

Nützliche Netzwerktools unter Linux

Alle nachfolgend beschriebenen Kommandos sind in den Manual Pages ausführlich beschrieben. Soweit angegeben, sollte zum Ausführen der vollständige Pfad eingegeben werden.

arp

Lesen und ändern des aktuellen ARP-Cache. Nicht benutzte Einträge werden nach 5 Minuten automatisch gelöscht. Änderungen im ARP-Cache sind ausschliesslich *root* vorbehalten.

Pfad: `/sbin/arp`

iptables

Zum Einstellen der Paketfilterung (Firewalling). Hierzu gibt es [spezielle Tutorials](#).

Pfad: `/usr/sbin/iptables`

netstat

Sehr vielseitiges Tool, u.a. zum Anzeigen der aktiven Netzwerkverbindungen.

Beispiele:

- Auflisten aller aktiven TCP-Verbindungen:

```
netstat -tanp
```

(**-tanp** steht für **tcp,all,numeric,processes**)

- Auflisten aller aktiven UDP-Verbindungen:

```
netstat -uanp
```

(**-uanp** steht für **udp,all,numeric,processes**)

Tipp: Informationen zum Zustand und Ablauf eines Prozesses liefert das Kommando **strace**:

```
strace -p PID
```

ping

Messen der RTT einer Strecke zu einer Netzwerkkomponente mittels ICMP.

Beispiel:

```
ping -c 10 -s 1000 141.22.26.1
```

Dabei wird mit **-c** die Zahl der Pakete und mit **-s** die Paketgrösse angegeben. (Optional.)

Tipp: Zur Durchführung von Messreihen kann man die Stellprogrammierung nutzen:

```
for packetSize in 50 100 1000
do
    ping -c 10 -s $packetSize 192.168.18.100
done
```

route

Lesen/Einstellen der Routingtabelle.

Ziele können Geräte (**-host**) oder Netze (**-net**) sein. Netze müssen als Adresse mit Netzmaske (Option **netmask**) oder in Prefix-Schreibweise (*adresse/prefix*, dabei ist *prefix* Zahl der Bits im Netzwerkanteil) angegeben werden. Beispiel:

```
/sbin/route add -net 192.112.0.0/16 gw 141.22.29.1
```

Pfad: `/sbin/route`

ssh

Login auf einem anderen Rechner über eine gesicherte Netzwerkverbindung mit einer Shell.
Beispiel:

```
ssh myuser@lab21
```

Sollen auf dem Zielrechner Programme mit GUIs (X-Funktionalität) gestartet werden, sollte das Display mit der Option **-X** auf den eigenen Rechner umgeleitet werden:

```
ssh -X myuser@lab21
```

Wichtige Hinweise: Der Login mit **ssh** auf einen Laborrechner ist nur innerhalb des Labornetzes möglich! Bei der Partition „RNP-Special“ ist die Passwortregelung zu beachten.

Fingerzeig: Zum Kopieren von Dateien und Verzeichnissen von/zu einem anderen Rechner über das Netz kann man den Befehl **scp** (= **Secure Copy**) ähnlich dem **cp**-Befehl einsetzen:

```
scp -r mydirX myfileY myuser@lab21:~/some-other-dir
```

socat

(= **SO**cket **CAT**). Sehr vielseitiges Werkzeug zum Verbinden zweier Sockets. Nützlich zum Testen bei der Netzwerkprogrammierung.

Syntax: `socat [options] socketadresse1 socketadresse2`

Als Socketadresse kann auch die Standardein- und -ausgabe (der Shell) mit '-' angegeben werden.

Einen TCP-Server mit Listening-Port 9999 und Ein-/Ausgabe auf der Shell startet man z.B. so:

```
socat tcp-listen:9999 -
```

Läuft der Server auf dem Host *lab21*, startet man den entsprechenden TCP-Client mit:

```
socat tcp:lab21:9999 -
```

Beispiel für einen TCP-Server wie oben, aber diesmal für mehrere Clients:

```
socat tcp-listen:9999,fork -
```

Ähnlich kann auch über UDP kommuniziert werden:

```
socat udp-listen:8888 -
```

Entsprechende Gegenstelle dazu:

```
socat udp:lab21:8888 -
```

Hinweise:

Mit der Option **-x** gibt **socat** die übertragenen Zeichen zusätzlich auf *stderr* aus. Das Format ist hier jedoch hexadezimal, sodass auch nicht-druckbare Zeichen sichtbar werden.

```
socat -x tcp:lab21:9999 -
```

Wird der Server abgebrochen, kann der benutzte Listening-Port nicht sofort wieder benutzt werden, sondern erst nach einer Wartezeit. (Dann Fehlermeldung "...Address already in use"). Um dies zu vermeiden, kann beim Serverstart die Option **reuseaddr** eingesetzt werden:

```
socat tcp-listen:9999,reuseaddr -
```

Ein animiertes Client/Server-Beispiel finden Sie [hier](#).

wireshark

Netzwerksniffer. Hierzu gibt es eine [Kurzanleitung](#) und eine [komplette Dokumentation](#).

Pfad: `/usr/bin/wireshark`