

Tinc > Wiki > ubuntuusers.de

[Originalartikel](#)

[Backup](#)

`<html> <div class=„box tested_for“> <h3 class=„box tested_for“>Dieser Artikel wurde für die folgenden Ubuntu-Versionen getestet:</h3> </div> <div class=„box advanced“ readability=„6“> <h3 class=„box advanced“>Artikel für fortgeschrittene Anwender</h3> <div class=„contents“ readability=„37“> <p>Dieser Artikel erfordert mehr Erfahrung im Umgang mit Linux und ist daher nur für fortgeschrittene Benutzer gedacht.</p> </div> </div> <div class=„box knowledge“> <h3 class=„box knowledge“>Zum Verständnis dieses Artikels sind folgende Seiten hilfreich:</h3> </div> <div class=„toc toc-depth-1“> <p>Inhaltsverzeichnis</p> <ol class=„arabic“>Installation Konfiguration Beispiele Problembehebung Links </div> <p> Tinc ist eine dezentrale Virtual-Private-Network- oder kurz VPN-Software. Sie zeichnet sich durch eine sehr schlanke und einfache Konfiguration, plattformübergreifende Unterstützung (Linux, BSD, Solaris, Mac OS X und Windows), vollständige asymmetrische Verschlüsselung und dezentrales Mesh Routing aus.</p> <p>Bei letzterem verbinden sich alle Teilnehmer (Knoten) miteinander, um jeweils die beste Verbindung zu erreichen. Gleichzeitig können aber beliebig viele Knoten ausfallen, ohne dass die Verbindung zwischen zwei Knoten beeinträchtigt wird, solange noch eine einzige indirekte Verbindung verbleibt. Das virtuelle Netzwerk wird dabei standardmäßig auf Layer 3 (IP) im OSI-Modell erstellt.</p> <div class=„box notice“ readability=„8“> <h3 class=„box notice“>Hinweis:</h3> <div class=„contents“ readability=„41“> <p>Um diesem Artikel folgen zu können, werden grundlegende Netzwerkkennntnisse vorausgesetzt. Als Beispiel wird das Subnet`

10.8.0.0/16

verwendet, welches alle IP-Adressen von


10.8.0.1

bis

10.8.255.254

beinhaltet.

Experten-Info:

Vor Version 1.1 unterstützt Tinc das sogenannte [Hole Punching](https://de.wikipedia.org/wiki/Hole%20Punching) nicht, wie etwa die VPN-Software [n2n](http://www.ntop.org/products/n2n/) . D.h. bei mindestens einem Knoten muss der von Tinc verwendete Port

655

weitergeleitet werden, und über diesen Knoten fließt dann der komplette Netzwerkverkehr.

Installation

Tinc kann direkt aus den offiziellen Paketquellen installiert [werden](https://wiki.ubuntuusers.de/Tinc/#source-1):

- tinc** (**universe**)

[!\[\]\(339a16584d5da0f0a3ca4e9ec17bf6a1_img.jpg\) Wiki/Vorlagen/Installbutton/button.png](apt:tinc) mit [apturl](https://wiki.ubuntuusers.de/apturl/)

Paketliste zum Kopieren:

```
sudo apt-get install tinc
```

```
sudo aptitude install tinc
```

Allerdings ist die enthaltene Version nicht immer die aktuellste. Für die neueste Ausgabe der Produktiv- oder Entwicklungsversion empfiehlt sich daher die Kompilierung von Hand.

Manuell

Den Quellcode findet man auf der [Homepage](http://tinc-vpn.org/download/)   des Projekts, die Kompilierung erfolgt wie in [beschrieben](https://wiki.ubuntuusers.de/Tinc/#source-2). Zu beachten gilt dabei, dass bei einer manuellen Kompilierung per Standard alles in das Verzeichnis **/usr/local/** (siehe [Verzeichnisstruktur](https://wiki.ubuntuusers.de/Verzeichnisstruktur/#usr)) verschoben wird. Dieses Verhalten kann wie in [Programme kompilieren](https://wiki.ubuntuusers.de/Programme_kompilieren/#Weitere-Optionen) beschrieben geändert werden. Meist müssen neben den von über [:](https://wiki.ubuntuusers.de/Tinc/#source-3)

```
sudo apt-get build-dep tinc
```

 gefundenen Bibliotheken noch die Pakete **libreadline6-dev** und **libncurses5-dev** nachinstalliert werden.

Tinc 1.1 unterscheidet sich von Version 1.0 u.a. durch eine vereinfachte Konfiguration und Unterstützung von [Elliptic Curve Cryptography](https://de.wikipedia.org/wiki/Elliptic%20Curve%20Cryptography). Allerdings ist diese Version noch nicht als stabil gekennzeichnet und sollte somit nur zum Testen benutzt werden.

Konfiguration

Die Konfigurationsdateien finden sich im Verzeichnis **/etc/tinc/**. Werden mehrere Netzwerke verwendet, so legt man für jedes ein Unterverzeichnis an, die später mit dem Parameter `-n Netzwerkname` ausgewählt werden. Es wird empfohlen, auch für ein einzelnes

https://schnipsel.ggelm.de/

Printed on 2025/11/13 06:06

Netzwerk ein Unterverzeichnis anzulegen.

Allgemeine Einstellungen werden in der Datei `/etc/tinc/NETZWERK/tinc.conf` festgelegt. Für jeden Knoten gibt es dann nochmals eine eigene Datei im Unterverzeichnis `hosts`. Das hosts-Verzeichnis ist dabei bei allen Knoten identisch. Diese Syntax in den Dateien folgt dem Schema `Variablenname = Variablenwert` . Zudem gibt es die Skripte `tinc-up` und `tinc-down`, welche beim Starten bzw. Beenden von Tinc ausgeführt werden.

Beispielsweise existiert folgende Konfiguration:

```

/etc/tinc/NETZWERK/tinc.conf /etc/tinc/NETZWERK/hosts/ersterknoten
/etc/tinc/NETZWERK/hosts/zweiterknoten /etc/tinc/NETZWERK/tinc-up /etc/tinc/NETZWERK/tinc-down

```

tinc.conf

Zuerst wird der Name des eigenen Knotens definiert <https://wiki.ubuntuusers.de/Tinc/#source-5> [5]:

```

Name = ersterknoten

```

Danach der Ort des TUN-Adapters (unter Linux `/dev/net/tun`):

```

Device = /dev/net/tun

```

Zuletzt die Angabe, zu welchen anderen Knoten verbunden werden soll (ohne diese Angabe wartet der Knoten nur auf eingehende Verbindungen):

```

ConnectTo = zweiterknoten

```

Hinweis:

Ab Tinc 1.1 lässt sich dieser Schritt komfortabel mit

```

tinc -n NETZWERK init ersterknoten

```

```

tinc -n NETZWERK add connectto zweiterknoten

```

ausführen.

hosts

Man legt eine neue Datei mit dem Namen `/etc/tinc/NETZWERK/hosts/ersterknoten`, der in `tinc.conf` angegeben wurde, an. Zuerst wird die öffentliche IP-Adresse (oder eine darauf verweisende Domain) des Knotens angegeben.

```

Address =
oeffentliche.ip.adresse.des.servers.org

```

Anschließend das Subnet des eigenen Knotens (angenommen, der Knoten besteht nur aus einem Rechner, hat das Subnet die Maske `/32`):

```

Subnet = 10.8.0.1/32

```

Ebenso möglich wäre aber auch, jedem Knoten 255 Adressen zuzuweisen (z.B. von `10.8.1.0` bis `10.8.1.255`):

```

Subnet = 10.8.1.0/24

```

Schlüssel generieren

Dazu wird der folgende Befehl ausgeführt:

```

tincd -n NETZWERK -K

```

Hinweis:

Dieser Schritt entfällt ab Version 1.1

tinc-up

Zuletzt wird noch das ausführbare <https://wiki.ubuntuusers.de/Tinc/#source-6> [6]: Skript `tinc-up` angelegt.

	<div class="nottranslate syntax" readability="6"> <pre> #!/bin/sh ifconfig \$INTERFACE 10.8.0.1 netmask 255.255.0.0 </pre> </div>
--	---

Die IP-Adresse gibt diejenige Adresse aus dem Knoten-Subnet an, unter der man später erreichbar sein wird. Die Netzmaske entspricht hierbei der des „kompletten“ Tinc-Netzwerkes.

Weitere Knoten

Diese Schritte werden für jeden Knoten ausgeführt, jeweils mit einem eigenen Subnet / eigener IP-Adresse. Als letztes werden die Dateien im host-Verzeichnis an jeden anderen Knoten weitergegeben.

Hinweis:

Tinc 1.1 unterstützt hierbei die komfortablen Optionen `import`, `export` und `exchange`. So kann

Qgelm - <https://schnipsel.qgelm.de/>

man die Konfiguration via SSH mit dem Server `foobar.example.org` synchronisieren:

```
tinc -n netname export | ssh foobar.example.org tinc -n netname exchange | tinc -n netname import
```

Experten-Info:

Es gibt noch einige weitere Konfigurationsmöglichkeiten, u.a. kann Tinc auch ein auf Layer-2-basiertes Netzwerk erstellen oder IPv6 und ECDSA nutzen. Hier sei auf die umfangreichen Manpages von Tinc verwiesen.

Tinc starten

Tinc wird mit dem Kommando `tincd` bedient. Zum Starten dient die Option `start`:

```
tincd -n NETZWERK start
```

Hinweis:

Ab Tinc 1.1 werden alle Kommandos mit dem Befehl `tinc` (statt `tincd`) durchgeführt. Außerdem gibt es eine Möglichkeit das Netzwerk mit dem Parameter `top`.

Optional: Symbolische Namen

Man kann das eigene virtuelle Netzwerk in die Datei `/etc/networks` eintragen. Als Beispiel:

```
NETZWERK 10.0.0.0 # Netzwerkname und Subnet
```

Außerdem kann man in der Datei `/etc/services` dem Tinc-Service eine Port-Nummer (Standard: `655`) zuweisen, dadurch zeigen manche Programme statt der Portnummer den Namen des Services an:

```
tinc 655/tcp TINC tinc 655/udp TINC
```

Beispiele

Tinc als Default-Gateway

Wenn man Tinc als Default-Gateway nutzt, so bestimmt man einen Knoten, über den dann der komplette eigene Internetverkehr geroutet wird - mit Ausnahme der Verbindung zu diesem Knoten.

Dann benutzt man diesen Knoten als VPN, um z.B. Zensur zu umgehen oder Man-in-the-middle-Attacken in öffentlichen WLANs vorzubeugen. Bei OpenVPN ist dies als `redirect-gateway` bekannt. Im Folgenden wird der Knoten, der das VPN bereitstellt, „Server“ genannt, die Knoten, welche ihre Verbindung über den Server leiten, „Client“.

Server

Allgemein muss dazu in der Datei `/etc/sysctl.conf` die IP-Weiterleitung aktiviert sein. Dazu entfernt man das Kommentarzeichen vor der entsprechenden Zeile:

```
net.ipv4.ip_forward=1 # für IPv4
```

```
net.ipv6.conf.all.forwarding=1 # für IPv6
```

und übernimmt die Einstellungen mit:

```
sysctl -p
```

Dann ändert man in `/etc/hosts/server` das Subnet auf `Subnet = 0.0.0.0/0` (IPv4) bzw. `:::0` (IPv6). Zudem fügt man der Datei `/etc/rc.local` folgendes hinzu (`eth0` muss durch den Netzwerkadapter mit der öffentlichen IP des Servers ersetzt werden, `tun0` durch den Netzwerkadapter mit der VPN-internen IP, `10.8.0.0/16` gibt das Tinc-Subnet an):

```
iptables -A FORWARD -o eth0 -i tun0 -s 10.8.0.0/16 -m conntrack -ctstate NEW -j ACCEPT
iptables -A FORWARD -m conntrack -ctstate ESTABLISHED,RELATED -j ACCEPT
iptables -t nat -A POSTROUTING -o eth0 -s 10.8.0.0/16 -j MASQUERADE
```

Sollte dies nicht funktionieren, kann man teilweise

auch Source-NAT verwenden, siehe <a

href=„<https://wiki.ubuntuusers.de/Tinc/#Virtualisierte-Systeme>“

class=„crosslink“>Problembhebung.</p> </div> <div class=„section_3“ readability=„6“>

<h4 id=„Client“>Client</h4> <p>Auf den Clients werden nun die host-up und host-down Dateien angepasst.</p> <ul readability=„1“><li readability=„2“>

<p>/etc/tinc/NETZWERK/hosts/server-up (INTERNE_SERVER_IP wird durch die

VPN-interne Netzwerkadresse des Servers ersetzt):</p> <div class=„code“> <table class=„nottranslate syntaxtable“ readability=„2“><tr readability=„5“><td class=„linenos“> <div class=„linenodiv“> <pre> 1 2 3 4 5 6 7 8 </pre></div> </td> <td class=„code“ readability=„5“> <div class=„nottranslate syntax“ readability=„10“> <pre> #!/bin/sh

VPN_GATEWAY=INTERNE_SERVER_IP ORIGINAL_GATEWAY=`ip route show | grep ^default | cut -d ' ' -f 2-5` ip route add \$REMOTEADDRESS \$ORIGINAL_GATEWAY ip route add \$VPN_GATEWAY dev \$INTERFACE ip route add 0.0.0.0/1 via \$VPN_GATEWAY dev \$INTERFACE ip route add 128.0.0.0/1 via \$VPN_GATEWAY dev \$INTERFACE </pre></div> </td> </tr></table></div> <li

readability=„1“> <p>/etc/tinc/NETZWERK/hosts/server-down:</p> <div

class=„code“> <table class=„nottranslate syntaxtable“ readability=„2“><tr readability=„4“><td class=„linenos“> <div class=„linenodiv“> <pre> 1 2 3 4 5 6 7 </pre></div> </td> <td class=„code“ readability=„5“> <div class=„nottranslate syntax“ readability=„9“> <pre> #!/bin/sh ORIGINAL_GATEWAY=`ip route show | grep ^default | cut -d ' ' -f 2-5` ip route del \$REMOTEADDRESS \$ORIGINAL_GATEWAY ip route del \$VPN_GATEWAY dev \$INTERFACE ip route del 0.0.0.0/1 dev \$INTERFACE ip route del 128.0.0.0/1 dev \$INTERFACE </pre></div> </td> </tr></table></div>

 </div> </div> </div> <div class=„section_1“ readability=„11“> <h2

id=„Problembhebung“>Problembhebung</h2> <p>Sollte etwas nicht wie gewünscht

funktionieren, so lässt sich Tinc mit den Parametern <code class=„nottranslate“>-D -d5</code> im Debug-Modus starten. Wird dadurch sehr viel ausgegeben, so kann man auch eine geringere Zahl als Parameter angeben.</p> <div class=„section_2“ readability=„10“> <h3

id=„Firewall“>Firewall</h3> <div class=„section_3“ readability=„11“> <h4 id=„UDP-Pakete-

werden-blockiert“>UDP Pakete werden blockiert</h4> <p>Tinc lässt sich auch über eine TCP-Verbindung tunnelt, was bei einigen Firewalls notwendig ist, allerdings die Geschwindigkeit beeinflussen kann. In der entsprechenden host-Datei wird folgendes

eingefügt:</p> <pre class=„nottranslate“> TCPonly = yes </pre></div> <div

class=„section_3“ readability=„10“> <h4 id=„Port-wird-blockiert“>Port wird blockiert</h4> <p>Um den Port, über den ein Knoten erreichbar ist, zu ändern, müssen dessen

tinc.conf und host-Datei angepasst werden:</p> <ul

readability=„1“><li readability=„2“> <p>tinc.conf:</p> <pre class=„nottranslate“> BindToAddress = *:1234 </pre> <p>Anstelle des Sternchens lässt sich auch eine spezielle Netzwerkadresse angeben.</p>

<p>hosts/knoten:</p> <pre class=„nottranslate“> Port = 12345 </pre>

</div> </div> <div class=„section_2“ readability=„13“> <h3 id=„Virtualisierte-

Systeme“>Virtualisierte Systeme</h3> <div class=„section_3“ readability=„14“> <h4 id=„TUN-TAP“>TUN/TAP</h4> <p>Auf vielen virtuellen Servern teilt sich das virtualisierte Betriebssystem den Kernel mit dem Host, sodass einige benötigte Module nicht verfügbar sein können. Ob das TUN/TAP-Modul geladen wurde, erkennt man mit dem Befehl:</p> <div class=„bash

contents“> <pre class=„nottranslate“> lsmod | grep tun </pre></div> <p>Sollte dies nicht der Fall sein (der Befehl gibt nichts zurück), so muss man das Gerät selbst erstellen:</p> <div class=„bash contents“ readability=„36“> <pre class=„nottranslate“> sudo mkdir -p /dev/net sudo mknod /dev/net/tun c 10 200 sudo chmod 600 /dev/net/tun </pre></div> </div> <div

class=„section_3“ readability=„14“> <h4 id=„iptables-masquerading“>iptables masquerading</h4> <p>Sollte das Masquerading-Modul von <a href=„<https://wiki.ubuntuusers.de/iptables/>“

class=„internal“>iptables nicht verfügbar sein, so kann man auch Source-NAT benutzen. Dazu ersetzt man in /etc/rc.local:</p> <pre class=„nottranslate“> iptables -t nat -

A POSTROUTING -o eth0 -s 10.8.0.0/16 -j MASQUERADE # das geht nur, wenn das Interface nicht venet0:0 (also keine Map) ist </pre> <p>durch</p> <pre class=„nottranslate“> iptables -t nat -A POSTROUTING -s 10.8.0.0/16 -j SNAT -to-source EXT_IP </pre> <p>wobei <code class=„nottranslate“>EXT_IP</code> durch die öffentliche IP des Servers ersetzt wird.</p> <div class=„box notice“ readability=„6“> <h3 class=„box notice“>Hinweis:</h3> <div class=„contents“ readability=„37“> <p>Oftmals reicht eine Anfrage an die Administratoren des Servers, um die benötigten Module zu laden.</p> </div> </div> </div> </div> <div class=„section_1“> <ul readability=„1“> <p>Projektseite </p> <p>Dokumentation </p> <li readability=„0“> <p>Dezentrales VPN mit Tinc - Tinc vermascht Rechner sicher übers Internet - heise Netze, 09/2009</p> <li readability=„0“> <p>Tunnelbau - Virtual Private Networks mit Tinc - Artikel ADMIN Magazin, 02/2009</p> <li readability=„1“> <p>VPN Programmübersicht</p> <li readability=„1“> <p>Tunneling 101 – von überall ins Netz (SSH, Tinc, Socks, Krypto) - Vortrag, Ubucon Berlin 2015</p> </div> </html>

From:

<https://schnipsl.qgelm.de/> - Qgelm

Permanent link:

<https://schnipsl.qgelm.de/doku.php?id=wallabag:tinc--wiki--ubuntuusers.de>

Last update: **2021/12/06 15:24**

