VPNs einrichten und benutzen

Tübix 2019

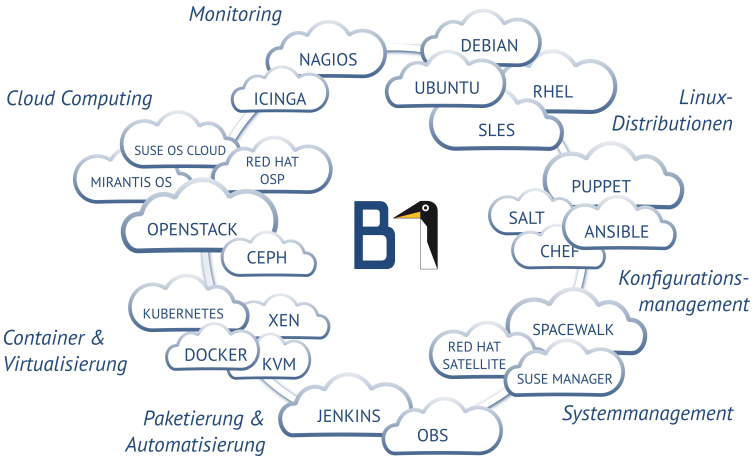
6. Juli 2019

Tilman Kranz
Linux Consultant & Trainer
B1 Systems GmbH
kranz@b1-systems.de

Vorstellung B1 Systems

- gegründet 2004
- primär Linux/Open Source-Themen
- national & international tätig
- über 100 Mitarbeiter
- unabhängig von Soft- und Hardware-Herstellern
- Leistungsangebot:
  - Beratung & Consulting
  - Support
  - Entwicklung
  - Training
  - Betrieb
  - Lösungen
- Standorte in Rockolding, Köln, Berlin & Dresden

Schwerpunkte



VPNs einrichten und verwenden

Mögliche Gründe für VPN

- Zusammenfassen physikalisch getrennter Netze
- Absichern des Datenverkehrs
- Privatsphäre
- Provider-seitiges Port- und Geoblocking
- IPv4 an IPv6-only-Zugängen
- ...

OpenVPN – Übersicht

- Clients und Server für Linux, Windows, Mac, pfSense, ...
- Clients (mit GUI) für Android, IOS, ...
- starke Verschlüsselung (z. B. AES-256-CBC)
- TLS und PKI für Zugangsmanagement
- zwei Betriebsmodi: *Tunneling* und *Bridging*

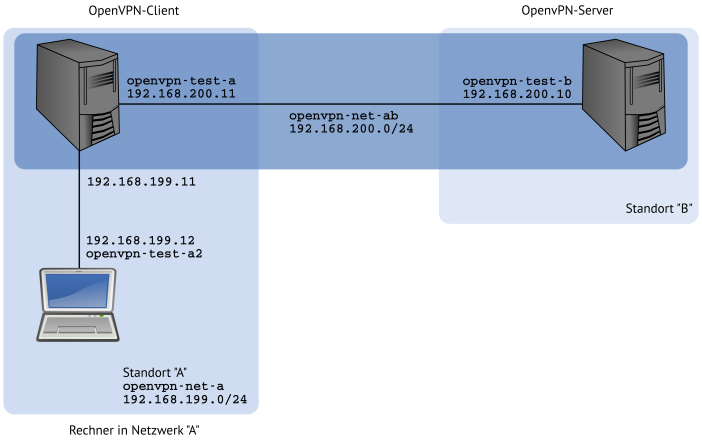
OpenVPN – Tunneling

- Layer-3-Verbindungen (TUN-Devices)
- IP-only; dediziertes IP-Subnetz
- OpenVPN-Server vergibt IP-Adressen aus Pool
- optional: Client-to-Client-Sichtbarkeit
- optional: Netzwerke mit Routing verbinden (Layer 3)

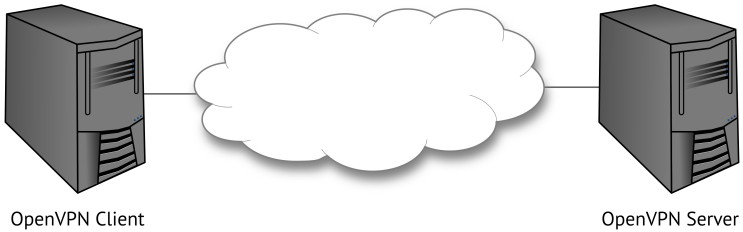
OpenVPN – Bridging

- Layer-2-Verbindungen (TAP-Devices)
- erfordert Bridge auf dem Server (`brctl`)
- beliebige Protokolle (IP, NetBEUI, AppleTalk, ...)

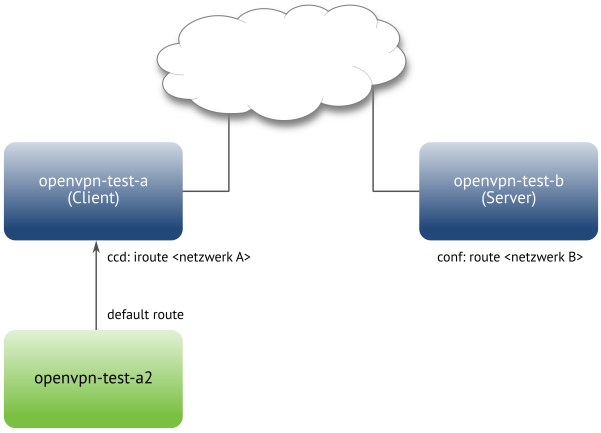
Beispiel: Routing, topology=subnet



Beispiel: Vereinfachte Anwendung



Beispiel: Konfiguration



OpenVPN & easy-rsa installieren

Installation

```
# apt -y install easy-rsa openvpn
```

easy-rsa einrichten

Installation

```
# make-cadir openvpn-ca
# cd openvpn-ca
# cp openssl-1.0.0.cnf openssl.cnf
# vim vars
# cd ..
```

Server: Diffie-Hellman-Parameter erzeugen

Diffie-Hellman-Parameter erzeugen

```
# openssl dhparam -out \
  /etc/openvpn/server/dh2048.pem \
  2048
```

Wichtig

DH-Parameter mit mindestens 2048 Bit erstellen!

Server: TLS-Key erzeugen

TLS-Key erzeugen

```
# openvpn --genkey --secret tls-crypt.key
```

Server-Key erzeugen

Server-Key erzeugen

```
# . vars
# ./pki-tool --server <SERVER-FQDN>
```

Certs & Key „inline“ (Client & Server)

Certs & Key „inline“

```
<ca>
...Inhalt von openvpn-ca/keys/ca.crt...
</ca>
<cert>
...Inhalt von openvpn-ca/keys/<FQDN>.crt...
</cert>
<key>
...Inhalt von openvpn-ca/keys/<FQDN>.key...
</key>
<tls-crypt>
...Inhalt von tls-crypt.key...
<tls-crypt>
```

Server-Konfiguration

/etc/openvpn/server/my-server.conf

```
topology subnet
server 10.8.0.0 255.255.255.0
port 1194
dh dh2048.pem
client-config-dir my-server/ccd
dev tun
verb 3
...Certs und Keys inline...
```

Client-Key erstellen

Client-Key erstellen

```
. vars
./pki-tool <CLIENT-FQDN>
```

Client-Konfiguration

/etc/openvpn/client/client1.conf

```
remote my-server
client
port 1194
dev tun
verb 3
...Certs und Keys inline...
```

Verschlüsselungs-Optionen (minimal)

Verschlüsselungs-Optionen (minimal)

```
tls-version-min 1.0      # nur Server
ncp-ciphers AES-128-CBC  # nur Server
```

Verschlüsselungs-Optionen (empfohlen)

Verschlüsselungs-Optionen (minimal)

```
tls-cipher TLS-ECDHE-ECDSA-WITH-AES-256-GCM-SHA384
tls-version-min 1.2      # nur Server
                         # breaks cryptoapicert!
                         # breaks clients<=2.3.3!
ncp-ciphers AES-256-GCM  # nur Server
auth SHA512
<tls-crypt>
...
<tls-crypt>
```

Server: Client Configuration Directory

Client Configuration Directory

```
# mkdir -p /etc/openvpn/server/my-server/ccd

# /etc/openvpn/server/my-server/ccd/<CLIENT-NAME>:
ifconfig-push <CLIENT-VPN-IP> <VPN-NETMASK>
iroute <CLIENT-NETWORK> <CLIENT-NETWORK-MASK>
```

Start mit systemd (Server)

Start mit systemd

```
# systemctl start openvpn-server@my-server
```

Start mit systemd (Client)

Start mit systemd

```
# systemctl start openvpn-client@client1
```

Alternativen

- IPSEC
- Wireguard
- Softether
- SOCKS

Vielen Dank für Ihre Aufmerksamkeit!

Bei weiteren Fragen wenden Sie sich bitte an info@b1-systems.de oder +49 (0)8457 - 931096