bundle.crt. If not, downloads it from the reflector server.
5. Initiates encrypted connection
6. Verifies the reflector server certificate against ca-bundle.crt
7. Verifies that the hostname used to connect to the reflector match the hostname (CN) in the certificate
8. If there already is a crt-file (CALL.crt), sends it to the reflector to authenticate. If not, sends the CSR to the reflector. The reflector will store the CSR in the pending_csrs directory.
9. If the reflector has a signed node certificate (CALL.crt), sends it to the node. The node will then disconnect and reconnect authenticating using the certificate.
10. Retry if failed.

Sprachdateien

Für die Nutzung von svxreflector ist eine aktuelle Version der Sprachdateien am Repeater (also bei svxlink) notwendig, zuletzt wurden folgende Sprachdateien ergänzt:

```
- Core/talk_group
- Core/qsy
- Core/ignored
- Core/monitor
- Default/previous
```

Diese Dateien sind auch im Download verfügbar.