

Einrichtung eines virtuellen Clients

In diesem Artikel wird beschrieben, wie du einen virtuellen Client für Freifunk Münster einrichten kannst. Ein virtueller Client ist beispielsweise ein Root-Server, der sich in das Freifunk-Netz einwählt und dort Dienste anbietet, dabei aber nicht über einen Knoten (also einem von den Gluon-Routern) angebunden ist.

Die Wahl der Linux-Distribution

Wir verwenden bisher auf unseren virtuellen Clients Debian Jessie, haben daher auch die meiste Erfahrung damit.

Paketquellen

Folgende zusätzliche Paketquellen werden benötigt:

```
deb http://repo.universe-factory.net/debian/ sid main
```

```
deb http://download.opensuse.org/repositories/home:/fusselkater:/ffms/Debian_7.0/ /
```

Um die nötigen Schlüssel hinzuzufügen, mache folgendes:

```
gpg --keyserver pgpkeys.mit.edu --recv-key 16EF3F64CB201D9C  
gpg -a --export 16EF3F64CB201D9C | apt-key add -
```

```
wget http://download.opensuse.org/repositories/home:fusselkater:ffms/Debian_7.0/Release.key  
apt-key add - < Release.key
```

Nun sind die benötigten Repos eingetragen und du kannst mit

```
aptitude update
```

die Paketquellen aktualisieren.

Notwendige Pakete

Folgende Pakete müssen nun auf deinem Client installiert werden:

Paket	Beschreibung
bridge-utils	Verwaltung von Netzwerkbrücken
batctl	B.A.T.M.A.N. Verwaltungstools
fastd	VPN für Verbindung zu den Gateways
git	Versionsverwaltungssystem
alfred-stable	A.L.F.R.E.D. Datenübertragung
batman-adv-dkms	Batman-Kernelmodul in Version 2013.04

Oder als ein Befehl:

```
aptitude install bridge-utils batctl fastd git alfred-stable alfred-json batman-adv-dkms
```

batman-adv installieren

Nachdem batman-adv-dkms installiert ist, schmeißt dkms einen Fehler, da in den aktuellen Kernel-Versionen eine neuere batman-adv Version ist, als über dkms gebaut wurde. Es ist nötig, die Installation des 2013.04er Modules zu erzwingen. Dazu macht man folgendes:

```
dkms remove batman-adv/2013.4.0 --all  
dkms --force install batman-adv/2013.4.0
```

Anschließend trägst du in der Datei `/etc/modules` das Modul `batman-adv` ein.

Danach einmal rebooten. Nun sollte batman-adv Version 2013.4 geladen sein. Verifizieren kann man dies mit:

```
cat /sys/module/batman_adv/version
```

Netzwerkconfiguration

Als erstes brauchen wir eine Netzwerkbrücke, als Schnittstelle zwischen dem Mesh-Netz und dem MULLVAD-VPN.

Dazu füge in die Konfigurationsdatei `/etc/network/interfaces` folgendes hinzu:
<file - interfaces>

1. Netzwerkbrücke für Freifunk
2. - Hier läuft der Traffic von den einzelnen Routern und dem externen VPN zusammen

3. - Unter der hier konfigurierten IP ist der Server selber im Freifunk Netz erreichbar
4. - `bridge_ports none` sorgt dafür, dass die brücke auch ohne Interface erstellt wird

```
auto br0

iface br0 inet static
    address 10.43.0.z
    netmask 255.255.0.0
    bridge_ports none
```

```
iface br0 inet6 static
    address 2a03:2260:115::z
    netmask 48
```

Das `z` ist hier gegen die Zahl zu ersetzen, die wir dem neuen Client geben möchten.

Als nächstes muss das `bat0`-Interface konfiguriert werden. Dazu bearbeitest du wieder die `/etc/network/interfaces`

1. Batman Interface
2. - Erstellt das virtuelle Interface für das Batman-Modul und bindet dieses an die Netzwerkbrücke

```
allow-hotplug bat0

iface bat0 inet6 manual
    pre-up modprobe batman-adv
    post-up ip link set dev bat0 up
    post-up brctl addif br0 bat0
    post-up batctl it 10000
```

Zur Erklärung:

Bevor das Netzwerkinterface gestartet wird, wird zur Sicherheit nochmal `batman-adv` geladen.

Am Ende wird noch `alfred` gestartet. Dieser ist Optional und wird benötigt, falls du Daten von den Knoten verarbeiten möchtest.

Um diese Änderung zu aktivieren, startest du das Netzwerk einmal per `service networking restart neu`.

Einrichtung des Mesh-VPNs

Für das Mesh-VPN wird `fastd` eingesetzt. Hierzu erzeugst du als erstes das Konfigurationsverzeichnis:

```
mkdir -p /etc/fastd/vpn/peers
```

Als nächstes erzeugst du die Schlüssel für deinen Client. Dazu führst du folgendes aus:

```
fastd --generate-key
```

dieser Befehl zeigt dir die Schlüssel lediglich an. Heißt du musst sie dir aus dem Terminal kopieren, um sie im nächsten Schritt verwenden zu können.

Jetzt legst du eine Konfigurationsdatei `/etc/fastd/vpn/fastd.conf` mit folgendem Inhalt an:

```
interface "mesh-vpn";
user "nobody";
mode tap;
method "salsa2012+gmac";
mtu 1406; # 1406 - ipv6 header - fastd header
secret "SERVER-SECRET-KEY";

log to syslog level debug;
#folgende Zeile sorgt dafuer das jeder Peer akzeptiert wird
#on verify "true";

include peers from "/var/gateway-ffms/backbone/";
on up "
ip link set dev $INTERFACE address de:ad:be:ef:43:XX
ip link set dev $INTERFACE up
ifup bat0
batctl if add $INTERFACE

";
```

Hierbei ist `SERVER-SECRET-KEY` gegen deinen geheimen Schlüssel zu ersetzen.
XX ist gegen die Mac-Adresse zu ersetzen, die wir dem Client geben möchten.

Nun brauchst du die Peers aus unserem GIT-Repository. Wechsle dazu in das Verzeichnis `/var` und führe folgendes aus:

```
git clone https://github.com/FreiFunkMuenster/gateway-ffms.git
```

Zum testen der `fastd`-Konfiguration führe folgendes aus:

```
fastd -c /etc/fastd/vpn/fastd.conf
```

Anschließend kannst du fastd mit

```
service fastd start
```

starten.

Den Öffentlichen Schlüssel deines Clients schickst du uns, damit wir ihn einpflegen können.

Git-Repository automatisch aktualisieren

Das gateway-ffms Repository muss regelmäßig automatisch aktualisiert werden. Dazu befindet sich bereits ein kleines Shell-Script in dem Repository. Du musst lediglich einen Cronjob anlegen, der dieses Script aufruft:
/etc/cron.d/update-gw

```
0 * * * * root /bin/bash /var/gateway-ffms/update.sh
```

Abschluss

Dein Client ist nun für den Betrieb als Gateway für Freifunk Münster eingerichtet. Am besten startest du den Client einmal neu, und kontrollierst, dass alles korrekt hoch kommt.