OpenVPN Server Administration

Installation - manuell

Basisinstallation

Ausgangssituation: Blanke Debian 10 Installation mit SSH-only und sudo

Notwendige Pakete installieren:

```
sudo apt update
sudo apt install openvpn
# Schon vorhanden: iptables openssl ca-certificates
```

IPv4 Forwarding aktivieren

```
echo "1" > /proc/sys/net/ipv4/ip_forward
```

in `/etc/sysctl.conf` diese Zeile auskommentieren, um das permanent zu schalten:

```
net.ipv4.ip_forward = 1
```

Anlegen eines VPN Bereiches

folgt generell: https://wiki.debian.org/OpenVPN

Für jeden Bereich ein gesondertes easy-rsa Verzeichnis: `/etc/openvpn/easyrsa_BEREICH/`:

```
cd /etc/openvpn
make-cadir easyrsa_BEREICH/

vim /etc/openvpn/easyrsa_BEREICH/vars # Edit variables

cd easyrsa_BEREICH
./easyrsa init-pki

./easyrsa build-ca / hier ca-key-pass vergeben

./easyrsa build-server-full server_BEREICH ## erst server-key-pass vergeben, dann ca-key-pass eingeben für Signierung

./easyrsa gen-dh

echo "$server-key-pass" > pki/private/server_BEREICH.pass
chmod go-rwx pki/private/server_BEREICH.pass
```

```
touch /etc/openvpn/server/ipp_BEREICH.txt
```

Jetzt in `/etc/openvpn/server` configfile `server_BEREICH.conf` anlegen:

```
local 10.4.110.18 # Lokale listen-IP des Servers
port 1194 # unique port pro Bereich wählen!
proto udp
dev tun # muss pro Bereich durchnummeriert werden?

ca      /etc/openvpn/easyrsa_BEREICH/pki/ca.crt
cert    /etc/openvpn/easyrsa_BEREICH/pki/issued/server_BEREICH.crt
key     /etc/openvpn/easyrsa_BEREICH/pki/private/server_BEREICH.key
dh      /etc/openvpn/easyrsa_BEREICH/pki/dh.pem

auth SHA512
cipher AES-256-CBC

server 10.8.1.0 255.255.255.0 # für clients zu vergebender IP-Bereich
topology subnet

push "route 10.4.0.0 255.255.0.0" # Route die gepusht werden soll via VPN
push "dhcp-option DNS 10.4.110.21" # DNS Server
push "dhcp-option DOMAIN drkb4.drk-cannstatt.de" # local domain

ifconfig-pool-persist ipp_BEREICH.txt #

keepalive 10 120

persist-key
persist-tun

user nobody
group nogroup

status log/openvpn_BEREICH-status.log

verb 3

explicit-exit-notify

askpass /etc/openvpn/easyrsa_BEREICH/pki/private/server_BEREICH.pass # Der
Server braucht bei jedem Start das server-key-pass und liest es hier ein
```

Server starten und registrieren mit:

```
systemctl start openvpn-server@server_BEREICH
systemctl enable openvpn-server@server_BEREICH
```

CoreFirewall: IP → Firewall → NAT

- Add dst-nat:
  - DST-Address 46.189.75.134
  - Protocol udp
  - DST-port <PORT>
  - in Interface ether3
  - Action dest-nat
  - Log_Prefix ovpn_
  - ToAddresses: <VPN Server IP>

IP → Routes

- add Route:
  - 10.8.x.0/24
  - gateway <VPN Server IP>

IP → Firewall → Filter Rules

- Chain dst_Server, add Rule:
  - Dst.Address: <VPN Server IP>
  - Protocol: udp
  - Dst. Port: <Server PORT>
  - Action: accept

IP → Firewall → Address Lists:

- Add AddressList
  - Name: range_VPN<BEREICH>
  - Address: 10.8.x.0/24

Dann für internes Routing, Weiterleitungs-Regeln entsprechend

https://linuxhint.com/enable_ip_forwarding_ipv4_debian_linux/

MFA einrichten

| oath_material.sh

```
#!/bin/sh
#
# Sample script to verify MFA using oath-tool

passfile=$1

# Get the user/pass from the tmp file
user=$(head -1 $passfile)
pass=$(tail -1 $passfile)


# Find the entry in our oath.secrets file, ignore case
#secret=$(grep -i -m 1 "$user:" /etc/openvpn/server/material.secrets |
cut -d: -f2)
```

```
secretline=$(grep -i -m 1 "$user:"
/etc/openvpn/server/material.secrets)
secret=$(echo "$secretline" | cut -d: -f2)

# Calculate the code we should expect
#code=$(oathtool --totp $secret)
code=`oathtool --totp $secret`

if [ "$code" = "$pass" ];
then
        exit 0
fi

# See if we have password and MFA, or just MFA

echo "$pass" | grep -q -i :

if [ $? -eq 0 ];
then
        realpass=$(echo "$pass" | cut -d: -f1)
        mfatoken=$(echo "$pass" | cut -d: -f2)

        # put code here to verify $realpass, the code below the if
validates $mfatoken or $pass if false
        # exit 0 if the password is correct, the exit below will deny
access otherwise
fi

# If we make it here, auth hasn't succeeded, don't grant access
exit 1
```

oath_material.sh und material.secrets müssen lesbar sein von nobody:nogroup. z.B. so:

```
-rwxr----- 1 root nogroup   40 Feb 18 18:57 material.secrets
-rwxr-x--- 1 root nogroup  999 Feb 18 19:36 oath_material.sh
```

in server.conf einfügen:

```
script-security 2
auth-user-pass-verify /etc/openvpn/server/oath_material.sh via-file
```

in client.conf einfügen:

```
auth-user-pass
```

User hinzufügen

```
cd /etc/openvpn/easyrsa_BEREICH
```

```
./easyrsa build-client-full CLIENTNAME          # mit User-Passwort
./easyrsa build-client-full CLIENTNAME nopass   # kein User-Passwort
```

für Erstellung ovpn config benötigt:

- pki/ca.crt
- pki/issued/clientname.crt
- pki/private/clientname.key

clientname.ovpn

```
client
dev tun
proto udp
remote 46.189.75.134 1195
resolv-retry infinite
nobind
persist-key
persist-tun
auth SHA512
cipher AES-256-CBC
verb 3
remote-cert-tls server
auth-user-pass
auth-nocache
<ca>
[...]
</ca>
<cert>
[...]
</cert>
<key>
[...]
</key>
```

verschlüsseltes zip erstellen:

```
zip -e clientname.zip clientname/
```

Eventuell noch MFA Code und Secret erstellen mit Script:

```
./oath-secret-gen.sh USERNAME
```

Dann Secret in /etc/openvpn/server/BEREICH.secrets eintragen, Code an User geben

Weitere Info

https://openvpn.net/diy-mfa-setup-community-edition/

Installation - Skript (alt)

Installscripte sind auf GitHub geforkt:
https://github.com/pjotre42/openvpn-install

Zum installieren:

1.: Ordner `/home/rohmannp/openvpn-install` anlegen

update_script.sh

```
#!/bin/bash
cd /home/rohmannp/openvpn-install
rm openvpn-install.sh
wget https://raw.githubusercontent.com/pjotre42/openvpn-
install/master/openvpn-install.sh
chmod 0700 openvpn-install.sh
```

Dann `openvpn-install.sh` via sudo ausführen

Es sollte der interne DNS Resolver verwendet werden (oder hardkodiert IP des lokalen DNS)

CA/Server Zertifikate und erster Nutzer werden automatisch angelegt. Dieser User hat kein Passwort

Ergebnis sollte so ähnlich aussehen:

```
Finished!
Your client configuration is available at: /root/<client_name>.ovpn
If you want to add more clients, just run this script again!
```

Note: Auf Raspi wird nicht /home/rohmannp/openvpn-install/ sondern /home/drkadmin/openvpn-install/ verwendet!

Konfiguration (alt)

Server wird vorkonfiguriert installiert

Config liegt in: `/etc/openvpn/server/server.conf`

Standardport: 1194

Userverwaltung (alt)

Um neue User anzulegen, und bestehende User zu löschen:
`openvpn-install.sh` via sudo ausführen
–> Textbasierte Menüführung

Netzwerkeinstellungen

Firewall & NAT Forwarding

Port 1194 (UDP) weiterleiten auf OpenVPN Server IP

NAT Traversal: TODO

Lokale IP konfig: TODO

Bestehende VPN Zugänge

- Wasenwache (via REN - 10.4.110.23)
- Ulmer Straße (via PI - im Aufbau)

Server-Admin

WaWa-Admin

start/stop/restart/status:

```
sudo systemctl <ACTION> openvpn-server@server.service
```

Server-Config

WaWa-Admin (alt)

On VM REN

server.conf

```
local 10.4.110.23
port 1194
proto udp
dev tun
```

```
ca ca.crt
cert server.crt
key server.key
dh dh.pem
auth SHA512
tls-crypt tc.key
topology subnet
server 10.8.0.0 255.255.255.0
# push "redirect-gateway def1 bypass-dhcp"
push "route 10.4.0.0 255.255.0.0"
ifconfig-pool-persist ipp.txt
push "dhcp-option DNS 10.4.110.11"
push "dhcp-option DNS 10.4.110.21"
push "dhcp-option DNS 10.4.110.1"
keepalive 10 120
cipher AES-256-CBC
user nobody
group nobody
persist-key
persist-tun
status openvpn-status.log
verb 3
crl-verify crl.pem
explicit-exit-notify
```

WaWa-Material

On VM ...

server.conf

```
```

Client-Config

WaWa-Admin (alt)

Beispiel .ovpn File

user.ovpn

```
client
dev tun
proto udp
```

```
remote 46.189.75.134 1194
resolv-retry infinite
nobind
persist-key
persist-tun
remote-cert-tls server
auth SHA512
cipher AES-256-CBC
ignore-unknown-option block-outside-dns
block-outside-dns
verb 3
<ca>
-----BEGIN CERTIFICATE-----
[...]
-----END CERTIFICATE-----
</ca>
<cert>
-----BEGIN CERTIFICATE-----
[...]
-----END CERTIFICATE-----
</cert>
<key>
-----BEGIN ENCRYPTED PRIVATE KEY-----
[...]
-----END ENCRYPTED PRIVATE KEY-----
</key>
<tls-crypt>
-----BEGIN OpenVPN Static key V1-----
[...]
-----END OpenVPN Static key V1-----
</tls-crypt>
```

openVPN-GUI

UbuntuNetwork Manager

Anlegen mittels New connection → VPN (openvpn)

Vorbereitung: .ovpn Datei in einzelne Dateien ca.crt, cert.crt, private.key, tls-crypt.key aufsplitten

- General Configuration
- VPN (openvpn)
  - Gateway: 46.189.75.134
  - Connection Type: Password with Certifcates (TLS)
  - CA Certificate: ca.crt
  - User Certificate: cert.crt
  - Private Key: private.key
  - Private Key Password: Ask for this password every time

    - Username: <username (xyz@drk-cannstatt.de)>
    - Password: Ask for this password every time
    - Advanced
        - General
            - Use custom gateway port: 1194
            - Set virtual device type: TUN
            - All other options are off
        - Security
            - Cipher: AES-256-CBC
            - HMAC Authentication: SHA-512
        - TLS Settings
            - Server certificate check: Don't verify certificate identification
            - Mode: TLS-Crypt
            - Keyfile: tls-crypt.key
        - Proxy
            - Proxy Type: not required
- IPv4: Keep standard
- IPv6: Keep standard

Bei Verbindung muss 2x das selbe User-Passwort eingegeben werden