

VPN Zugang unter Linux einrichten

In dieser Anleitung wird beschrieben, wie man den VPN Zugang der Georg-Simon-Ohm Fachhochschule Nürnberg einrichtet. Diese Anleitung ist grundsätzlich distributionsunabhängig. Eine funktionierende und bereits eingerichtete WLAN Karte ist jedoch die Grundvoraussetzung. Die WLAN Karte sollte so eingestellt sein, dass die IP Adresse vom DHCP Server automatisch bezogen wird und die Verbindung zum OHMnet aufgebaut werden kann. Dafür könnte man z.B. den dhcpd Daemon nutzen. Desweiteren sollten auch grundlegende Kenntnisse der Kernelkonfiguration sowie Installation der Pakete vorhanden sein.

1. Konfiguration des Kernels

Im Kernel müssen folgende Optionen gesetzt sein. Selbstverständlich könnte man die nachfolgenden Optionen nicht als Module <M> kompilieren, sondern fest <*> in den Kernel einbauen. Jedoch sollte man hierbei beachten, dass man Module nach Bedarf laden kann und hiermit einfach mehr Flexibilität geboten wird.

```
Device Drivers --->
Network device support --->
<M> PPP (point-to-point protocol) support
<M> PPP support for async serial ports
<M> PPP MPPE compression (encryption) (EXPERIMENTAL)
Cryptographic options --->
<M> SHA1 digest algorithm
```

Nach dem der neue Kernel kompiliert und gestartet wurde, fährt man mit der Installation der notwendigen Softwarepakete fort.

2. Installation der notwendigen Pakete - PPP und PPTPCLIENT

Für den VPN Verbindungsaufbau, werden **ppp** und **pptpclient** Pakete benötigt. Diese müssen nun installiert werden, z.B. bei Gentoo mit dem Befehl:

```
# emerge -av ppp pptpclient
```

Hinweis: Es könnte durchaus vorkommen, dass die o.g. Pakete sich aufgrund fehlender Kernelparameter beschweren und weitere PPP-Kerneloptionen erfordern. Diese sind jedoch optional, d.h. sie können ausgewählt werden, müssen aber nicht. Die für VPN relevante Kerneloptionen wurden bereits oben genannt.

3. Konfiguration der PPP und PPTPCLIENT

Man erstellt mit einem beliebigen Editor (z.B. mit nano) eine Datei unter **/etc/ppp/peers/vpn-fh** mit folgendem Inhalt:

```
linkname vpn-fh
pty "pptp vpn1.cs.fh-nuernberg.de --nolaunchpppd"
name vpn_benutzerkennung
remotename vpn-fh
require-mschap-v2
file /etc/ppp/options.pptp
ipparam vpn-fh
```

Anstatt **vpn_benutzerkennung** sollte man seine eigene VPN Benutzerkennung eintragen, die man zuvor im Rechenzentrum erhalten hat.

Und als nächstes wird die Datei **/etc/ppp/chap-secrets** angelegt und bearbeitet. Der Inhalt sollte wie folgt aussehen:

```
vpn_benutzerkennung * vpn_passwort *
```

Anstatt **vpn_benutzerkennung** trägt man wieder seine Benutzerkennung ein, und statt **vpn_passwort** das dazugehörige Kennwort

Als letztes sollte die Datei **/etc/ppp/options.pptp** mit folgendem Inhalt angelegt werden. Hier sollte nichts geändert werden.

```
lock
noauth
nobsdcomp
nodeflate
require-mppe-128
```

4. VPN Verbindung aufbauen

Zuerst sollte man prüfen ob das virtuelle **net.ppp0** device existiert. Sollte es nicht der Fall sein, so erstellt man es mit den Befehlen:

```
# cd /etc/init.d
# ln -s net.lo net.ppp0
```

Jetzt sollte bereits alles soweit vorhanden sein, um eine VPN Verbindung aufbauen zu können.

Hinweis: Sollte man die Kerneloptionen als Module kompiliert haben, so muss man vor dem Verbindungsaufbau noch folgenden Befehl ausführen:

```
# mknod /dev/ppp c 108 0
```

Wenn man alle PPP Optionen im Kernel fest eingebaut hat <*>, überspringt man diesen Hinweis es einfach.

Jetzt muss man noch die Route festlegen:

```
# route add -host vpn1.cs.fh-nuernberg.de dev eth1
```

Bei **eth1** sollte der Kürzel der WLAN Karte eingetragen werden (bei mir heisst es eth1).

Mit dem nachfolgenden Befehl baut man den VPN Tunnel auf:

```
# pon vpn-fh
```

Hinweis: Wenn man sich eine detaillierte Ausgabe wünscht, um beispielsweise Fehler zu lokalisieren, so benutzt man:

```
# pon vpn-fh debug dump logfd 2 nodetach
```

Jetzt sollte man ca. 10 Sekunden abwarten bis die Autorisierung beim VPM Server erfolgreich war und die Verbindung tatsächlich besteht. Und dann leitet man noch die Route über das virtuelle **ppp0** Device:

```
# route add default dev ppp0
```

Hinweis: Um die Befehle nicht jedes Mal per Hand einzugeben, kann man den Vorgang mit Hilfe eines Skripts automatisieren. Und zwar legt man hierzu einen Skript mit dem Namen **wlanfh.sh** und mit dem Inhalt an:

```
#!/bin/bash
mknod /dev/ppp c 108 0
route add -host vpn1.cs.fh-nuernberg.de dev eth1
pon vpn-fh
sleep 10
route add default dev ppp0
fi
```

Danach sollte man nicht vergessen, den Skript noch ausführbar zu machen mit:

```
# chmod +x wlanfh.sh
```

Jetzt kann man die Verbindung aufbauen mit:

```
# /pfad_zum_skript/wlanfh.sh
```

Herzlichen Glückwunsch, Sie haben es geschafft!

Für Fragen bzw. Änderungswünsche stehe ich unter lomik@gmx.de zur Verfügung

Michael Loptev