Wiki / Router

Dieser Artikel wurde für die folgenden Ubuntu-Versionen getestet:

Dieser Artikel ist größtenteils für alle Ubuntu-Versionen gültig.

Zum Verständnis dieses Artikels sind folgende Seiten hilfreich:

1. **Installation von Programmen**
2. **Einen Editor öffnen**
3. **root-Rechte**
4. **Ein Terminal öffnen**

Inhaltsverzeichnis

[//media-cdn.ubuntu-de.org/wiki/attachments/37/41/applications-internet.png] Ein Router ist ein Verbindungsrechner, der Daten zwischen zwei Netzwerken verteilt. Im Zeitalter von schnellem Internet über DSL verwendet man im allgemeinen einen **Hardware-Router** [https://wiki.ubuntuusers.de/Hardware-Router] als Vermittler zwischen dem öffentlichen Internet und einem lokalen Netzwerk.

Dazu sind verschiedene Hardware-Komponenten, wie das DSL-Modem , diverse Schnittstellen, ein kleiner **Netzwerk-Switch** [http://de.wikipedia.org/wiki/Switch_(Computertechnik)] und über Software gesteuerte Funktionen wie **NAT** [http://de.wikipedia.org/wiki/Network_Address_Translation], ein **DNS** [http://de.wikipedia.org/wiki/Domain_Name_System]-System, ein **DHCP** [http://de.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol]-Server und eine **Firewall** [http://de.wikipedia.org/wiki/Firewall] in einem einzigen Gerät vereint.

Unter Ubuntu lässt sich diese Aufgabe ebenfalls lösen und auf einem Rechner umsetzen. Dieser Artikel bildet das Grundgerüst für folgende weiterführende und ergänzende Artikel zu dem Thema hier im uu-Wiki und geht ausführlicher auf die Basiskonfiguration ein.

- **Internetverbindungsfreigabe** [https://wiki.ubuntuusers.de/Internetverbindungsfreigabe]
- **WLAN Router** [https://wiki.ubuntuusers.de/WLAN_Router]

Für den praktischen Einsatz bieten sich insbesondere **Embedded-PC** [http://de.wikipedia.org/wiki/Embedded-PC]-Systeme wie z.B. von **Alix** [https://wiki.ubuntuusers.de/Alix] oder **VIA Epia** [http://de.wikipedia.org/wiki/EPIA] an, die zwei Netzwerkschnittstellen besitzen und besonders wenig Strom verbrauchen. Allerdings muss man berücksichtigen, dass die Kernel neuerer Ubuntu-Versionen mit dieser Hardware nicht mehr zwangsläufig funktionieren.

Linux als Gateway

Konfiguration Server

Der Dienst **dnsmasq** bietet einen DHCP- und DNS-Cache in einem. Dafür wird das folgende Paket benötigt [1]:

- **dnsmasq** (*universe* - siehe auch **Dnsmasq** [https://wiki.ubuntuusers.de/Dnsmasq])

[apt://dnsmasq] mit **apturl** [https://wiki.ubuntuusers.de/apturl]

```
sudo apt-get install dnsmasq
```

Die Konfigurationsdateien können mit einem Editor [2] und Root-Rechten [3] bearbeitet werden. In der Datei **/etc/dnsmasq.conf** werden sämtliche Einstellungen vorgenommen. In der der nach der Installation bereits vorhandenen Konfigurationsdatei sind mögliche Parameter anhand von Beispielen erklärt. Die Konfiguration kann, je nach Einsatzzweck recht umfangreich werden. Für eine Basiskonfiguration werden zunächst jedoch nur ein paar Einstellungen benötigt.

- interface beschränkt den Server-Dienst nur auf ein oder auch mehrere Interfaces - in diesem Beispiel für eth1 wird folgendes eingetragen:

```
# DHCP-Server aktiv für Interface
interface=eth1
```

Für jede weitere Schnittstelle muss ein neuer Eintrag angehängt werden.

- no-dhcp-interface Kein DHCP-Service für die angegebenen Schnittstellen. Die Schnittstelle welche z.B. die Verbindung zum **WAN** [http://de.wikipedia.org/wiki/Wide_Area_Network] herstellt, soll von dnsmasq ignoriert werden:

```
# DHCP-Server nicht aktiv für Interface
no-dhcp-interface=eth0
```

Auch hier sind mehrere Einträge möglich.

- dhcp-range legt den zu vergebenden Adressbereich fest. Der Schnittstelle eth1 muss zuvor über die Datei **interfaces** [https://wiki.ubuntuusers.de/interfaces] eine feste IP-Adresse zugewiesen werden. Passend zum Beispiel hier die Adresse 192.168.3.1. Für weitere Schnittstellen muss identisch verfahren werden.

```
dhcp-range=192.168.3.20,192.168.3.50,12h
```

In diesem Beispiel befindet sich der zu vergebende IP-Adressbereich zwischen 192.168.3.20 und 192.168.3.50. 12h bedeutet, dass die IP-Adresse 12 Stunden lang gültig ist und dann automatisch erneuert wird. Alternativ kann die automatisch zugewiesene Adresse auch dauerhaft vergeben werden:

```
dhcp-range=192.168.3.20,192.168.3.50,infinite
```

Bei mehreren Schnittstellen/Subnetzen sollte der eingestellte Adressbereich der zugehörigen Schnittstelle zugewiesen werden:

```
dhcp-range=interface:eth1,192.168.3.20,192.168.3.50,infinite
```

```
dhcp-range=interface:eth2,192.168.4.20,192.168.4.50,infinite
```

LAN-Schnittstellen kann z.B. auch ein Adressbereich im selben Subnetz zugewiesen werden, bei LAN- und WLAN-Schnittstellen in Kombination funktioniert das jedoch nicht zuverlässig. Beispiel:

```
dhcp-range=interface:eth1,192.168.3.20,192.168.3.50,infinite
dhcp-range=interface:eth2,192.168.3.60,192.168.3.80,infinite
dhcp-range=interface:wlan0,192.168.4.10,192.168.4.30,infinite
```

Je zwei Schnittstellen können auch auf einen gemeinsamen Adressbereich festgelegt werden. Beispiel:

```
dhcp-range=interface:eth1,eth2,192.168.3.20,192.168.3.60,infinite
```

Auch hier funktioniert das bei LAN- und WLAN-Schnittstellen in Kombination nicht zuverlässig.

- dhcp-host ermöglicht es Rechnern über den Namen und/oder über die MAC-Adresse der jeweiligen Netzwerkkarte eine feste IP-Adresse zuzuweisen.

```
dhcp-host=0B:01:02:03:25:67,192.168.3.10,infinite
```

Hier wird z.B. dem Rechner oder der Netzwerkkarte mit der MAC-Adresse 0B:01:02:03:25:67 die IP-Adresse: 192.168.3.10 für unbegrenzte Zeit zugewiesen. Diese Einstellung empfiehlt sich für Rechner, die z.B. Drucker- oder Dateifreigaben im Netzwerk zur Verfügung stellen oder für bestimmte Dienste Portfreigaben benötigen.

Syntax:

```
dhcp-host=<MAC-Adresse>,<Rechnername>,<IP-Adresse>,infinite
dhcp-host=<MAC-Adresse>,<IP-Adresse>,infinite
dhcp-host=<Rechnername>,<IP-Adresse>,infinite
```

Auf die Konfiguration, mit welcher für LAN und WLAN ein gemeinsamer Adressbereich verwendet werden kann, wird in **WLAN_Router - Netzwerkbrücke** [https://wiki.ubuntuusers.de /WLAN_Router#Netzwerkbruecke] näher eingegangen.

Alternativ zu **dnsmasq** kann auch als **dhcp3** [https://wiki.ubuntuusers.de/ISC-DHCPD] als DHCP-Server konfiguriert werden.

Portweiterleitung

Damit Daten vom WAN über Schnittstelle 1 (eth0) jetzt in das lokale Netz (eth1, eth2, wlan0...) übertragen werden können, muss die **Portweiterleitung** [http://de.wikipedia.org/wiki/Portweiterleitung] des Kernels (IP-Forwarding) über **sysctl** aktiviert und über entsprechende **iptables** [https://wiki.ubuntuusers.de/iptables2] Filterregeln festgelegt werden, wohin genau die Datenpakete weitergeleitet werden sollen (**NAT** [http://de.wikipedia.org/wiki/Network_Address_Translation]). Zudem wird über sog. Masquerading das lokale Netzwerk von der Außenwelt, dem Internet, abgeschirmt (maskiert).

Hier wird also von eth0 auf die Schnittstellen, an denen die Rechner des lokalen Netzwerks angeschlossen sind, weitergeleitet. Dabei kann es sich sowohl um kabelgebundene als auch um Drahtlos-Verbindungen

handeln.

Die Einrichtung einer **Portweiterleitung** [https://wiki.ubuntuusers.de/Portweiterleitung] bei betriebsfertigen DSL-Routern der Internetanbieter, ohne Ubuntu und über eine Weboberfläche, wird in einem eigenen Artikel beschrieben.

Forwarding

- IP-Forwarding aktivieren [4]:

```
sudo sysctl -w net.ipv4.ip_forward=1
```

- Filterregeln unter Angabe der konfigurierten Schnittstelle und der benötigten **Netzmaske** [http://de.wikipedia.org/wiki/Netzmaske]:

```
sudo iptables -A FORWARD -o eth0 -i eth1 -s 192.168.3.0/24 -m conntrack
--ctstate NEW -j ACCEPT
```

  Mit -i wird die Eingangs-Schnittstelle festgelegt, wenn mehrere interne Netzwerke so auf das Internet zugreifen sollen, kann diese Option weggelassen werden. -s gibt die Quell-IP (inkl. Netzwerkmaske) an. Möchte man z.B. alle privaten Netzwerke erlauben, so muss hier 192.168.0.0/16 eingetragen werden.
- IP-Forwarding für mehrere Subnetze erlauben:

```
sudo iptables -A FORWARD -o eth0 -s 192.168.0.0/16 -m conntrack --ctstate
NEW -j ACCEPT
```

- Einmal zugelassene Verbindungen weiterhin akzeptieren:

```
sudo iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j
ACCEPT
```

NAT

- Adressumsetzung (NAT) aktivieren und die Schnittstelle maskieren:

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Die Einstellungen sind temporär und gehen bei einem Neustart des Systems verloren, können aber auch gespeichert und bei Systemstart wieder abgerufen werden.

iptables-Tabellen speichern

Beispiel um iptables-Tabellen zu speichern:

```
sudo iptables-save > /etc/iptables_01.save
```

Beispiel um iptables-Tabellen wieder zu laden. Vorhandenen Regeln werden dabei gelöscht bzw. überschrieben.

```
sudo iptables-restore < /etc/iptables_01.save
```

So werden bereits vorhandene Regeln nicht gelöscht:

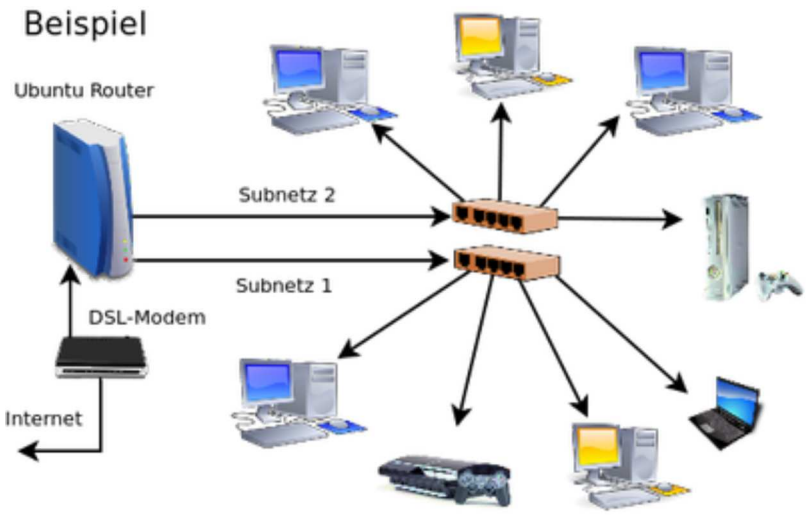
```
sudo iptables-restore -n < /etc/iptables_01.save
sudo iptables-restore --noflush < /etc/iptables_01.save
```

So können natürlich auch mehrere Tabellen gespeichert und je nach Bedarf wieder geladen werden.

iptables-Tabellen löschen

```
sudo iptables --flush
```

[//media-cdn.ubuntu-de.org /wiki/attachments/26/07 /Netzwerk_Router01.png]



Praktische Anwendung

Die Arbeitsweise soll nun an einem praktischen Beispiel dargestellt werden. Die Konfiguration erfolgt hier über die Datei **interfaces** [https://wiki.ubuntuusers.de/interfaces] und **Dnsmasq** [https://wiki.ubuntuusers.de/Dnsmasq]. Die Internetverbindung wird über eth0 und einem angeschlossenen Modem mittels PPPoE-Verbindung aufgebaut, die Verbindung auf zwei Teilnetze an eth1 und eth2 weitergeleitet (geroutet). Die gezeigten MAC-Geräteadressen sind natürlich nur Beispiele.

Die Konfiguration kann mittels geeigneter WLAN-Hardware um einen Access-Point ergänzt und so zu einem vollwertigen **WLAN-Router** [https://wiki.ubuntuusers.de/WLAN_Router] ausgebaut werden. Außerdem kann über einen Transparenten **Proxy-Server** [http://de.wikipedia.org/wiki/wiki /Proxy_(Rechnernetz)] wie z.B. **Squid** [https://wiki.ubuntuusers.de/Squid] in Kombination mit **SquidGuard** [https://wiki.ubuntuusers.de/squidGuard] ein Daten-Cache und Web-Filter eingerichtet werden. Squid **Beispielkonfigurationen** [https://wiki.ubuntuusers.de/Internetverbindungsfreigabe#Proxyserver].

Aufgaben

Zusätzlich sollen folgende Vorgaben erfüllt werden:

- in Teilnetz 1 (eth1) soll dem Rechner mit der MAC-Adresse 00:24:21:0e:8f:4a **Reverse-VNC** [https://wiki.ubuntuusers.de/VNC#VNC-reverse-connection] über Port 5500 ermöglicht werden
  - Um Reverse-VNC effektiv nutzen zu können, kann ein entsprechender **DynDNS-Client** [https://wiki.ubuntuusers.de/DynDNS-Clients] eingerichtet werden.
- in Teilnetz 1 (eth1) soll einer angeschlossenen Spielekonsole (PlayStation 3) der Zugriff auf das PlayStation-Netzwerk ermöglicht werden
- in Teilnetz 2 (eth2) soll die Verwendung eines **BitTorrent** [https://wiki.ubuntuusers.de/BitTorrent]-Clients

gesperrt werden
- in Teilnetz 2 (eth2) soll einer angeschlossenen Spielekonsole (xBox) der Zugriff auf das Online-Portal xBox-Live ermöglicht werden
- für beide Teilnetze soll eine Bandbreitenverteilung mittels **Traffic-Controll** [https://wiki.ubuntuusers.de/Skripte/Traffic-Shaping] für eine gleichmäßige Last sorgen und Blockaden verhindern

Für eine einfache Bandbreitenverteilung bedienen wir uns hier des Skripts **Wondershaper** [http://lartc.org/wondershaper/] , welches in den Paketquellen zur Verfügung steht.

- **wondershaper** (*universe* - Traffic-Control des Kernels (tc) einfach konfigurieren)

Jetzt installieren [apt://wondershaper] mit **apturl** [https://wiki.ubuntuusers.de/apturl]

```
sudo apt-get install wondershaper
```

Syntax:

```
wondershaper clear [interface]                # Einstellungen zurücksetzen
wondershaper [interface] [downlink] [uplink]
```

Die Angaben erfolgen in Kilobit pro Sekunde. Wir gehen hier im Beispiel von einem ADSL2+-Anschluß mit 12000 kbits/s effektiver Empfangsrate (engl. "download rate") und 1000 kbit/s Senderate (engl. "upload rate") aus.

Konfiguration

Konfiguration der **/etc/network/interfaces**:

```
auto lo
iface lo inet loopback

# LAN-Anschluß 1
# an eth0 ist das DSL-Modem angeschlossen
auto eth0
iface eth0 inet manual

# pppoe Verbindung
auto dsl-provider
iface dsl-provider inet ppp
pre-up /sbin/ifconfig eth0 up   # line maintained by pppoeconf
provider dsl-provider

# lokales Netzwerk LAN-Anschluß 2
# Teilnetz 1
auto eth1
iface eth1 inet static
```

```
address 192.168.3.1
netmask 255.255.255.0
broadcast 192.168.3.255

# lokales Netzwerk LAN-Anschluß 3
# Teilnetz 2
auto eth2
iface eth2 inet static
address 192.168.4.1
netmask 255.255.255.0
broadcast 192.168.4.255

# Maskieren der LAN-Schnittstelle, Port-Forwarding & NAT aktivieren
# vorhandene Regeln und Ketten zuerst löschen / Reset Traffic-Control
# Restart-Funktionalität

up /sbin/iptables -F
 up /sbin/iptables -X
  up /sbin/iptables -t nat -F
   up /usr/sbin/wondershaper clear eth1
    up /usr/sbin/wondershaper clear eth2

# Forwarding für alle verwendeten Schnittstellen im lokalen Netz aktivieren
up /sbin/iptables -A FORWARD -o ppp0 -s 192.168.0.0/16 -m conntrack --ctstate NEW
-j ACCEPT
 up /sbin/iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
  up /sbin/iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
   up /sbin/sysctl -w net.ipv4.ip_forward=1

# dnsmasq neu starten
up /etc/init.d/dnsmasq restart

# Traffic-Control
up /usr/sbin/wondershaper eth1 500 6000
 up /usr/sbin/wondershaper eth2 500 6000

## Portfreigaben

# Port TCP 5500 an IP-Adresse 192.168.3.10 (Reverse-VNC)
up /sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 5500 -j DNAT
--to-destination 192.168.3.10
 up /sbin/iptables -A FORWARD -i eth0 -d 192.168.3.10 -p tcp --dport 5500 -j ACCEPT
```

```
# Ports UDP 3478,3479,3658  / TCP 80,443,5223 an 192.168.10.253 (Playstation 3)
up /sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp -m multiport --dports
80,443,5223 -j DNAT --to-destination 192.168.3.20
 up /sbin/iptables -t nat -A PREROUTING -i eth0 -p udp -m multiport --dports
3478,3479,3658 -j DNAT --to-destination 192.168.3.20
  up /sbin/iptables -A FORWARD -i eth0 -d 192.168.3.20 -p tcp -m multiport
--dports 80,443,5223 -j ACCEPT
   up /sbin/iptables -A FORWARD -i eth0 -d 192.168.3.20 -p udp -m multiport
--dports 3478,3479,3658 -j ACCEPT

# Ports UDP 88, 3074 / TCP 3074 an IP-Adresse 192.168.4.10 (xBox)
up /sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 3074 -j DNAT
--to-destination 192.168.4.10
 up /sbin/iptables -t nat -A PREROUTING -i eth0 -p udp -m multiport --dports
88,3074 -j DNAT --to-destination 192.168.4.10
  up /sbin/iptables -A FORWARD -i eth0 -d 192.168.4.10 -p tcp --dport 3074 -j
ACCEPT
```

Basiskonfiguration der **/etc/dnsmasq.conf**:

```
sudo cp /etc/dnsmasq.conf /etc/dnsmasq.conf.bak         # Originaldatei vorab
sichern
```

Basiskonfiguration:

```
# DHCP-Server aktiv für Interface
interface=eth1
interface=eth2

# DHCP-Server nicht aktiv für Interface
no-dhcp-interface=eth0

# IP-Adressbereich / Lease-Time
dhcp-range=interface:eth1,192.168.3.20,192.168.3.50,infinite   # Verteile an eth1
(Teilnetz 1) Adressen 192.168.3.20..192.168.3.50
dhcp-range=interface:eth2,192.168.4.20,192.168.4.50,infinite   # Verteile an eth2
(Teilnetz 2) Adressen 192.168.4.20..192.168.4.50

# feste IP-Adressen definieren
# PC in Teilnetz 1 / bekommt Portfreigabe für Reverse-VNC
dhcp-host=00:24:21:0e:8f:4a,192.168.3.10,infinite

# PS3 in Teilnetz 1 / bekommt Portfreigabe für PS-Netzwerk
dhcp-host=00:16:42:1e:3e:4e,192.168.3.20,infinite
```

Statische IP-Adressen können auch auf den Clients eingestellt werden, da es dann aber zu doppelt vergebenen Adressen kommen könnte, sollte eine automatische Konfiguration mit **DHCP** [http://de.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol] und einer Konfiguration über **dnsmasq** bevorzugt werden.

Funktionsprüfung

Nun kann das Netzwerk neu gestartet werden:

```
sudo /etc/init.d/networking restart
```

Fehlerfreie Ausgabe nach einem Neustart der Netzwerkkonfiguration:

```
* Reconfiguring network interfaces ...
# die automatisch konfigurierte DSL-Verbindung über ''pppoe'' ist hier ausgeblendet
...
net.ipv4.ip_forward = 1
 * Restarting DNS forwarder and DHCP server dnsmasq
   ...done.
```

Die erzeugten Filtertabellen:

```
sudo iptables -t nat -n -L -v
```

Ausgabe für die Kette "POSTROUTING" passend zum Beispiel:

```
Chain POSTROUTING (policy ACCEPT 180 packets, 17217 bytes)
 pkts bytes target     prot opt in     out     source
destination
  62  2734 MASQUERADE  all  --  *      eth0    0.0.0.0/0            0.0.0.0/0
    0     0 DNAT       tcp  --  eth0   *       0.0.0.0/0
0.0.0.0/0           multiport dports 80,443,5223 to:192.168.3.20
    0     0 DNAT       udp  --  eth0   *       0.0.0.0/0
0.0.0.0/0           multiport dports 3478,3479,3658 to:192.168.3.20
    0     0 DNAT       tcp  --  eth0   *       0.0.0.0/0
```

Ausgabe für die Kette "PREROUTING" passend zum Beispiel:

```
Chain PREROUTING (policy ACCEPT 4 packets, 246 bytes)
 pkts bytes target     prot opt in     out     source
destination
    0     0 DNAT       tcp  --  eth0   *       0.0.0.0/0
0.0.0.0/0           tcp dpt:5500 to:192.168.3.10
    0     0 DNAT       tcp  --  eth0   *       0.0.0.0/0
0.0.0.0/0           multiport dports 80,443,5223 to:192.168.3.20
    0     0 DNAT       udp  --  eth0   *       0.0.0.0/0
0.0.0.0/0           multiport dports 3478,3479,3658 to:192.168.3.20
```

```
    0     0 DNAT       tcp  --  eth0   *       0.0.0.0/0
0.0.0.0/0           tcp dpt:3074 to:192.168.4.10
    0     0 DNAT       udp  --  eth0   *       0.0.0.0/0
0.0.0.0/0           multiport dports 88,3074 to:192.168.4.10
```

```
sudo iptables -t filter -n -L -v
```

Ausgabe für die Kette "FORWARD", passend zum Beispiel:

```
Chain FORWARD (policy ACCEPT 13 packets, 8162 bytes)
 pkts bytes target     prot opt in     out     source
destination
    0     0 ACCEPT     all  --  *      eth0    192.168.0.0/16
0.0.0.0/0           ctstate NEW
    7   842 ACCEPT     all  --  *      *       0.0.0.0/0
0.0.0.0/0           ctstate RELATED,ESTABLISHED
    0     0 ACCEPT     tcp  --  eth0   *       0.0.0.0/0
192.168.3.10        tcp dpt:5500
    0     0 ACCEPT     tcp  --  eth0   *       0.0.0.0/0
192.168.3.20        multiport dports 80,443,5223
```

Die Anzahl der übertragenen Datenpakete und Bytes ändert sich natürlich. Bei korrekter Konfiguration können die angeschlossenen Clients nun eine Verbindung über DHCP herstellen.

Für eine erweiterte Konfiguration und die Einrichtung spezieller Filtertabellen bietet sich **nathelper** [https://wiki.ubuntuusers.de/Skripte/nathelper], **ufw** [https://wiki.ubuntuusers.de/ufw] oder auch **anfd** [https://wiki.ubuntuusers.de/Skripte/anfd] an. Entsprechende Befehle können ebenfalls in die Datei **interfaces** übernommen werden. Ebenso lassen sich diverse **Serverdienste** [https://wiki.ubuntuusers.de/Serverdienste] einrichten. Beispielkonfiguration für einen **MAC-Adressfilter** [https://wiki.ubuntuusers.de/Internetverbindungsfreigabe#MAC-Adressfilter-einrichten].

Systemlog abfragen

Ereignisse zu dnsmasq werden im Systemlogbuch unter **/var/log/syslog** abgelegt und können mit einem entsprechenden Filter ausgelesen werden:

```
grep dnsmasq /var/log/syslog
```

Filterregeln auslagern

Die erstellten iptables-Filterregeln werden sehr schnell recht umfangreich. Um eine bessere Übersicht zu behalten und die erstellten Regeln testen zu können, ohne das Netzwerk jedesmal neu starten zu müssen, bietet es sich an, alle Regeln und Aufrufe der Dienste in ein Skript auszulagern.

Dazu erstellt man die Datei **/usr/sbin/my_iptables_rules.sh** und macht diese anschließend noch ausführbar

```
touch my_iptables_rules.sh
chmod +x my_iptables_rules.sh
```

Inhalt passend zum Beispiel:

```
#!/bin/bash
## Iptables Regeln
## sonstiges

# vorhandene Regeln und Ketten zuerst löschen / Reset Traffic-Control
# Restart-Funktionalität

/sbin/iptables -F
 /sbin/iptables -X
  /sbin/iptables -t nat -F
   /usr/sbin/wondershaper clear eth1
    /usr/sbin/wondershaper clear eth2

# Forwarding für alle verwendeten Schnittstellen im lokalen Netz aktivieren
/sbin/iptables -A FORWARD -o ppp0 -s 192.168.0.0/16 -m conntrack --ctstate NEW -j
ACCEPT
 /sbin/iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
  /sbin/iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
   /sbin/sysctl -w net.ipv4.ip_forward=1

# dnsmasq neu starten
/etc/init.d/dnsmasq restart

# Traffic-Control
/usr/sbin/wondershaper eth1 500 6000
 /usr/sbin/wondershaper eth2 500 6000

## Portfreigaben

# Port TCP 5500 an IP-Adresse 192.168.3.10 (Reverse-VNC)
/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 5500 -j DNAT
--to-destination 192.168.3.10
 /sbin/iptables -A FORWARD -i eth0 -d 192.168.3.10 -p tcp --dport 5500 -j ACCEPT

# Ports UDP 3478,3479,3658  / TCP 80,443,5223 an 192.168.10.253 (Playstation 3)
/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp -m multiport --dports
80,443,5223 -j DNAT --to-destination 192.168.3.20
 /sbin/iptables -t nat -A PREROUTING -i eth0 -p udp -m multiport --dports
```

```
3478,3479,3658 -j DNAT --to-destination 192.168.3.20
  /sbin/iptables -A FORWARD -i eth0 -d 192.168.3.20 -p tcp -m multiport --dports
80,443,5223 -j ACCEPT
   /sbin/iptables -A FORWARD -i eth0 -d 192.168.3.20 -p udp -m multiport --dports
3478,3479,3658 -j ACCEPT
```

Durch direkten Aufruf kann die Funktion nun getestet werden:

```
sudo ./my_iptables_rules.sh
```

Funktioniert alles einwandfrei, wird das Skript nach **/usr/sbin/** kopiert:

```
sudo cp my_iptables_rules.sh /usr/sbin
```

Die Einträge in der **/etc/network/interfaces** können natürlich entfallen. Stattdessen muss nach der Konfiguration der einzelnen Schnittstellen nur der Verweis auf das Skript eingetragen werden.

```
...
post-up /usr/sbin/my_iptables_rules.sh
```

Fernzugriff / Administration

Um das System von einem anderen Rechner aus verwalten zu können, bieten sich **SSH** [https://wiki.ubuntuusers.de/SSH] oder auch ein Zugriff über **VNC** [https://wiki.ubuntuusers.de/VNC] an. Der Zugriff von einen Windows-System aus wäre mit dem Programm **PuTTY** [https://wiki.ubuntuusers.de/PuTTY] über eine SSH-Verbindung möglich.

Spezielle Distributionen

Es gibt einige Linux-Distributionen, die speziell für diesen Anwendungszweck konzipiert wurden und die benötigten Komponenten bereits integriert haben oder sich durch Zusatzmodule ("Add-Ons") entsprechend erweitern lassen. Die Verwaltung erfolgt z.B. bequem über eine Weboberfläche.

- **ipfire** [http://www.ipfire.org/]
- **fli4l** [http://www.fli4l.de/] - **Screenshots** [http://www.fli4l.de/home/galerie/screenshots/]
- **IPCop** [http://www.ipcop.org/] - **Screenshots** [http://sourceforge.net/apps/trac/ipcop/wiki/Screenshots]
- **ClearOS** [http://www.clearfoundation.com/Software/overview.html] - **Live-Demosystem** [http://www.clearfoundation.com/Software/live-demo.html]
- **Coyote Linux** [http://coyotelinux.com/] - **Screenshots** [http://coyotelinux.com/screenshots/]
- **m0n0wall** [http://m0n0.ch/wall/] - **Screenshots** [http://m0n0.ch/wall/screenshots.php]

Weitere Informationen und vergleichbare Distributionen findet man auf **distrowatch.com** [http://distrowatch.com/search.php?category=Firewall&origin=All&basedon=All&notbasedon=None&desktop=All&architecture=All&status=Active]

Zusatzinformationen

- **Ports/registrierte Portnummern** [http://de.wikipedia.org/wiki/Port_(Protokoll)]
- **Dnsmasq** [https://wiki.ubuntuusers.de/Dnsmasq]
- **sysctl** [http://www.freebsd.org/doc/de/books/handbook/configtuning-sysctl.html] - Artikel aus dem Linux-Handbuch auf freebsd.org
- **sysctl** [http://www.linux-praxis.de/lpic2/lpi201/manpages/sysctl.html] - Kurzbeschreibung auf linux-praxis.de

Freie DNS

Siehe auch **DNS** [http://de.wikipedia.org/wiki/Domain_Name_System]:

- **Liste freier DNS auf wiki.ak-zensur.de** [http://wiki.ak-zensur.de/index.php/Unzensierte_DNS_Server]
- **Liste freier DNS www.friendly-board.de** [http://www.friendly-board.de/forum/freie-dns-server-t552.html]

Links

Intern

- **interfaces** [https://wiki.ubuntuusers.de/interfaces]
- **iptables** [https://wiki.ubuntuusers.de/iptables2]
- **WLAN Router** [https://wiki.ubuntuusers.de/WLAN_Router]
- **Internetverbindungsfreigabe** [https://wiki.ubuntuusers.de/Internetverbindungsfreigabe]
- **Portweiterleitung** [https://wiki.ubuntuusers.de/Portweiterleitung]

Extern

- **Internet Connection Sharing** [https://help.ubuntu.com/community/Internet/ConnectionSharing#Ubuntu%209.10%20Method] - Ubuntu-Dokumentation
- **Arno's IPTABLES Firewall Script** [http://freecode.com/projects/iptables-firewall] - Router-Installationsskript
- **GRASE Hotspot** [http://grasehotspot.org/] - stellt in Kombination mit einem Hardware-Router und einem Ubuntu-Server einen WLAN-Hotspot zur Verfügung

---

Diese Revision [https://wiki.ubuntuusers.de/Router?rev=818923] wurde am 24. Mai 2015 02:43 von **ubot** erstellt.
Die folgenden Schlagworte wurden dem Artikel zugewiesen: **Netzwerk** [https://wiki.ubuntuusers.de/Wiki/Tags?tag=Netzwerk], **Internet** [https://wiki.ubuntuusers.de/Wiki/Tags?tag=Internet]