Wiki / Dnsmasq

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 erlangen**

Inhaltsverzeichnis

[//media-cdn.ubuntu-de.org/wiki/attachments/59/28/service.png] **Dnsmasq** [http://thekelleys.org.uk/dnsmasq/doc.html] ist ein einfacher DNS- und DHCP-Server für kleine Netzwerke. Es werden die Namen aus dem lokalen Netz entsprechend der Datei **/etc/hosts** [https://wiki.ubuntuusers.de/hosts] aufgelöst. Unbekannte Namensanfragen werden weitergeleitet und im Cache gespeichert.

Dnsmasq ist somit:

- ein Nameserver-Forwarder
- ein DHCP-Server
- optimiert für Netzwerke mit Dialup-Außenverbindung

Installation

Das Programm ist in den offiziellen Paketquellen enthalten. Es kann über das folgende Paket installiert [1] werden:

- **dnsmasq** (*universe*)

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

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

Konfiguration als DNS-Server

Dnsmasq greift zur lokalen Namensauflösung auf die Datei **/etc/hosts** [https://wiki.ubuntuusers.de/hosts] zurück. Daher sollte man diese mit einem Editor [3] und Root-Rechten bearbeiteten [3] und Zeilen in dieser Art hinzufügen.

```
## Lokales Netzwerk
#  = IP =     =  Domainname =    = Rechnername =
192.168.1.1     router.foo.bar       router          ## Router
192.168.1.2     neo.foo.bar          neo             ## Workstation
192.168.1.3     felidae.foo.bar      felidae         ## ein Client-Rechner
192.168.1.5     link.foo.bar         link            ## Notebook
192.168.1.11    zeus.foo.bar         zeus            ## Der Server
192.168.1.66    bridge.foo.bar       bridge          ## WLAN-Brücke
```

Als Beispiel-Domain wird *"foo.bar"* genutzt. Üblicherweise besitzt man in einem LAN keine echte Domain, man sollte hier eine Bezeichnung nutzen, die nicht im Internet vorkommt. Als Schutz vor Namenskollisionen empfiehlt die **ICANN** [http://www.icann.org] jedoch, besser eigene FQDN zu nutzen (z.B. beispiel.de) und dann für interne Netze eine Subdomain (z.B. intranet.beispiel.de) zu verwenden (**PDF** [http://www.icann.org/en/about/staff/security/ssr/name-collision-mitigation-05dec13-en.pdf]). Weiterhin sollte die Endung *".local"* nicht verwendet werden, da diese von **Avahi** [https://wiki.ubuntuusers.de/Avahi] benutzt wird und es dabei zu Konflikten kommen kann. Wird im Netzwerk dennoch diese Endung vergeben und kommt es genau dabei zu keiner erfolgreichen Auflösung, so lässt sich Avahi zur Analyse mit folgendem Befehl vorübergehend beenden:

```
sudo service avahi-daemon stop
```

Verwendung des Cache

Damit alle DNS-Anfragen über Dnsmasq verarbeitet und bei Bedarf an externe DNS-Server weitergereicht werden, ist der einfachste Weg, die Datei **/etc/resolv.conf** zu bearbeiten. Dabei ist sicher zu stellen, dass der lokale DNS-Server (Dnsmasq) als erste Zeile eingetragen wird (127.0.0.1). Als nächste Zeile muss mindestens ein weiterer DNS-Server definiert werden, damit Dnsmasq die Anfragen weiterreichen kann. Beispiel mit Weiterleitung an den DNS-Cache der Fritz-Box:

```
nameserver 127.0.0.1
nameserver 192.168.178.1
```

Achtung: Bei der Verwendung von **Wicd** [https://wiki.ubuntuusers.de/Wicd] wird **resolv.conf** automatisch verwaltet bzw. überschrieben, daher müssen statische DNS in genau dieser Reihenfolge festgelegt werden.

Bei Verwendung des **NetworkManager** [https://wiki.ubuntuusers.de/NetworkManager] darf **resolv.conf** nicht manuell bearbeitet werden. Stattdessen kann die Verbindung im Netzwerkmanager auf *"Automatisch (DHCP), nur Adressen"* oder auf *"manuell"* gesetzt werden. Dann können die IP-Adressen von bis zu drei DNS-Servern - durch Kommata getrennt - in das vorgesehene Feld eingetragen werden. Werden die Änderungen gespeichert,

werden diese Einträge vom Netzwerkmanager in die Datei **resolv.conf** geschrieben.

Das erfolgreiche lokale Cachen überprüft man am Terminal:

```
$ dig ubuntuusers.de
```

antwortet am Ende mit

```
;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
```

Soll Dnsmasq wirklich nur als DNS-Server für den eigenen Rechner arbeiten, braucht man nichts mehr zu konfigurieren.

Dnsmasq als DNS-Server für andere Rechner im lokalen Netz

Möchte man den DNS-Server auch für andere Rechner im lokalen Netz nutzen, dann muss man Dnsmasq dazu bringen, dass es auch an der Netzwerkschnittstelle nach DNS-Abfragen lauscht. Damit die Clients wissen, wie sie den DSN-Server finden, muss der Rechner, auf dem Dnsmasq läuft, dann mit einer statischen Adresse versehen sein.

/etc/dnsmasq.conf muss mit Administrator-Rechten bearbeitet werden und folgende Zeilen werden geändert bzw. hinzugefügt:

```
listen-address=127.0.0.1
listen-address=aaa.bbb.ccc.ddd
```

wobei aaa.bbb.ccc.ddd die lokale statische IP-Nr. des Dnsmasq-Rechners ist, also z.B. 192.168.0.20; es muss sowohl die 127.0.0.1 als auch die aaa.bbb.ccc.ddd angegeben werden, damit Dnsmasq sowohl eigene als auch fremde Anfragen bearbeitet. Hat der Rechner mehrere IP-Nrn. muss man - sofern gewünscht oder nötig - auch die zusätzlichen IP-Nrn. angeben. Damit kann man z.B. für ein ganzes Heim-Netzwerk, ein Büro oder auch eine grössere Firma zentral die Zugriffe zum Internet regeln (s. unten).

Ob das funktioniert, kann man mit dem Konsolenbefehl **sudo netstat -tulpen | grep dnsmasq** überprüfen, man sieht dann in der Abteilung LISTEN, ob Dnsmasq auf 127.0.0.1 **und** aaa.bbb.ccc.ddd lauscht.

Rechner im LAN, die diesen DNS-Server nutzen wollen, sollten danach so konfiguriert werden, dass sie den Server auch nutzen. Bei Windows-Systemen geht dies in den Netzwerkeinstellungen, und auch bei Linux-Clients lässt sich der zu nutzende DNS-Server leicht **einstellen** [https://wiki.ubuntuusers.de/IP-Adresse_wechseln].

Umleiten und Manipulieren von DNS-Abfragen

Umleiten

Das Umleiten von DNS-Abfragen kann nötig oder sinnvoll sein, wenn man sich in einem Firmennetzwerk mit eigenem Nameserver befindet. Man möchte vielleicht nicht, dass mittels Logs auf dem DNS-Server mitprotokolliert werden kann, wo man herumsurft, braucht aber dennoch die Namensauflösung fürs Firmenintranet. Alle Seiten, die z.B. auf *.intranet* enden, sollen vom firmeninternen DNS-Server behandelt werden. Alle anderen Namen sollen von einem externen Nameserver aufgelöst werden.

Unter dnsmasq ist es einfach möglich, für verschiedene Namensbereiche explizit einen oder mehrere DNS-Server zu bestimmen. Dazu trägt man entsprechende Zeilen in die Datei **/etc/dnsmasq.conf** ein:

```
server=/intranet/10.17.1.1
```

Dieser Eintrag würde alle DNS-Anfragen für die Toplevel-Domain *.intranet*, also Seiten wie *firmenportal.intranet* an den Server *10.17.1.1* schicken. Alle übrigen Anfragen z.B. an *ubuntuusers.de* würden von dem Nameserver bearbeitet werden, der in der Datei **/etc/resolv.conf** eingetragen ist.

Hier zwei weitere mehr oder weniger sinnvolle Beispiele:

```
server=/google.com/8.8.8.8 # leitet alle DNS-Anfragen für google.com-Seiten an den
Google Nameserver weiter
server=/de/213.73.91.35 # leitet alle DNS-Anfragen für .de-Seiten an den Nameserver
des Berliner CCC weiter
```

Manipulieren

Es gibt verschiedene Gründe, warum man DNS-Abfragen manipulieren möchte. Ein Einsatzgebiet ist das Ausblenden von Werbung oder der Schutz der Privatsphäre. Der Ansatz ist hierbei der einer Blacklist. Das heißt, dass man eine Liste an Adressen pflegen muss, die man umleiten bzw. manipulieren möchte.

Beispielsweise könnte man den Zugriff auf Facebook und das Einblenden von *Like*-Buttons verhindern wollen. Dazu trägt man folgende zwei Zeilen in die Datei **/etc/dnsmasq.conf** ein und startet dann dnsmasq neu.

```
address=/facebook.com/127.0.0.1
address=/fbcdn.net/127.0.0.1
```

Dadurch werden alle DNS-Anfragen für *facebook.com*, *fbcdn.net* (das Facebook Content Delivery Network) und deren Subdomains (z.B. *www.facebook.com*) so manipuliert, dass die Anfragen auf den eigenen Rechner umgeleitet werden. Dort laufen sie dann ins Leere.

Werbenetzwerke (z.B. Double Click) und Statistikdienste (z.B. Google Analytics) kann man auf die gleiche Art und Weise aus seinem Netzwerk fernhalten:

```
address=/doubleclick.net/127.0.0.1
address=/google-analytics.com/127.0.0.1
```

Der Vorteil dieser Lösung ist eine zentrale Stelle, an der die Filterregeln angewendet werden, solange jeder Computer im Netzwerk diesen DNS-Server nutzt. So muss nicht jeder Nutzer an jedem Computer im Netzwerk in jedem Browser einen Werbeblocker installieren.

Nachteilig ist jedoch der manuelle Aufwand, da die Blacklist nicht automatisch gepflegt wird. Außerdem kann der Zugriff auf z.B. Facebook nicht verhindert werden, wenn direkt die IP genutzt wird, um auf die Webseite zu gelangen.

Damit DNS-Anfragen nicht weiter aufgelöst werden sollen, z.B. interne Domains wie .local oder fritz.box, kann man dies auch konfigurieren:

```
local=/localnet/
```

```
local=/fritz.box/
```

Alle DNS-Anfragen zu *.local oder *.fritz.box werden nur anhand der **/etc/hosts** ausgewertet und verlassen den Rechner nicht.

Konfiguration als DHCP-Server

Soll Dnsmasq zusätzlich noch als DHCP-Server fungieren, so muss man noch die Datei **/etc/dnsmasq.conf** entsprechend bearbeiten [3]. Üblicherweise braucht hier nicht viel eingestellt zu werden. Die wichtigsten Optionen werden im folgenden vorgestellt.

IP-Bereich

```
# Uncomment this to enable the integrated DHCP server, you need
# to supply the range of addresses available for lease and optionally
# a lease time. If you have more than one network, you will need to
# repeat this for each network on which you want to supply DHCP
# service.
dhcp-range=192.168.0.100,192.168.0.254,12h  ## 12h ist die Lease-Time
```

In diesem Abschnitt wird festgelegt, aus welchem Bereich der DHCP-Server Adressen vergibt. Üblicherweise kann man hier IP-Bereiche eintragen, die für **Private IP-Adressen** [http://de.wikipedia.org/wiki/Private_IP-Adresse] vorgesehen sind. Im angegebenen Beispiel würde der DHCP-Server Adressen aus dem Bereich *"192.168.0.100"* bis *"192.168.0.254"* vergeben und die Zuordnung Rechner <-> IP jeweils 12 Stunden lang beibehalten. Die Adressen *"192.168.0.0"* und *"192.168.0.255"* dürfen keinem Endgerät zugeordnet werden, da dies die Netz- bzw. Broadcast Adressen sind.

Feste IP-Adressen zuordnen

Beispiel 1

Für manche Anwendungen ist es praktisch, wenn ein Rechner immer unter der selben IP-Adresse zu erreichen ist. Wenn man dies möchte, so kann man der **MAC-Adresse** [http://de.wikipedia.org/wiki/MAC-Adresse] der Netzwerkkarte des betroffenen Rechners eine feste IP-Adresse zuordnen.

```
dhcp-host=00:07:95:26:2B:C9,neo,192.168.1.2,infinite
```

In diesem Beispiel würde also der Netzwerkkarte mit der MAC-Adresse 00:07:95:26:2B:C9 immer der Name neo und die IP-Adresse 192.168.1.2 für eine unbestimmte Zeit zugewiesen.

Beispiel 2

Benutzt man zum Beispiel ein Notebook, das im Wechsel mal per Kabelnetzwerk, mal via WLAN vernetzt ist, wird dies anhand der MAC-Adresse erkannt und eine freie IP-Adresse vergeben.

```
dhcp-host=00:00:0E:D2:DA:BE,link,infinite
dhcp-host=00:02:2d:a9:55:92,link,infinite
```

Zahlreiche weitere Beispiele und Konfigurationsmöglichkeiten sind in der **/etc/dnsmasq.conf** selber aufgeführt.

DHCP-Server Optionen

Der DHCP-Server kann seinen Client-Rechnern nicht nur IPs zuweisen, sondern auch alle notwendigen Netzwerkinformationen übermitteln. Diese und noch weitere Einstellungen lassen sich ebenfalls in der **/etc/dnsmasq.conf** setzen.

Gateway

```
## Syntax: erst die Option, dann die entsprechende Adresse
dhcp-option=3,192.168.1.1
```

Die IP-Adresse des **Gateways** [http://de.wikipedia.org/wiki/Gateway_(Computer)], um Daten in das Internet schicken zu können. Läuft Dnsmasq auf demselben Rechner, der sich auch in das Internet einwählt, so ist das die lokale IP-Adresse des Rechners.

Mehrere Gateways in einem Netzwerk

Um mehrere Gateways (Internetverbindungen) in einem Netzwerk zu betreiben werden den einzelnen Gateways Namen (Tags) hinzugefügt. Diese wiederum werden den einzelnen Clienten zugeteilt. So erreicht man, dass jeder Client im Netzwerk eine bestimmte Internetverbindung verwendet.

```
# Gateways festlegen
dhcp-option=tag:gateway1,3,192.168.1.2
dhcp-option=tag:gateway2,3,192.168.1.3
```

```
# DHCP für Unbekannte
dhcp-range=set:gateway1,192.168.1.200,192.168.1.250,255.255.255.0,12h
```

```
# Clienten:
dhcp-host=set:gateway1,00:1A:4D:49:80:C2,192.168.1.111,12h # Client 192.168.1.111
verwendet 192.168.1.2 als Gateway
dhcp-host=set:gateway2,00:1A:4B:6F:C6:65,192.168.1.121,12h # Client 192.168.1.121
verwendet 192.168.1.3 als Gateway
```

NTP-Server

Der zu nutzende **NTP** [https://wiki.ubuntuusers.de/Systemzeit]-Zeitserver, um die exakte Uhrzeit aus dem Internet zu beziehen:

```
dhcp-option=42,130.133.1.10   # Timeserver time.fu-berlin.de
```

PXE-Server

Ein lokaler Rechner mit **PXE-Boot-Images** [http://de.wikipedia.org/wiki/Preboot_Execution_Environment], um über das Netzwerk den Rechner booten zu können:

```
dhcp-boot=/pxelinux.0,zeus,192.168.1.11
```

Loglevel

```
log-queries
```

Für Tests kann man den Loglevel erhöhen, also die Menge der Informationen, die **mitgeloggt** [https://wiki.ubuntuusers.de/Logdateien] wird. So kann man bei Fehlern oder Problemen schnell auf die Lösung kommen. Welche IP-Adressen Dnsmasq gerade vergeben hat kann man in der Datei **/var/lib /misc/dnsmasq.leases** nachlesen.

Weitere DNS-Server

```
server=/localnet/IP-Adresse_des_MasterDNS
```

Sollten sich im Netzwerk noch weitere DNS-Server befinden, so kann man diese hier noch zusätzlich angeben.

Firewall Einstellungen

Wenn **ufw** [https://wiki.ubuntuusers.de/ufw] als Firewall läuft, dann müssen eingehende Verbindungen am bootps Port (67 wie in **/etc/services** festgelegt) erlaubt werden, damit Dnsmasq als DHCP Server fungieren kann:

```
$ sudo ufw allow bootps
$ sudo ufw disable && sudo ufw enable
```

Konfigurationsänderungen aktivieren

Nach einer Änderung von **/etc/dnsmasq.conf** muss dnsmasq neu gestartet werden, um die Änderungen wirksam werden:

```
sudo service dnsmasq restart
```

Links

- **dnsmasq** [http://wiki.debian.org/HowTo/dnsmasq] - HowTo im Debian-Wiki
- **Dnsmasq** [https://help.ubuntu.com/community/Dnsmasq] im Ubuntu-Wiki
- **dnsmasq** [http://linuxwiki.org/dnsmasq] - Tipps und Tricks von LinuxWiki.org

---

Diese Revision [https://wiki.ubuntuusers.de/Dnsmasq?rev=819733] wurde am 24. Mai 2015 21:16 von **GunnarScherf** erstellt.
Die folgenden Schlagworte wurden dem Artikel zugewiesen: **Netzwerk** [https://wiki.ubuntuusers.de/Wiki/Tags?tag=Netzwerk], **Server** [https://wiki.ubuntuusers.de/Wiki/Tags?tag=Server], **dhcp** [https://wiki.ubuntuusers.de/Wiki/Tags?tag=dhcp], **dns** [https://wiki.ubuntuusers.de/Wiki/Tags?tag=dns]