SSH:

SSH steht für Secure Shell und sollte "telnet" mittlerweile komplett abgelöst haben. Mit SSH kann man sich remote auf anderen Rechnern anmelden und dort arbeiten, auch ein X-Forward ist möglich.

ssh test@192.168.4.4
ssh -l test 192.168.4.4

Mit diesem Befehl meldet man sich als der Benutzer "test" auf dem Rechner 192.168.4.4 an. Bei der ersten Verbindung wird man gefragt ob der Host-Key des Remote-Rechners abgespeichert werden soll. Dieser Key wird in der Datei

~/.ssh/known_hosts

abgespeichert.

Sollte bei einem erneuten Verbindungsversuch zum Rechner 192.168.4.4 der Host-Key nicht mehr der selbe sein, wird die Verbindung unterbrochen.

ssh -X -l test 192.168.4.4
ssh -X test@192.168.4.4

Hiermit wird das X-Forwarding aktiviert, nun ist es möglich Grafische Programme auf dem Remote-Rechner zu starten und auf dem Lokalen Rechner anzuzeigen.

ssh -Y → X-Forwarting ohne Fehlerprüfung (schneller)
-x → Verhindert das X-Forwarding

SSH-Key's:

Es gibt die Möglichkeit einen SSH-Key zu erstellen um sich dann auf bestimmten Rechnern ohne Passwordabfrage anzumelden.

Auf dem Eigenen Rechner z.B.: als kalle folgenden Befehl eingeben:

ssh-keygen -t rsa -b 1024

Nun wurden 2 Dateien erstellt: der Private Key "***/home/kalle/.ssh/id_rsa***" und der Public-Key "***/home/kalle/.ssh/id_rsa.pub***". Der Public-Key muß nun auf den entfernten Rechner im Homeverzeichnis des Users als der man sich anmelden möchte, in die Datei ~/.ssh/authorized_keys geschrieben werden. z.B.:

ssh-copy-id -i /home/kalle/.ssh/id_rsa.pub klaus@192.168.4.4

Nun kann sich kalle auf dem Localen Rechner als klaus auf dem Rechner 192.168.4.4 ohne Passwort anmelden.

Konfiguration des SSH-Client's:

Konfigurationsdatei: ***/etc/ssh/ssh_config***

Hier können für jeden Server Konfigurationen hinterlegt werden. z.B.:

```
host 192.168.4.4
   port 222
   ForwardX11 yes
host 192.168.2.2
   port 22
```

Wenn nun "***ssh 192.168.4.4***" eingegeben wird, versucht der SSH-Client eine Verbindung auf port 222 aufzubauen und X-Forwarding zu aktivieren.
Für den Rechner 192.168.2.2 wird der Standardport verwendet.

Der SSH-Server:

Starten des SSH-Servers: ***/etc/init.d/ssh(d) start***

Konfigurationsdatei: ***/etc/ssh/sshd_config***

port 22	→ Der SSH-Server läuft auf Port 22
ListenAddress 192.168.1.1	→ Der SSH-Server läuft auf der Nezwerkkarte mit der IP 192.168.1.1
PermitRootLogin no	→ Root Darf sich nicht per ssh Anmelden
X11Forwarding yes	→ X-Forwarding ist aktiviert
UsePAM yes	→ SSH nutzt PAM zur Benutzerauthentifizierung

SSH-Tunnel:

Mit dem Befehl:

ssh -L 192.168.1.1:333:localhost:631 root@192.168.4.2

Wird der Dienst der auf dem Rechner 192.168.4.2 auf dem Port "localhost:631" läuft, auf den Port 333 auf meiner eigenen Adresse 192.168.1.1 getunnelt. Gebe ich nun im Browser "192.168.1.1:333" ein lande ich auf dem CUPS-Server des Rechners 192.168.4.2
Dieser Tunnel ist nur solange Aktiv, wie die SSH-Verbindung zu diesem Rechner besteht.

ssh -R 192.168.4.2:333:localhost:631 root@192.168.4.2

Nun kann über die Adresse: "192.168.4.2:333" auf den Cups-Server von meinem Rechner zugegriffen werden. (Hierzu muß beim Server in der /etc/ssh/sshd_config "GatewayPorts yes" eigetragen sein.)