SSH Tunnel / Local Port Forwarding

Michael Nagel
michael.nagel@devzero.de

10. Mai 2013

„HILFE!!! Ich möchte von meinem Laptop auf einen MySQL-Server zugreifen, der aus Sicherheitsgründen nur lokale Verbindungen zulässt!!!“ *Typischer Hilfeschrei*

Jemand möchte von seinem Rechner auf einen bestimmten Port auf einem Server zugreifen. Eine Firewall blockiert den direkten Zugriff. Ein weiterer Rechner, der per SSH erreichbar ist, hat jedoch Zugriff auf den Port auf dem Server.

Lösung: SSH Tunnel / Local Port Forwarding

Lokales SSH Port Forwarding erlaubt es (unter anderem) über einen sogenannten Hopping Host Zugriff auf einen Server hinter einer Firewall zu erhalten.

Folgende Angaben sind dazu erforderlich:

- lokaler Host und lokaler Port – lokaler Host ist dort, wo SSH/putty läuft
- Hopping Host
- Ziel-Host (aus Sicht von Hopping Host (localhost)) und Ziel-Port

Beispiel: Jemand möchte von seinem Laptop aus auf eine Datenbank auf example.de zugreifen. Er hat SSH-Zugriff auf example.de.

Mit OpenSSH (Mac, Linux) sieht der Aufruf zur Tunnel-Erstellung folgendermaßen aus:

```
$ ssh login@example.de -L 9999:localhost:3306
```

Solange das Port-Forwarding aktiv ist, kommen Daten, die an „lokaler Host:lokaler Port“ gesendet werden, bei „Ziel-Host:Ziel-Port“ an. Statt „mysql -h example.de -P 3306“ kann jetzt „mysql -h localhost -P 9999“ verwendet werden, um sich mit dem Datenbank-Server zu verbinden und die Daten erreichen über den Tunnel ihr bislang unerreichbares Ziel.

Unter Windows mit PuTTY sind bei einer example.de-Session die folgenden Einstellungen nötig:

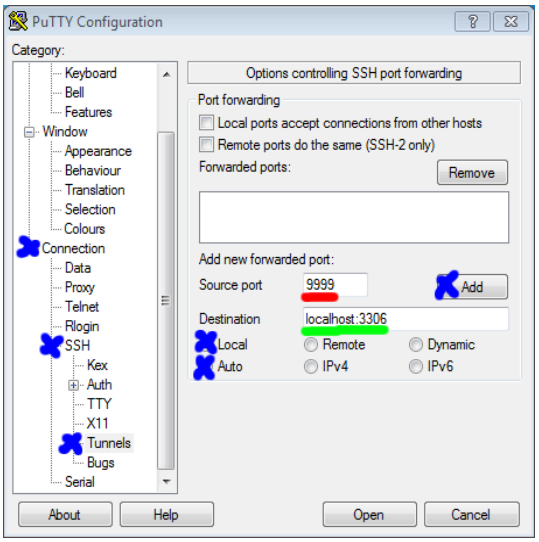

Abbildung 1: PuTTY Tunnel-Einstellungen