

Tunneln mit sshuttle

Julian Fietkau

am 2. Dezember 2010
im KunterBuntenSeminar

Versuche, von zuhause aus eine beliebige Webseite aufzurufen, die nur aus dem Informatik-Netz erreichbar ist:

Zugriff verweigert!

Der Zugriff auf das angeforderte Verzeichnis ist nicht möglich. Entweder ist kein Index-Dokument vorhanden oder das Verzeichnis ist zugriffsgeschützt.

Sofern Sie dies für eine Fehlfunktion des Servers halten, informieren Sie bitte den Webmaster hierüber.

Möglichkeit 1: VPN

`fbivpn.informatik.uni-hamburg.de`

- ▶ Zugriff auf fachbereichsinterne Seiten möglich
- ▶ Taugt im UHH-WLAN gleichzeitig als Authentisierung für Internet
- ▶ Aller IP-Verkehr systemweit geht durch das VPN
- ▶ schwerfällig, Verbindungsaufbau dauert
- ▶ nicht immer stabil (Verbindungsabbrüche)

Möglichkeit 2: SSH FBI

```
home$ ssh 0nachnam@rzssh1...  
ikum$ wget ...
```

- ▶ schlank
- ▶ schnell
- ▶ kaum Konfiguration und keine zusätzliche Software nötig
- ▶ Nur die SSH-Verbindung wird getunnelt
- ▶ auf Dauer umständlich und lästig

Möglichkeit 3: SSH + DynamicForward + tsocks

```
home$ ssh fbi & tsocks  
home$ firefox ...
```

- ▶ immer noch schlanker als VPN
- ▶ immer noch schneller als VPN
- ▶ leicht zu handhaben
- ▶ Nur Traffic der tsocks-Shell wird getunnelt
- ▶ umständlich einzurichten

**There's a new tunneling
solution in town...**

sshuttle: where transparent proxy meets VPN meets ssh

„As far as I know, sshuttle is the only program that solves the following common case:

- ▶ Your client machine (or router) is Linux, FreeBSD, or MacOS.
- ▶ You have access to a remote network via ssh.
- ▶ You don't necessarily have admin access on the remote network.
- ▶ The remote network has no VPN, or only stupid/complex VPN protocols (IPsec, PPTP, etc). Or maybe you are the admin and you just got frustrated with the awful state of VPN tools

sshuttle: where transparent proxy meets VPN meets ssh

- ▶ You don't want to create an ssh port forward for every single host/port on the remote network.
- ▶ You hate openssh's port forwarding because it's randomly slow and/or stupid.
- ▶ You can't use openssh's PermitTunnel feature because it's disabled by default on openssh servers; plus it does TCP-over-TCP, which has terrible performance (see below).“

Möglichkeit 4: sshuttle

```
home$ git clone git://github.com/apenwarr/sshuttle
home$ ./sshuttle -r fbi 134.100.0.0/16
```

- ▶ schnell, im Sinne von **schnell** - subjektiv quasi nicht spürbar
- ▶ sehr leichtgewichtig
- ▶ Traffic wird systemweit getunnelt
- ▶ braucht root-Rechte auf dem Client
- ▶ Verbindung bricht manchmal ab, lässt sich jedoch automatisiert neu starten

Funktionsweise von sshuttle

- ▶ sshuttle lädt sich selbst auf den SSH-Server
- ▶ Auf dem Client werden lokale Routing-Regeln aktiviert, so dass sshuttle den Traffic abgreifen kann
- ▶ Der IP-Traffic geht über die SSH-Verbindung ans serverseitige sshuttle und von dort ans Ziel
- ▶ übrigens: komplett in Python geschrieben

Verwendung

- ▶ `sshuttle -r fbi 134.100.0.0/16` in einem lokalen Terminal
- ▶ surfen, chatten, etc. wie gewohnt - jeglicher Traffic zu Uni-Servern wird dabei transparent über `rzssh1` getunnelt
- ▶ zum Beenden: `Strg+C` in dem Terminal in dem `sshuttle` läuft
- ▶ bei Verbindungsabbruch: neu starten
 - ▶ primitive Selbstreparatur:

```
while true; do sshuttle -r ikum 134.100.0.0/16; sleep 2; done
```

Danke für die Aufmerksamkeit!



http://www.julian-fietkau.de/tunneln_mit_sshuttle

