

sshuttle

„Transparent proxy server that works as a poor man's VPN. Forwards over ssh. Doesn't require admin. Works with Linux and MacOS. Supports DNS tunneling.“

– Quelle

Warum nicht ssh-Tunnel?

- Keine VPN-Funktionalität
- Wird nervig bei vielen Ports und Rechnern
- Syntax
`ssh -L 54322:ziel:22 user@remote`
- Lokale ports & localhost

Hintergrund / Geschichte

- Erdacht von Avery Pennarun
- Author von „Tunnel Vision“ (1998), „Fast Forward“, „Double Vision“ & Prototype für „Slipstream“
- sshuttle kombiniert deren Konzepte in einem Programm
→ <https://sshuttle.readthedocs.org/en/latest/trivia.html>
- Derzeitiger Maintainer: Brian May

Wie funktioniert sshuttle?

- Es funktioniert einfach.™
 - Kein VPN & keine Port-Forwardings : „It’s kind of both, and kind of neither.”
 - „Normally, a VPN forwards your data one packet at a time, and doesn’t care about individual connections; ie. it’s “stateless” with respect to the traffic. sshuttle is the opposite of stateless; it tracks every single connection.”
 - „sshuttle assembles the TCP stream locally, multiplexes it statefully over an ssh session, and disassembles it back into packets at the other end. So it never ends up doing TCP-over-TCP. It’s just data-over-TCP, which is safe.”
- <https://sshuttle.readthedocs.org/en/latest/how-it-works.html>

Requirements / Voraussetzungen

Client

- sudo / root
- Python 2.7 oder
- Python 3.5
- sshuttle
- OS X: pfctl Kommando

Server

- Python 2.7 oder
- Python 3.5
- ssh Zugriff

→ <https://sshuttle.readthedocs.org/en/latest/requirements.html>

OS Support & IPversion

- Linux \o/, Mac OS X (nur IPv4) und Windows **nicht wirklich**
- Linux IPv4 nutzt NAT
iptables DNAT, REDIRECT, und ttl Module
- Linux IPv6 nutzt TPROXY
Siehe Dokumentation

Installation

- Via Repo:

```
apt install sshuttle  
dnf install sshuttle
```

- Via pip (python installer)

```
pip install sshuttle
```

- Via git repo

```
git clone https://github.com/sshuttle/sshuttle.git  
./setup.py install
```

Verwendung

Syntax:

```
sshuttle [options] [remote] [subnet]
```

Beispiel:

```
sshuttle --dns -r user@remote 192.172.1.0/24
```


Optionen

- `--dns` DNS weiterleiten (aus remote net)
- `-r` remote ssh-Server
- `-H`, `--auto-hosts` übernimmt die `/etc/hosts` vom Server
- `-N`, `--auto-nets` übernimmt die subnets aus der Routing-Tabelle des Server
- `--python` Python-Pfad auf Server
- `-x`, `--exclude=subnet` spezifizierte subnets nicht routen
- `-X`, `--exclude-from=file`
- `-v` Verbose-Log

Tipps

- 0/0 als Kürzel für 0.0.0.0/0
- -vv falls die Verbindung schlecht ist und manchmal stoppt.