

LUG Ahaus



<http://lugah.de>

# WLAN-Sicherheit mit OpenVPN

## WLAN-Sicherheit mit OpenVPN

Linux User Group Ahaus

Jul. 04

Rainer Ostendorf  
[rainer@lugah.de](mailto:rainer@lugah.de)



# WLAN-Sicherheit mit OpenVPN

## Warum ein VPN fürs WLAN?

- Die meisten WLANs sind unverschlüsselt oder setzen WEP ein
- WEP ist als Stromchiffrierer grundsätzlich problematisch
- WEP im Speziellen ist angreifbar, es genügt ausreichend viele Daten mitzuhören
- “Wer sollte schon mein LAN angreifen?” - Mentalität
- Public HotSpots sind 'ne gute Sache, aber bitteschön kontrolliert.



# WLAN-Sicherheit mit OpenVPN

## Warum OpenVPN?

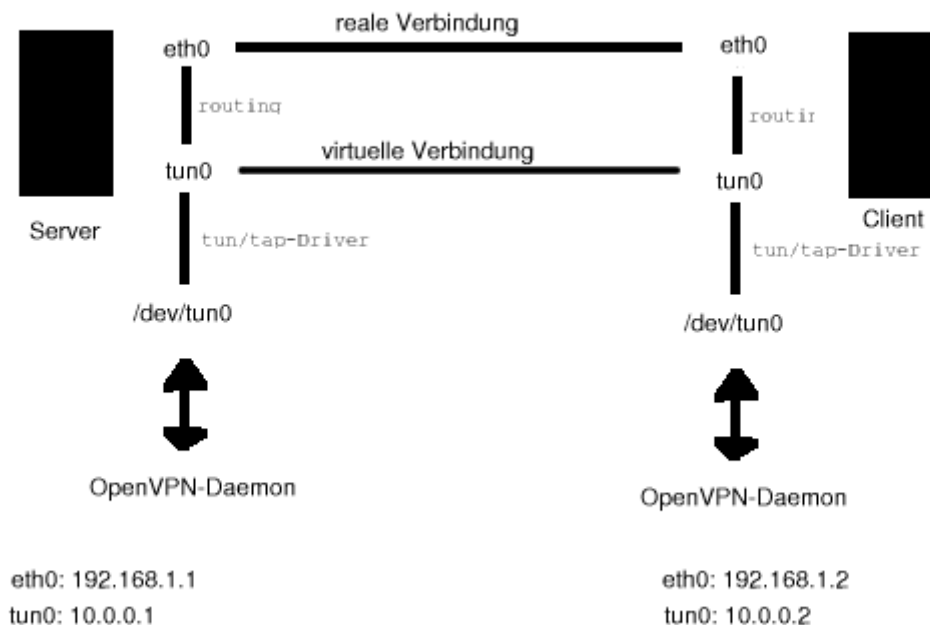
- OpenVPN läuft im Userspace über das tun/tap-Device → kein Kernel patchen/kompilieren erforderlich
- OpenVPN nutzt bewährte OpenSSL-Technik und gilt als sichere Lösung für VPNs
- OpenVPN ist für sehr viele Plattformen verfügbar, u.a. Linux, Windows 2k/XP, MacOS, Free-/Net-/OpenBSD, Solaris
- IPsec ist sicher mächtiger, aber auch komplexer. Wurde mit Kernel 2.6 einfacher, dank IKE im Kernel.



# WLAN-Sicherheit mit OpenVPN

## Wie funktioniert?

- OpenVPN benutzt das tun/tap-Device des Linux-Kernels. Als Modul meist verfügbar, ansonsten schnell hinzugefügt.
- Das tun-Device bildet eine Schnittstelle zwischen UserSpace und Kernel-IP-Stack
- Programme können im Userspace arbeiten und ihre Kommunikation über `/dev/net/tun` abwickeln:



Jedes Packet, dass der Kernel an tun0 sendet, kann aus `/dev/net/tun` gelesen werden – und umgekehrt.



# WLAN-Sicherheit mit OpenVPN

## Wie installieren?

- Tun/Tap-Modul laden:

```
# modprobe tun
```

- Testen ob's geladen ist kann man mit:

```
# lsmod | grep tun  
tun 4512 0 (unused)
```

- OpenVPN herunterladen:

- Bei vielen Distros schon dabei, bei Debian z.B. reicht ein “apt-get install openvpn”
- Sonst: Source laden und kompilieren:

```
# wget http://cesnet.dl.sourceforge.net/sourceforge/openvpn/openvpn-  
1.5.0.tar.gz
```

```
# tar xfvz openvpn-1.5.0.tar.gz  
# ./configure --enable-pthread  
# make  
# make install
```



# WLAN-Sicherheit mit OpenVPN

## Wie konfigurieren?

- PreShared-Key Variante - Das heisst: jeder Client bekommt einen geheimen Schlüssel vorab.
- Für zuhause völlig ausreichend, bei vielen Clients in Firmen Lösung über Zertifikate (X.509) mächtiger
- Szenario: Linux Server/Router mit DSL. An seperater Netzwerkkarte AP mit WLAN. Ein Laptop im WLAN unter Linux oder Windows 2k/XP.
- Router-Firewall öffnet nur UDP-Port an WLAN-Interface, auf dem virtuellen Interface ist alles erlaubt
- Tunnel-Endpunkte liegen in eigenem Subnetz, damit einfaches Routing möglich.



# WLAN-Sicherheit mit OpenVPN

## Client-Konfiguration, Vorbereitungen:

- geheimen 2048bit-Schlüssel erzeugen:

```
# mkdir /etc/openvpn
# cd /etc/openvpn
# openvpn --gen-key --secret openvpn.sec
```

- den Schlüssel **sicher** auf den Server kopieren (scp, Diskette, kein “plain” WLAN)
- Die Netzwerkverbindung muss stehen: ein

```
# ping 192.168.1.2
```

muss also funktionieren.



# WLAN-Sicherheit mit OpenVPN

## Client-Konfiguration, Konfigdatei:

- Konfigurationsdatei **/etc/openvpn/config** anlegen. Inhalt:

```
# /etc/openvpn/config
# Client Configuration for virtual private Network

# Das verwendete Tun-Device
dev tun0

# IP-Adresse vom Router, Tunnelende:
remote 192.168.1.2

# IP-Zuweisung: erste Eintrag ist lokal (Laptop)
#   zweiter ist entfernt (Router)
ifconfig 10.0.0.2 10.0.0.1

# Die Datei mit dem geheimen Schlüssel:
secret /etc/openvpn/openvpn.sec

# Maximale Paketlänge 1500, keine Fragmentierung, MS-Fix für Windows Rechner
tun-mtu 1500
fragment 1500
mssfix
```



# WLAN-Sicherheit mit OpenVPN

## Server-Konfiguration, Konfigdatei:

- Konfigurationsdatei **/etc/openvpn/config** wie auf Client, nur die Adressen beim “ifconfig”-Eintrag müssen getauscht werden:

```
# /etc/openvpn/config
# Server Configuration for virtual private Network

dev tun0
remote 192.168.1.2
ifconfig 10.0.0.1 10.0.0.2
secret /etc/openvpn/openvpn.sec
tun-mtu 1500
fragment 1500
mssfix
```



# WLAN-Sicherheit mit OpenVPN

## Inbetriebnahme:

- Den OpenVPN auf dem Server starten (Konsole):

```
# openvpn --config /etc/openvpn/config
```

- Auf dem Client dasselbe -> Der Tunnel wird aufgebaut
- Bei Erfolg - "Connection initiated" - sollte ein:

```
# ping 10.0.0.1
```

auf dem Laptop funktionieren.

- Jetzt muss evtl. noch die Default-Route geändert werden:

```
# route del default && route add default gw 10.0.0.1
```

- Natürlich muss das Routing beim Router eingeschaltet sein :)

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



# WLAN-Sicherheit mit OpenVPN

## Absicherung per Firewall:

- Auf dem WLAN Interface soll nur OpenVPN erlaubt sein, auf dem tun-Interface alles:

```
#!/bin/bash
# /etc/openvpn/setup.sh
# Firewall-Ruleset für den Server laden

#Forwarding aktivieren, falls noch nicht passiert
echo "1" > /proc/sys/net/ipv4/ip_forward

# eth0: Nur UDP auf Port 5000 reinlassen (VPN)
iptables -A INPUT -i eth0 -p udp --dport 5000 -j ACCEPT
iptables -A INPUT -i eth0 -j DROP

# eth0: Nur UDP auf Port 5000 senden
iptables -A OUTPUT -o eth0 -p udp --dport 5000 -j ACCEPT
iptables -A OUTPUT -o eth0 -j DROP

# Forwarding über eth0 ausschalten
iptables -A FORWARD -i eth0 -j DROP

#Kommunikation über Tunnel erlauben
iptables -A INPUT -i tun0 -j ACCEPT
iptables -A OUTPUT -o tun0 -j ACCEPT
iptables -A FORWARD -i tun0 -j ACCEPT

# OpenVPN als Daemon starten
openvpn --config /etc/openvpn/config --daemon
```

LUG Ahaus



<http://lugah.de>

# WLAN-Sicherheit mit OpenVPN

das wars.

Danke für eure Aufmerksamkeit :)

-> praktisches Ausprobieren