Durchschlag

automatischer Tunnelbau für SSH

SSH ermöglicht sicheren Remotezugriff auf andere Rechner. Dies betrifft nicht nur den Zugang zu einem Terminal, sondern es können beliebige TCP Verbindungen *getunnelt* werden. Dies kann für Dienste auf Rechnern hinter einem Loginserver mit Boardmitteln transparent automatisiert werden.

tl;dr: Eintrag in ~/.ssh/config

```
Host *alias*
    User *user*
    HostName *remote.server*
    ProxyCommand ssh *user@login.server* -W "%h:%p"
```

Lemma

Sei *loginserver* die Maschine die für uns erreichbar ist, und wir haben hier die Benutzerkennung *username*. Der Zielrechner sei *remote*. Das Zielnetzwerk sei o.B.d.A. ti.rwth-aachen.de. Unsere SSH Konfigurationsdatei ist $HOME/.ssh/config. Der Zugriff auf den Loginserver über asymmetrische Schlüssel ist eingerichtet und funtioniert.

```
  local              .-,(  ),-.              ___          ___
  ___  _          .-(          )-.          |==|         |==|
 [___]|=|  ---->(    internet    )-------> |==| ------> |==|
 /::/|_|          '-(          ).-'         |___|        |___|
                     '-.(  ).-'          loginserver    remote
```

Schlüsselbrett

Wenn das automatische Einwählen auf den Loginserver nicht funktioniert, kann dies recht einfach einegrichtet werden:

```
user@local$ ssh-keygen
user@local$ ssh-copy-id *loginserver*.ti.rwth-aachen.de
```

Erster Befehl erzeugt zwei Schlüssel, einen privaten, welcher unbedingt geheim gehalten werden muss, und einen öffentlichen. Letzterer wird anderswo hinterlegt um sich Ausweisen zu können. ssh-keygen fragt bei der Erzeugung nach einer Passphrase. Dies ist kein Passwort welches über das Netzwerk geht, sondern dient zum Sichern des eigenen *privaten Schlüssels*. Diese sollte auf jeden Fall vergeben werden. Eine komfortable Verwaltung derselben leisten die meisten modernen Desktopumgebungen.

Konfiguration

Zuerst legen wir einen Eintrag in ~/.ssh/config für unser Zielsystem an: Hier ist ti ein durchaus lesbarer Aliasname für das Netz das wir erreichen wollen.

```
Host ti
    User *username*
    HostName *loginserver*.ti.rwth-aachen.de
    # IdentityFile ~/.ssh/id_rsa
    ProxyCommand none
    ForwardX11 no
    # RequestTTY no
    # ServerAliveInterval 15
    # ServerAliveCountMax 3
```

```
    # CheckHostIP yes
    # StrictHostKeyChecking yes
    PubkeyAuthentication yes
    # PreferredAuthentications publickey
    # KbdInteractiveAuthentication no
    PasswordAuthentication no
    # ConnectionAttempts 3
```

Wichtig ist hier einzig der Alias **ti** und die richtige Konfiguration für die Zugangsdaten (*username*,*loginserver* und Schlüssel). Der Alias *ti* wird in den weiteren Einträgen referenziert. Und dann brauchen wir Einträge für alle Rechner die über diesen Eintrag erreichbar sind:

```
Host alcor
    User *user*
    HostName alcor.ti.rwth-aachen.de
    ProxyCommand ssh ti -W "%h:%p"

Host *remote*
    User *username*
    HostName *remote*.ti.rwth-aachen.de
    ProxyCommand ssh ti -W "%h:%p"
```

Wird einer ältere Version von ssh verwendet kann für den ProxyCommand auch

```
    ProxyCommand ssh ti exec nc %h %p
```

verwendet werden. Andere mögliche Einträge können für

- Komprimierung
- Verschlüsselung

blala hinzugefügt werden. X11 geht auch (nicht auf dem Loginserver) und ist im Internet aber sehr, sehr sehr schnarchlangsam.

Test

Als erstes sollte verifiziert werden das das automatische Login funktioniert:

```
fred@jupiter$ ssh ti
   ____ ,_.
  |_   _‘_’
    | | | |
    |_| |_| loginserver

user@login$
```

Mit *exit* oder **Ctrl + D** kommt man wieder raus. Beim Einloggen wird ggfls. noch ein Banner angezeigt (welches auch anders aussehen kann). Es sollte nicht nach einem Passwort gefragt werden. Dann sollte jetzt auch jeder andere Rechner innerhalb der Ti erreichbar sein:

```
anke@saturn$ ssh -X nemausa
user@nemausa$ xeyes
```

et voilà, ebenfalls ohne Passwortabfrage. X11 Tunnel sollten auch funtionieren

Desktop

da hängt das Aussehen und die Funktionalität stark vom verwendetem Manager ab. XDG Kompatibel sind wahrscheinlich *.desktop* Dateien welche sich auch für entfernte (hier jetzt nicht im sinne von *gelöscht*, sondern *remote*) Anwendungen anlegen lassen. Wie z.Bsp:

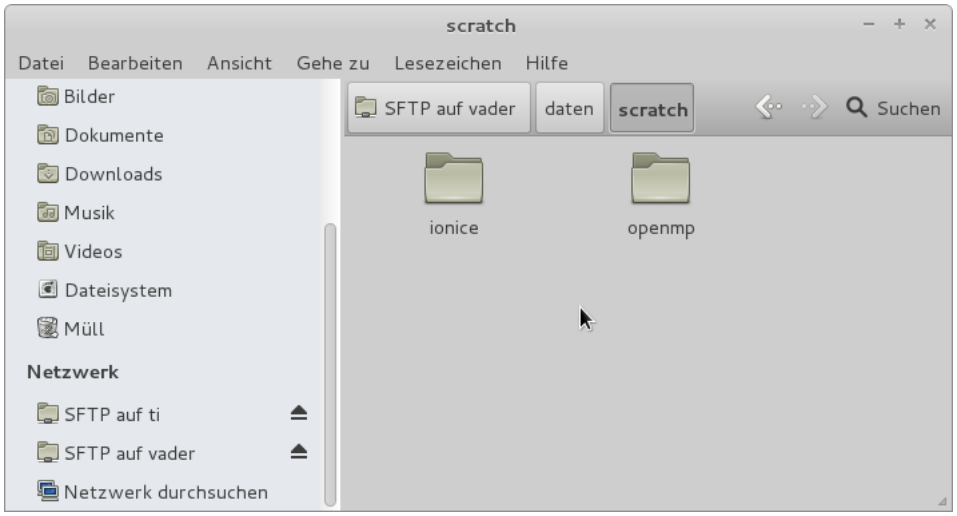
```
$ cat /usr/local/share/applications/matlab@nemausa.desktop
[Desktop Entry]
Version=1.0
Encoding=UTF-8
Name=Matlab on Nemausa
GenericName=Matlab@Nemausa
Comment=Run Matlab on a really powerfull box
Type=Application
Exec=ssh -fnX nemausa.ti.rwth-aachen.de /opt/matlab32R2010b/bin/matlab -desktop
#Terminal=false
Icon=matlab
Categories=Education;Science;
```

Eigene Starter landen zumeist in dem Verzeichnis:

```
~/.local/share/applications
```

Über **sshfs** lassen sich auch die entfernten (also, nicht die gelöschten, sondern die wo da nicht auf dem eigenen Rechner befindlichen) Dateisysteme einhängen. Zumeist wird hierbei auf die konfigurierten Einräge in der eigenen .ssh/config zurückgegriffen.

Es langt hier den ssh Eintrag anzugeben.



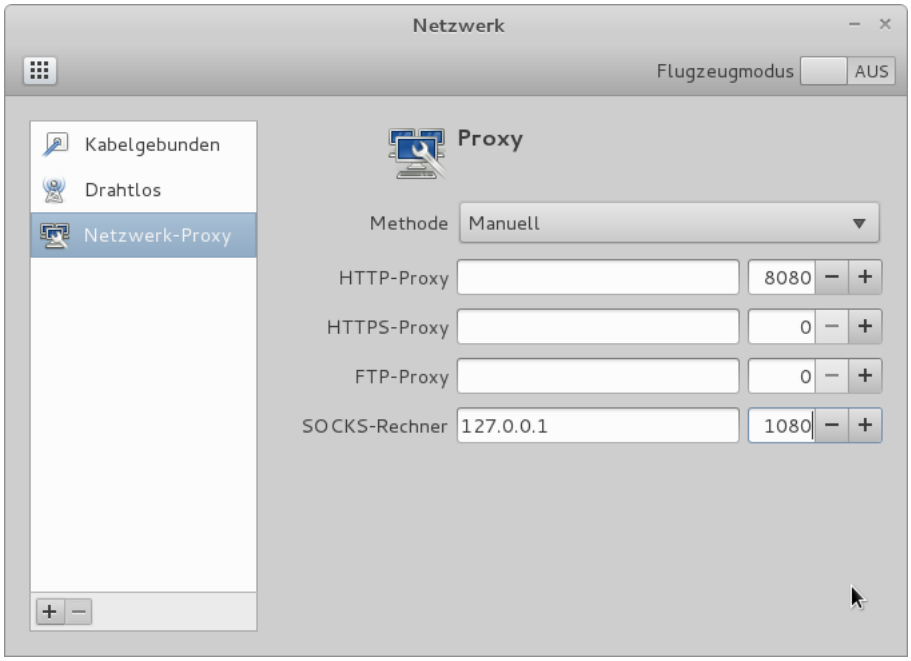
Socken-Vertreter

nun steht dieser Zugang nicht nur den *ssh* eigenen und abgeleiteten Diensten zur Verfügung, sondern kann auch anderen Programmen zur Verfügung gestellt werden. Soweit diese nur TCP Verbindungen benötigen und

mit einem *socks-proxy* zurechtkommen, reicht sowas wie:

```
kunta@kinte$ ssh -nNCS 127.0.0.1:1080 ti
```

und fortan können alle Programme wie Browser durch den Tunnel hindurch auf das Ti Netzwerk, und damit auch auf das gesamte RWTH-Netzt von *innen* zugreifen. (Einwickelskript findet sich auch hier: http://www.ti.rwth-aachen.de/~pflipsen/ssh-proxy)



Konfiguration ist durchaus verschieden. Einige Programme können auch systemweite Vorgaben verwenden.

Sollte eine Anwendung keinen weitern Bezug zu dieser Art von Kommunikationsanbahnung haben, kann diese *netztwerksmäßig* auch innerhalb des Ti Netzes ablaufen. (dies etwas eingeschränkt, Namensauflösung und UDP funkionieren nicht unbedingt) Hierzu dienen Einwickel- programme wie: tsocks, socksify, torsocks, Ringelsocken... Unter debian lässt sich sowas mit `sudo apt-get install tsocks` installieren. Dann zeigt:

```
tinki@winky$ tsocks curl http://tnx.nl/ip
```

die vermeintlich eigene IP Adresse an.