VPNs mit OpenVPN

von Michael Hartmann
<michael.hartmann@as-netz.de>

```
estel@box: /home/estel
root@OpenWrt:/etc/openvpn# ifconfig tap
ifconfig: tap: error fetching interface information: Device not found

root@OpenWrt:/etc/openvpn# ifconfig tap0
tap0      Link encap:Ethernet  HWaddr 00:FF:0B:19:63:E4
          inet addr:192.168.10.2  Bcast:192.168.10.255  Mask:255.255.255.0
          inet6 addr: fe80::2ff:bff:fe19:63e4/10 Scope:Link
          inet6 addr: 2001:8e0:abcd:5c::1/128 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:20095 errors:0 dropped:0 overruns:0 frame:0
          TX packets:21216 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:13431881 (12.8 MiB)  TX bytes:2401354 (2.2 MiB)

root@OpenWrt:/etc/openvpn#
```

Allgemeines

Was ist ein VPN?

- VPN: Virtual Privat Network (virtuelles, privates Netzwerk)
- Tunnel zwischen zwei Rechnern durch ein (unsicheres) Medium (z.B. Internet)
- Privat bedeutet nicht (zwingend) sicher/verschlüsselt!

Wofür ist es nützlich?

- Verbindung zwischen zwei internen Netzwerken (bzw. Rechnern)
- keine Standleitung (bzw. andere Direktverbindung) notwendig
- sicherer Tunnel durch ein unsicheres Medium (z.B. bei WLAN)

Vorteile von OpenVPN:

- einfach zu konfigurieren
- Verschlüsselung mit guten Authentifizierungsmöglichkeiten
- Kompression
- kein Kernelpatch notwendig / Userspace-Programm
- Plattformunabhängig (Windows, Linux, *BSD, Mac OS X, Solaris u.a.)

Vergleich mit Packetzustellung

bei Netzwerken:

Problem:

- keine öffentliche IP-Adresse (NAT) (zu mindest keine IPv4-Adresse)
- trotzdem Wunsch (sicher) auf interne Dienste zuzugreifen (z.B. Samba-Sever)

Lösung:

- Aufbau eines Tunnels
- Daten werden durch sicheren Tunnel geschickt
- an VPN-Endpunkten können Packete weitergeroutet werden

im RL (wirklichen Leben):

Problem:

- Brief an Person ohne eigene Wohnung (z.B. WG-Bewohnerin (-;)
- trotzdem Wunsch einen Brief (also ein Datenpacket) direkt zu verschicken (ohne Postadresse nicht möglich)

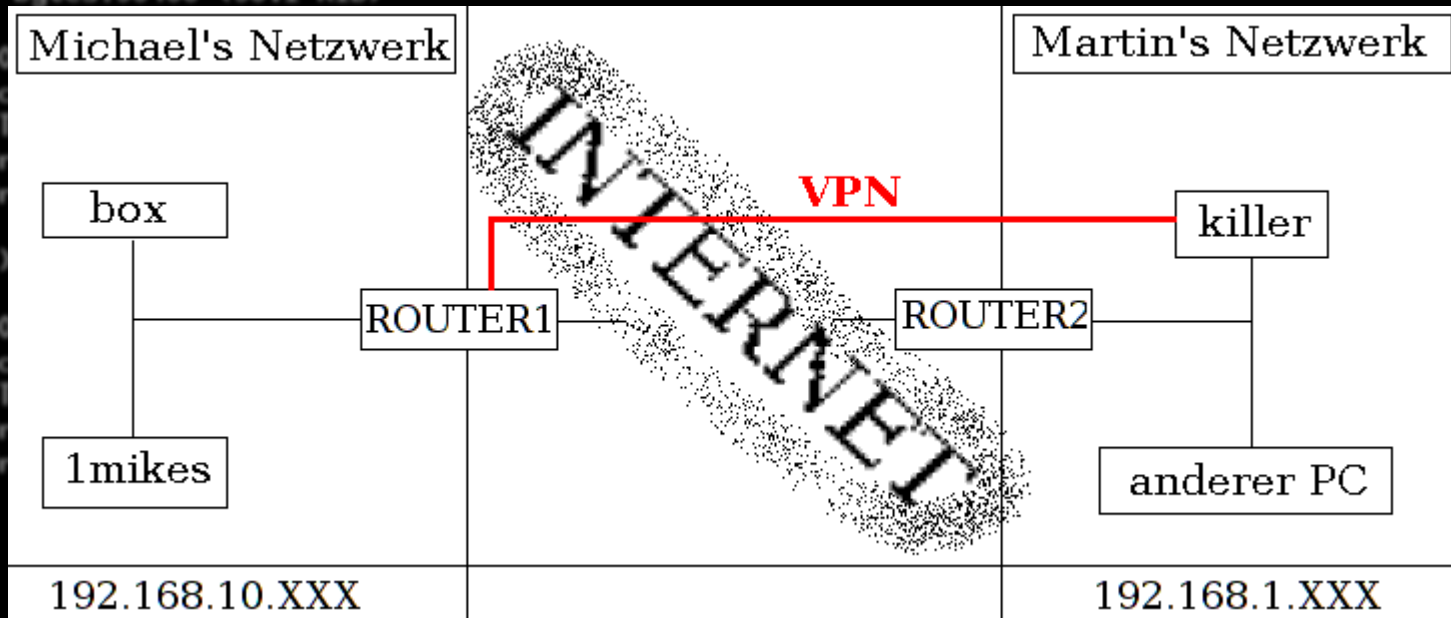
Lösung:

- Paket adressiert an Wohnungsbesitzer
- Paket enthält weiteres Packet adressiert an eigentliche(n) Empfänger(in)
- das 'Paket im Paket' ist verschlüsselt (z.B. doppelt oder vierfach rot13 (-;)

=> Pakete können jeweils 'direkt' verschickt werden (keine Probleme mit NAT)

Prinzip

- Tunnel wird zwischen Router1 und killer aufgebaut
- Router1 und killer müssen unterschiedliche Adressbereichen zugehören, da sonst Routing uneindeutig wird (Router1: 192.168.10.1; killer: 192.168.1.6)
- killer und anderer PC brauchen speziellen route-Eintrag, da killer in Martin's Netzwerk nicht die Standardroute ist und die Packete an Router2 verschicken würde
- in Michael's Netzwerk benötigt lediglich Router1 einen zusätzlichen Routing-Eintrag



Vorbereitungen

Voraussetzungen:

- zwei Rechner, von denen mindestens bei einem Port 5000 (oder anderer) vom Internet aus erreichbar ist (notfalls Port-Forwarding; Firewall!)
- statische Adresse (entweder IP oder z.B. dyndns)
- Systeme, die OpenVPN unterstützen (also Linux (-;)
- Kernel mit tun/tap-Support (modprobe tun)

Vorbereitungen:

- Port-Forwarding und Firewall ggf. konfigurieren (Port 5000)
- ggf. dyndns-Account einrichten (dyndns.org; Tool: z.B. inadyn)
- OpenVPN installieren (`apt-get install openvpn` unter Debian)
- tun-Modul laden (`modprobe tun`)
- Schlüssel erzeugen: `openvpn --genkey --secret static.key`
- Schlüssel sicher auf anderen Rechner übertragen (z.B. `scp`)

Konfiguration

router1: /etc/openvpn/openvpn.conf

```
# remote - zu diesem rechner verbinden wir
remote maaaa.dyndns.org

# schnittstelle sei /dev/net/tun*
dev tun

# 192.168.10.2: Michael's Endpunkt
# 192.168.1.6: Martin's Endpunkt (killer)
ifconfig 192.168.10.2 192.168.1.6

# datei mit dem key
secret /etc/openvpn/static.key

# verbindung aufrecht erhalten
ping 20
ping-restart 45
ping-timer-rem
persist-tun
persist-key

# wir wollen LZO kompression verwenden.
# cpu-lastiger aber schneller
comp-lzo

# evtl. Port hier ändern, wenn benötigt
#port 5000

# script, das nach dem verbinden ausge-
# führt wird (zum anlegen der routen)
up /etc/openvpn/route.up

# wir wollen normales log verhalten.
# geloggt wird in/var/log/syslog
verb 3
```

killer: /etc/openvpn/openvpn.conf

```
# remote - zu diesem rechner verbinden wir
remote estel.dyndns.org

# schnittstelle sei /dev/net/tun*
dev tun

# 192.168.1.6: Martin's Endpunkt (killer)
# 192.168.10.2: Michael's Endpunkt (wrt54gs-router)
ifconfig 192.168.1.6 192.168.10.2

# datei mit dem key
secret /etc/openvpn/static.key

# verbindung aufrecht erhalten
ping 20
ping-restart 45
ping-timer-rem
persist-tun
persist-key

# wir wollen LZO kompression verwenden.
# cpu-lastiger aber schneller
comp-lzo

# evtl. Port hier ändern, wenn benötigt
#port 5000

# script, das nach dem verbinden ausge-
# führt wird (zum anlegen der routen)
up /etc/openvpn/route.up

# wir wollen normales log verhalten.
# geloggt wird in/var/log/syslog
verb 3
```

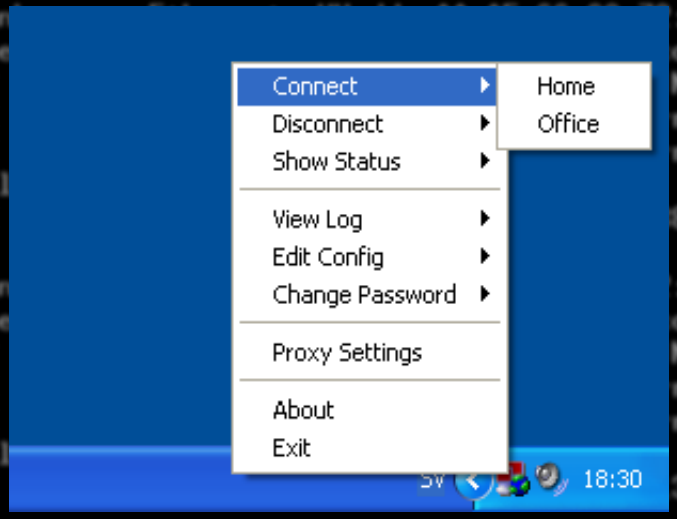
zusätzliche Konfiguration

- route.up:
  router1: `route add -net 192.168.1.0 netmask 255.255.255.0 gw 192.168.1.6`
  killer: `route add -net 192.168.10.0 netmask 255.255.255.0 gw 192.168.10.2`
- IP-Forwarding: `echo "1" > /proc/sys/net/ipv4/ip_forward`
  und evtl. sonstige Firewallregeln anwenden
- OpenVPN starten: - `openvpn --daemon --config /etc/openvpn/openvpn.conf`
  - `/etc/init.d/openvpn start`
  (alle Konfigurationsdateien in /etc/openvpn/ werden abgearbeitet)
- bei Fehlern: - Sniffer (z.B. `netcat`, `ethereal`)
  - Prüfen der Routingtabellen
  - Prüfen der Standardroute
  - `ping -I tun0 <ip>`
  - `ifconfig` (u.a: `route`, `ip` usw.)

Ausblick

weitere Features von OpenVPN:

- weitere Authentifizierungsmöglichkeiten
- OpenVPN-Server
- tap-Device: vollständiges Tunneln von Ethernet-Frames (Layer 2) (z.B. für den Einsatz von IPv6 oder IPX über den Tunnel)
- load-balanced VPN-Server
- GUI für Windows und Mac OS X



Autor

Michael Hartmann (<michael.hartmann@as-netz.de>)

Bei Fragen/Anregungen/Kritik/Lob/Fehler einfach eine e-Mail zu schicken.