WireGuard Gerät hinzufügen

PiVPN

Ich gehe davon aus, dass du als Server PiVPN nutzt. Das nimmt dir sehr viel Sachen ab und funst so weit.

neuer Client

```
pivpn -a
```

Du wirst bequem durch das Setup geleitet. Danach fällt ein QR-Code und .conf File bei raus.

- Bist du mit Smartphone unterwegs:
  - nutze die WireGuard App und scanne den QR code
- am PC:
  - zieh dir das File z.B. mit SCP
  - kopiere es nach `/etc/wireguard`

WireGuard

ich habe von irgendwo aus dem neuland mal nen script gefunden und es ein wenig angepasst, funktioniert soweit für mich:

```
#!/bin/bash

# This script creates a new "client peer" for a simple server-client WireGuard
# configuration by:
#
#   1.  Using `wg` to generate a private and public key (without exposing the
#       former in a command invocation to generate the latter)
#   2.  Creating client configuration files for both split-tunnel and
```

```
#       full-tunnel configurations
#   3.  Appending the new client to the peer list of the server configuration

# Run this command as the user who should have access to the client and server
# configuration files (i.e. probably root).


# SCRIPT CONFIGURATION

# directory storing all client and server configuration files
WG_DIR="/etc/wireguard/configs"

# VPN subnet (string-concatenated with IP_INDEX)
#   this script only supports the equivalent of netmask /24 but could be
#   modified to support others
IP_SUBNET="10.0.0"
IP_SUBNETv6="fd2d:64cb:1415:"

# location of the server configuration
SERVER_CONF="$WG_DIR/../wg0.conf"

# address and port of the WireGuard server
SERVER_ENDPOINT="vpn.hiajen.de:51820"

# server public key
SERVER_KEY_PUB=`cat $WG_DIR/../publickey`

###   ###   ###   ###   ###   ###   ###


if [ "$#" -ne 2 ]; then
    echo "Syntax: $0 CLIENT_NAME IP_INDEX"
    exit 1
fi

CLIENT_NAME="$1"
IP_INDEX="$2"

# specify the client configuration file names
CLIENT_PFX="$CLIENT_NAME"
CLIENT_CONF_SPLIT="$WG_DIR/${CLIENT_PFX}_split.conf"
```

```
CLIENT_CONF_FULL="$WG_DIR/${CLIENT_PFX}_full.conf"



# generate keys

# create a temporary file to store the private key so that it's not exposed in
# the command invocation to `wg pubkey`

CKEY_PRV_FILE=`mktemp -p "$WG_DIR"`
chmod 600 "$CKEY_PRV_FILE"
CKEY_PUB=`wg genkey | tee "$CKEY_PRV_FILE" | wg pubkey`
CKEY_PRV=`cat "$CKEY_PRV_FILE"`
rm -f "$CKEY_PRV_FILE"
CKEY_PSK=`wg genpsk`



# build the client configuration files

write_client_conf () {
cat << EOF > "$2"
[Interface]
Address = $IP_SUBNET.$IP_INDEX/24, $IP_SUBNETv6:$IP_INDEX/64
PrivateKey = $CKEY_PRV
DNS = $IP_SUBNET.1, $IP_SUBNETv6:1

[Peer]
# server
Endpoint = $SERVER_ENDPOINT
PublicKey = $SERVER_KEY_PUB
PresharedKey = $CKEY_PSK
AllowedIPs = $1, ::0/0
EOF
}

touch "$CLIENT_CONF_SPLIT" "$CLIENT_CONF_FULL"
chmod 600 "$CLIENT_CONF_SPLIT" "$CLIENT_CONF_FULL"
write_client_conf $IP_SUBNET.0/24 "$CLIENT_CONF_SPLIT"
write_client_conf 0.0.0.0/0 "$CLIENT_CONF_FULL"



# add the client peer to the server configuration
```

```
cat << EOF >> "$SERVER_CONF"

[Peer]
# $CLIENT_NAME
PublicKey = $CKEY_PUB
PresharedKey = $CKEY_PSK
AllowedIPs = $IP_SUBNET.$IP_INDEX/24, $IP_SUBNETv6:$IP_INDEX/64
EOF



echo "If you have qrencode installed, you can issue the following to configure"
echo "the client:"
echo "    qrencode -t ansiutf8 -r $CLIENT_CONF_SPLIT"
echo "    qrencode -t ansiutf8 -r $CLIENT_CONF_FULL"
```

Version #3
Erstellt: 2021-10-27 14:33:29 UTC von Hiajen
Zuletzt aktualisiert: 2023-01-17 11:23:45 UTC von Hiajen