13 Domain Name-Server einrichten

IP-Adressen identifizieren Rechner im Internet eindeutig. Diese Art der Adressierung ist für Maschinen praktisch, aber nicht für Menschen. Diesen kommt das hierarchische System von Domain-Namen in der Form `www.linuxbu.ch` oder auch allgemeiner `Host.ServerDomain.TopLevelDomain` entgegen.

Mehr zum Aufbau von Domain-Namen finden Sie in Internet-Büchern wie *Linux-Wegweiser für Netzwerker* von Olaf Kirch und im Internet bei jedem NIC (s. u.).

Ruft jemand eine Webseite des Servers `www.linuxbu.ch` auf, so muss der Browser die IP-Nummer von `www.linuxbu.ch` herausfinden. Diese Aufgabe überlässt er dem Domain Name Service (DNS).

Jedes Programm, das einen Host-Namen mitgeteilt bekommt, versucht sofort, ihn in eine IP-Adresse aufzulösen. Dazu benutzen Internet-Clients folgendes Verfahren:

Zuerst suchen sie eine Datei `hosts`, bei Windows 9x im Windows-Verzeichnis (meist `c:\windows`), bei Windows XP/Vista unter `Windows\system32\drivers\etc`, bei Linux im Verzeichnis `/etc`. Dort prüfen sie, ob in der Datei zu dem Domain-Namen eine IP-Adresse steht. Wenn nicht, nehmen sie mit den DNS-Servern Kontakt auf, die auf dem Client in den Eigenschaften von IP als DNS-Server eingetragen sind.

Host-Dateien auf Clients lokal zu pflegen, ist sehr aufwändig. Daher nimmt man gern die Dienste von DNS-Servern in Anspruch.

13.1 Wann Sie einen eigenen Name-Server brauchen

Eigene Name-Server sollte man immer dann einrichten, wenn man ein lokales Netz an das Internet anbindet. Lokale Name-Server haben folgende Aufgaben:

- Verwalten der Namen für das lokale Netz (Hosting genannt),
- Weiterleiten der DNS-Anfragen an den DNS-Server des Providers (Caching).

13.2 So funktionieren das Domain Name System und die Internet-Domains

Bis 1984 pflegte das Network Information Center (NIC) diese Zuordnung in Form einer großen Tabelle. Als diese Liste zu groß wurde, hat die Netzgemeinde den hierarchischen Domain Name Service eingeführt.

Bisher gab es hauptsächlich zwei Arten von Top-Level-Domains: Die nationalen, die mit zwei Buchstaben ein Land identifizieren und die ursprünglichen, die jeweils aus drei Buchstaben bestehen.

Die beiden Arten von Top-Level-Domains werden verschieden verwaltet: nationale NICs – Network Information Centers (`www.nic.de`, `www.nic.at`, `www.nic.ch`, `www.nic.fr`) – verwalten die Landesdomains wie `de` (Deutschland), `at` (Österreich), `ch` (Schweiz) und `fr` (Frankreich).

Die Drei-Buchstaben-Domains aus der Anfangszeit des Internet (`com`, `edu`, `gov`, `mil`, `net`, `org`, `int`) verwalten inzwischen zahlreiche konkurrierende Firmen. Dadurch kommt es immer häufiger zu Pannen wie etwa der Doppelvergabe von Domänen.

Für die neuen Top-Level-Domains `biz`, `info` etc. konnten sich Firmen um die Domain-Verwaltung bewerben. Auch wenn die Vergabe nicht immer ganz transparent wurde, ist die eindeutige Zuständigkeit inzwischen geklärt.

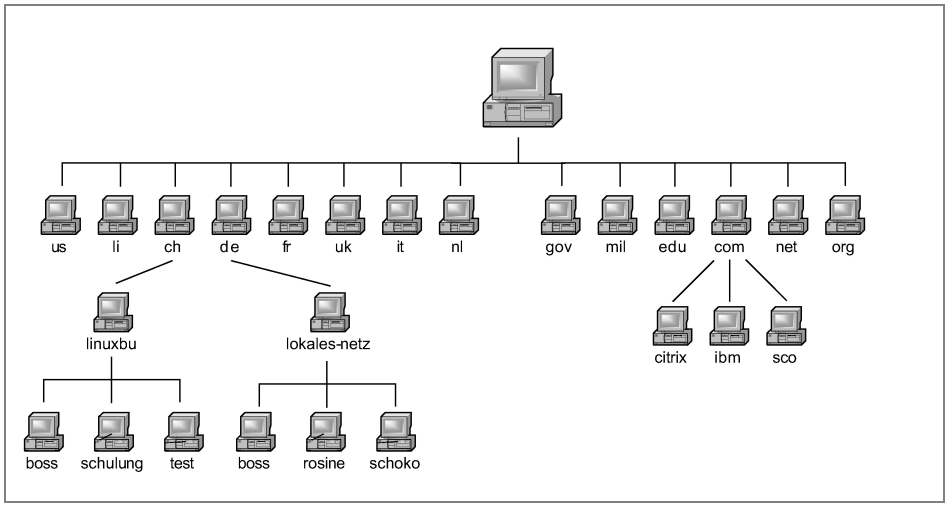

Abbildung 13.1: Baumstruktur

Der Ablauf einer Namensanfrage ist folgendermaßen:

1. Ruft jemand in den USA die Webadresse `www.linuxbu.ch` auf, so landet dessen Name-Server-Anfrage über Zwischenschritte beim zentralen Name-Server des NIC, dem *root*-Server.
2. Dieser übergibt die Anfrage an den Name-Server des Ch-NIC, der Sie dann an den für `linuxbu.ch` zuständigen Name-Server (`nameserv.deltaweb.de`) weitergibt, von wo er nun endgültig die IP-Adresse (`193.239.104.29`) bekommt.
3. Diese IP-Adresse geht dann auf dem langen Weg über die beteiligten Name-Server an den anfragenden Rechner weiter.

Da sich die meisten Name-Server Adressen in einem Cache merken, nehmen Anfragen nur selten diesen langen Weg. Dieser Cache hat aber auch den Nachteil, dass es je nach Konfiguration ein paar Tage dauern kann, bis der letzte Name-Server einen neuen Eintrag oder eine Änderung mitbekommen hat.

Zusätzlich zu diesen Anfragen, die zu einem Namen eine IP-Adresse ermitteln, muss ein Name-Server auch umgekehrt den Namen, der zu einer IP-Adresse gehört, ermitteln. (Reverse Lookup).

Auch die Zuordnung eines zuständigen Mail-Servers für einen Adressbereich erfolgt über den Name-Server. Hierfür sind die *mx-records* (Mail Exchanger-Einträge) zuständig.

13.2.1 Die Hosts-Datei

In kleineren Netzen ist ein eigener Name-Server nicht notwendig. Hier kann man die vorhandenen Rechner einfach in die `Hosts`-Datei eines jeden Rechners eintragen. Das Format dieser Datei ist für Linux und Windows identisch.

Bearbeiten können Sie die Datei entweder direkt mit einem Texteditor oder der Funktion im YaST-Kontrollzentrum, die Sie unter *Netzwerkdienste* • *Hostnamen* finden.

`/etc/hosts`

```
#
# hosts         This file describes a number of hostname-to-address
#               mappings for the TCP/IP subsystem.  It is mostly
#               used at boot time, when no name servers are running.
#               On small systems, this file can be used instead of a
#               "named" name server.
# Syntax:
#
# IP-Address  Full-Qualified-Hostname  Short-Hostname
#

127.0.0.1       localhost
```

```
# special IPv6 addresses
::1             localhost ipv6-localhost ipv6-loopback

fe00::0         ipv6-localnet

ff00::0         ipv6-mcastprefix
ff02::1         ipv6-allnodes
ff02::2         ipv6-allrouters
ff02::3         ipv6-allhosts


192.168.1.2     boss.lokales-netz.de    boss
```

Zumindest die Zeilen, die den lokalen Rechner beschreiben – hier die beiden hervorgehobenen Zeilen – müssen sich immer in der `Hosts`-Datei finden. So kann der Server zumindest seine eigenen Adressen immer auflösen.

Einen großen Teil der Datei können Sie ignorieren, er ist für die Erweiterung des IP-Adressformates auf 16 Byte bedeutsam.

13.2.2 Name-Server installieren und konfigurieren

Der Name-Server befindet sich bei OpenSUSE im Paket `bind` der Paketgruppe `Netzwerk`. Die Standardinstallation richtet das Paket nicht ein. Man muss dies also nachholen, bevor man den DNS konfiguriert.

Folgende Dateien sind für die beschriebene Konfiguration wichtig:

Datei	*Bedeutung*
`/usr/sbin/named`	Die Binärdatei, die den Name-Server bildet.
`/etc/hosts`	Liste mit IP-Adressen und zugehörigen Rechnernamen.
`/etc/host.conf`	Bestimmt die Art der Namensauflösung.
`/etc/resolv.conf`	Konfiguration für den Name Resolver (Namensauflöser).
`/etc/named.conf`	Hauptkonfigurationsdatei.
`/var/lib/named/root.hint`	Datei mit den *root*-Name-Servern (Standard-Nameserver).
`/var/lib/named/privat.zone`	Datei für die Namenszuordnung im lokalen Netz, der Dateiname ist frei wählbar, hier im Beispiel *privat*.
`/var/lib/named/localhost.zone`	Namenszuordnung für `localhost` im lokalen Netz.

Datei	*Bedeutung*
`/var/lib/named/tavirp.zone`	Umgekehrte Zuordnung von IP zu Name, der Name ist frei wählbar, hier `privat` in umgekehrter Reihenfolge.
`/var/lib/named/127.0.0.zone`	Umgekehrte Zuordnung `127.0.0.1` zu `localhost`.

Tabelle 13.1: Konfigurationsdateien des Name-Servers

Hinweis: Sie können den Name-Server erst starten, wenn Sie alle Konfigurationsdateien angelegt haben.

Damit der Rechner selber später auch auf den Name-Server zugreifen kann, sollte man zuerst das YaST-Kontollzentrum starten und dort unter *Netzwerkgeräte* • *Netzwerkeinstellungen* • *Hostname/DNS* die notwendigen Angaben machen. Hier gibt man die IP-Adresse (`192.168.1.2`) bzw. die IP-Adressen für den oder die Name-Server sowie den Domainnamen (`lokales-netz.de`) an. Wichtig ist, dass die Checkboxen für DHCP deaktiviert sind, da der Rechner seine Daten ja nicht von einem anderen System dynamisch beziehen soll.

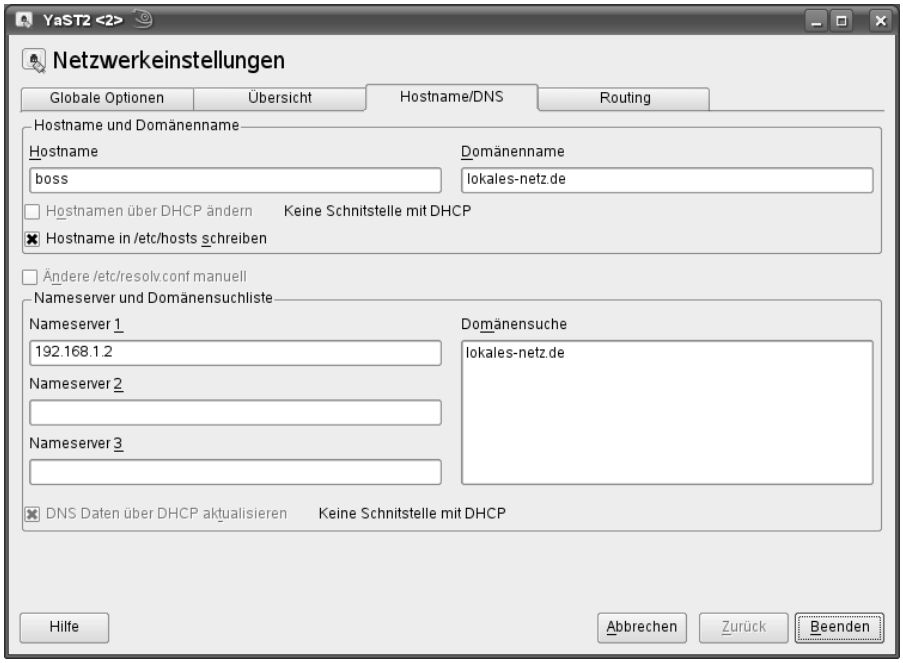

Abbildung 13.2: Konfiguration des Name-Servers

YaST erzeugt bzw. verändert dann die Dateien `/etc/host.conf` und `/etc/resolv.conf`.

`/etc/host.conf`

```
#
# /etc/host.conf - resolver configuration file
#
# Please read the manual page host.conf(5) for more information.
#
#
# The following option is only used by binaries linked against
# libc4 or libc5. This line should be in sync with the "hosts"
# option in /etc/nsswitch.conf.
#
order hosts, bind
#
# The following options are used by the resolver library:
#
multi on
```

Dies legt fest, wie Namen aufgelöst werden. Zuerst sehen die Dienste in der Datei `/etc/hosts` nach. Falls sie die gesuchte Adresse dort nicht finden, fragen Sie den Name-Server `bind`. Der Eintrag `multi on` bewirkt, dass man zu einem Rechnernamen in der `/etc/hosts` mehrere IP-Adressen angeben darf.

`/etc/resolv.conf`

```
search lokales-netz.de
nameserver 192.168.1.2
```

Die beiden Zeilen in dieser Datei bewirken, dass für die Suche nach Rechnern der Domain `lokales-netz.de` der Name-Server `192.168.1.2` befragt wird.

Der DNS-Server wertet beim Start die Konfigurationsdatei `named.conf` aus. Mit einem Texteditor legt man sie an und trägt in sie u. a. die Pfade und Namen aller weiteren Konfigurationsdateien ein.

YaST bietet unter *Netzwerkdienste* • *DNS-Server* ein Konfigurationstool für den Nameserver an. Sollte das Konfigurationstool nicht vorhanden sein, so installieren Sie das Paket `yast2-dns-server` aus der Paketgruppe `System` nach.

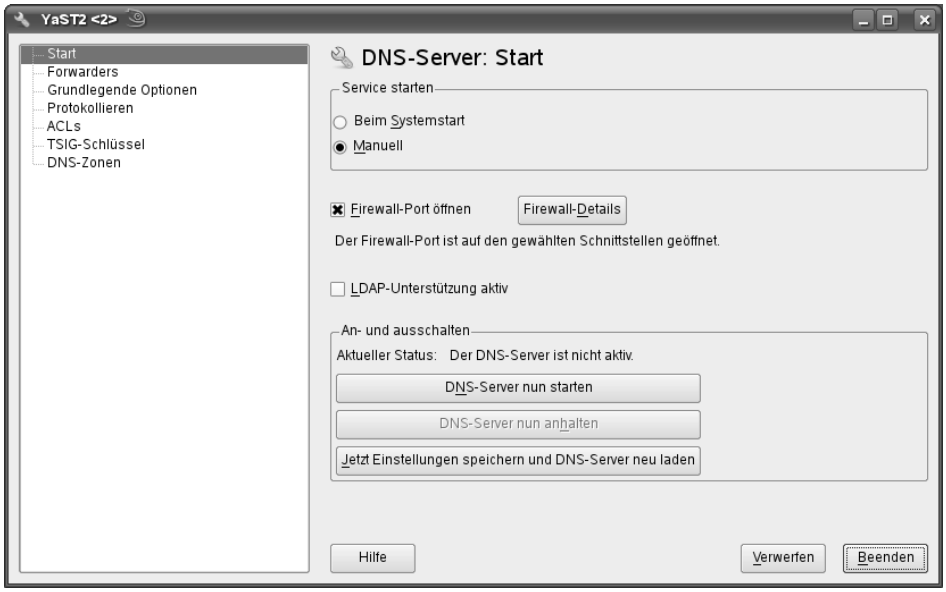

Abbildung 13.3: YaST: Name-Server

Wenn Ihr Nameserver auch übers Internet zugänglich sein soll, können Sie hier den zugehörigen Firewall-Port öffnen. Den Firewall-Port sollten Sie aber aus Sicherheitsgründen nur dann öffnen, wenn Ihr Nameserver unbedingt zugänglich sein muss.

Das YaST-Tool stellt nicht alle Funktionen zur Verfügung und andere Werkzeuge sind hier aufwändiger zu bedienen als der Texteditor. Für kleinere Netze kann der Einsatz von YaST sinnvoll sein, aber in größeren Netzen ist Handarbeit effektiver. Sie bildet auch die Grundlage für die folgende Beschreibung. Entscheiden Sie selbst, in wie weit der Einsatz dieses YaST-Tools für Ihre Einsatzumgebung sinnvoll ist.

Die von OpenSUSE installierten Musterdateien können Sie für Ihre Bedürfnisse anpassen. Eine kurze Dokumentation zum Name-Server `Bind` findet sich im Ordner `/usr/share/doc/packages/bind`.

`/etc/named.conf` (gekürzt und bereits angepasst für Linuxbu.ch)

```
# Copyright (c) 2001-2004 SuSE Linux AG, Nuernberg, Germany.
# All rights reserved.
#
# Author: Frank Bodammer, Lars Mueller <lmuelle@suse.de>
#
# /etc/named.conf
#
# Copyright (c) 2001-2004 SuSE Linux AG, Nuernberg, Germany.
# All rights reserved.
#
# Author: Frank Bodammer, Lars Mueller <lmuelle@suse.de>
```

```
#
# /etc/named.conf
#
# This is a sample configuration file for the name server BIND 9.  It
works as
# a caching only name server without modification.
#
# A sample configuration for setting up your own domain can be found
in
# /usr/share/doc/packages/bind/sample-config.
#
# A description of all available options can be found in
# /usr/share/doc/packages/bind/misc/options.

options {

    # The directory statement defines the name server's working
directory

    directory "/var/lib/named";

    # Write dump and statistics file to the log subdirectory.  The
    # pathenames are relative to the chroot jail.

    dump-file "/var/log/named_dump.db";
    statistics-file "/var/log/named.stats";

    # The forwarders record contains a list of servers to which
queries
    # should be forwarded.  Enable this line and modify the IP
address to
    # your provider's name server.  Up to three servers may be
listed.

    #forwarders { 192.0.2.1; 192.0.2.2; };

…
    # The allow-query record contains a list of networks or IP
addresses
    # to accept and deny queries from. The default is to allow
queries
    # from all hosts.

    #allow-query { 127.0.0.1; };

    # If notify is set to yes (default), notify messages are sent to
other
```

```
    # name servers when the the zone data is changed.  Instead of
setting
    # a global 'notify' statement in the 'options' section, a
separate
    # 'notify' can be added to each zone definition.

    notify no;
    include "/etc/named.d/forwarders.conf";
};


zone "." in {
        type hint;
        file "root.hint";
};

zone "localhost" in {
        type master;
        file "localhost.zone";
};

zone "0.0.127.in-addr.arpa" in {
        type master;
        file "127.0.0.zone";
};

# Include the meta include file generated by createNamedConfInclude.
This
# includes all files as configured in NAMED_CONF_INCLUDE_FILES from
# /etc/sysconfig/named

include "/etc/named.conf.include";

# You can insert further zone records for your own domains below or
create
# single files in /etc/named.d/ and add the file names to
# NAMED_CONF_INCLUDE_FILES.
# See /usr/share/doc/packages/bind/README.SUSE for more details.

zone "lokales-netz.de" in {
        file "master/lokales-netz.de";
        type master;
};

zone "1.168.192.in-addr.arpa" in {
        type master;
        file "master/1.168.192.in-addr.arpa";
};
```

Zu den einzelnen Abschnitten dieser Datei:

```
# Copyright (c) 2001-2004 SuSE Linux AG, Nuernberg, Germany
#
# Author: Frank Bodammer, Lars Mueller <lmuelle@suse.de>
#
# /etc/named.conf
#
# This is a sample configuration file for the name server BIND 9.
```

Zeilen, die mit dem Lattenzaun # beginnen, sind Kommentare. Hier betonen die Autoren, dass es sich um eine Konfigurationsdatei für das aktuelle `Bind9` und nicht für ältere Versionen handelt.

```
options {

    # The directory statement defines the name server's
    # working directory

    directory "/var/lib/named";

    # The forwarders record contains a list of servers to
    # which queries should be forwarded. Enable this line and
    # modify the IP-address to your provider's name server.
    # Up to three servers may be listed.

    #forwarders { 192.0.2.1; 192.0.2.2; };
...
    # The allow-query record contains a list of networks or
    # IP-addresses to accept and deny queries from. The
    # default is to allow queries from all hosts.

    allow-query { 127.0/16; 192.168.1/24; };
```

Das Options-Statement gibt zuerst den Pfad zu den weiteren Konfigurationsdateien an. Dieser Pfad hat sich gegenüber den Vorgängerversionen leicht verändert. Unterhalb von `/var/lib/named` finden Sie bei aktuellen Installationen noch die Verzeichnisse `master`, `slave` und `dyn`. Im Verzeichnis `master` liegen die Zonendateien, für die der Server als Master zuständig ist. Im Verzeichnis `slave` befinden sich Kopien von Dateien, die von anderen Nameservern bezogen wurden und im Verzeichnis `dyn` Zonendateien, die der Dienst DHCPD verändern darf.

Anfragen, die der Name-Server nicht beantworten kann, gibt er an den oder die Name-Server weiter, die im `forwarders`-Statement aufgeführt sind. Als `forwarders` sollten Sie hier Name-Server Ihres Providers eintragen. Als Beispiel angegeben ist hier ein Name-Server der Telekom.

Später folgt dann eine Angabe, von wo aus auf den Name-Server zugegriffen werden darf. Hier ist ein Zugriff nur aus dem lokalen Netz heraus und vom Server selber zugelassen.

Die `forwarders` können Sie auch mit dem bereits erwähnten Konfigurations-Tool von YaST eingeben. Bei Einwahlverbindungen können Sie hier auch die Einstellung *PPP-Daemon legt Forwarders fest* aktivieren, dann werden die entsprechenden Adressen bei erfolgreicher Interneteinwahl automatisch eingetragen.

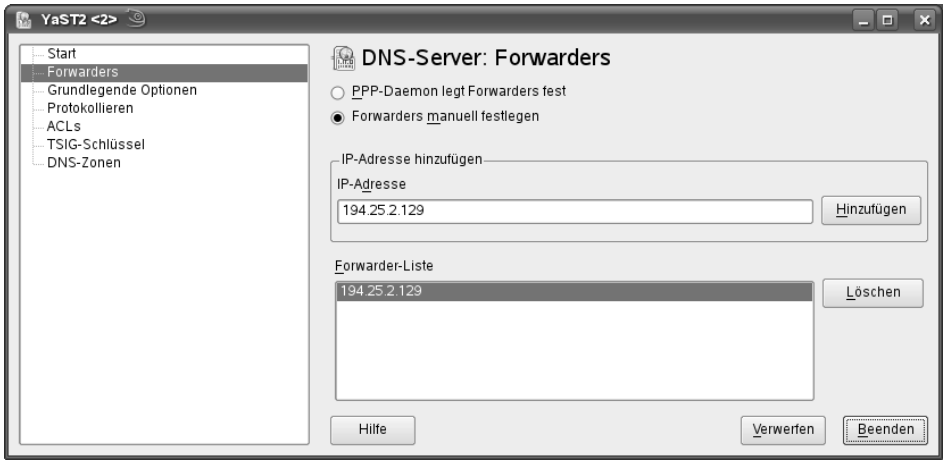

Abbildung 13.4: YaST: DNS-Forwarders

YaST übernimmt diese Einstellungen aber nicht direkt in die `named.conf`, sondern fügt dort nur die Zeile

```
include "/etc/named.d/forwarders.conf";
```

ein.

Diese Datei enthält die Zeilen zur Angabe der Name-Server für die Weiterleitung.

```
# Copyright (c) 2001-2004 SuSE Linux AG, Nuernberg, Germany.
# All rights reserved.
# Forwarders file for PPPD updates (only)
#
# /etc/named.d/forwarders.conf

forwarders { 194.25.2.129; };
```

Sehr wichtig sind die Zone-Statemens an Ende der Haupt-Konfigurationsdatei.

```
zone "lokales-netz.de" in {
   type master;
   file "master/lokales-netz.de";
};
```

Mit dem Zone-Statement bekommt der Name-Server die Zuständigkeit für `lokales-netz.de`. Er ist *primärer Name-Server* (*master*) für diese Domain. Neben einem primären Name-Server könnten Sie auch einen *Slave Name-Server* einrichten, der beim Ausfall des Masters dessen Aufgabe übernehmen kann. Die eigentlichen Adressen finden sich in der Datei `/var/lib/named/master/lokales-netz.de` (s. u.). Der gewählte Dateiname ist beliebig, stimmt hier aber mit dem Namen der Zone überein, was die Vorgabe des Konfigurationstools von YaST ist. Dadurch kann man die Grundlagen der Zonendatei mit YaST erstellen und dann im Texteditor erweitern.

```
zone "localhost" in {
   type master;
   file "localhost.zone";
};
```

Dieses Zone-Statement ist notwendig, damit der Server auch den Namen `localhost` zu `127.0.0.1` auflösen kann, der nichts mit `lokales-netz.de` zu tun hat. Die Zonendatei und die Zeilen in der Konfigurationsdatei hat OpenSUSE schon mit angelegt.

```
zone "1.168.192.in-addr.arpa" in {
   type master;
   file "master/1.168.192.in-addr.arpa";
};
```

Im vorliegenden Beispiel hat `boss.lokales-netz.de` die IP-Adresse `192.168.1.2`. Diese Zuordnung ergibt sich aus der Zonendatei `192.168.1.zone`. Für die Rückwärtsauflösung von `192.168.1.2` zu `boss.lokales-netz.de` ist diese Datei zuständig. Die Rückwärtsauflösung soll auch die IP-Angabe im Dateinamen andeuten.

```
zone "0.0.127.in-addr.arpa" in {
   type master;
   file "127.0.0.zone";
};
```

Für die Rückwärtsauflösung `127.0.0.1` zu `localhost` ist wieder eine eigene Zonendatei notwendig.

```
zone "." in {
   type hint;
   file "root.hint";
};
```

Diese fünfte Zonendatei enthält die IP-Adressen der *Root*-Name-Server. Die mitgelieferte Datei braucht normalerweise nicht geändert zu werden.

13.2.3 DNS-Zonen konfigurieren

Wichtigster Inhalt der Zonendateien (*Master Files*) sind die *Resource Records*, welche den Namen die IP-Adressen zuordnen bzw. umgekehrt den IP-Adressen die Namen. Die Dateien haben folgende Grundstruktur:

Sie beginnen mit Direktiven, die jeweils mit dem $-Zeichen anfangen:

- $ORIGIN legt fest, welche Domain an unvollständige Adressangaben angehängt werden soll. Fehlt diese Angabe, so benutzt bind den Zonennamen aus der /etc/named.conf. In den folgenden Beispielen findet sich diese Direktive daher nicht.
- $TTL (Time To Live) gibt eine Standardgültigkeitsdauer für die Resource Records vor, hier zwei Tage (2D).
- $GENERATE ist eine Bind8/Bind9- spezifische, nicht standardisierte Direktive, mit der man viele gleichartige Resource Records erzeugen kann. Eine genauere Beschreibung findet sich im Beispiel privat.zone.

Alle weiteren Zeilen sind dann Resource Records mit folgendem Aufbau:

```
<Name> IN <Typ> <Beschreibung>
```

Der erste Record ist am aufwändigsten, er ist vom Typ SOA (Start Of Authority) und beinhaltet Grundeinstellungen für die Zone. Dazu gehören die Angabe des Name-Servers und der E-Mail-Adresse der Kontaktperson. Bei dieser Mail-Adresse ersetzt man das @-Zeichen durch einen Punkt.

Danach kommen in Klammern eine Seriennummer und Zeitangaben für das Caching. Die Zeitangaben kann man einfach übernehmen, 3H steht für 3 Stunden, 15M für 15 Minuten, 1W für eine Woche und 1D für einen Tag.

Hat man auch Slave-Name-Server (sekundäre Name-Server) im Netz, so muss man die Seriennummer bei jeder Änderung erhöhen, damit die anderen Server Änderungen übernehmen. Baut das Nummernsystem auf dem Kalenderdatum auf, sollte man stets eine mehrstellige Nummer anfügen, z. B. 2008031203, für die dritte Version vom 12. März 2008.

Nun folgen einige Adressangaben. Vollständige DNS-Namen bekommen noch einen Punkt angehängt, an alle Namen ohne Punkt am Ende wird der betreffende Domainname angekoppelt.

Für die Datei lokales-netz.de ist es also gleichbedeutend, ob man

- boss.lokales-netz.de. (beachten Sie den Punkt am Ende) oder
- boss (kein Punkt am Ende) schreibt.

Die meisten Records sind vom Typ A und dienen der Adresszuordnung. Vor dem IN steht der Name des Rechners und nach dem A seine IP-Adresse.

Ein Record vom Typ CNAME vergibt einen weiteren Namen (Alias) für einen Rechner. Meist werden so www, ftp, mail und news definiert. Links von IN steht wieder der zu definierende Name und rechts vom CNAME der offizielle Name.

Ein Record vom Typ NS definiert Name-Server. Ein Netz mit ständiger Internetverbindung muss zwei Name-Server besitzen, damit beim Ausfall eines Name-Servers der andere einspringen kann.

Für den Austausch von Mail sind die `MX`-Records (Mail-Exchange) wichtig. Diese geben nach dem Schlüsselwort `MX` noch eine Priorität für den Rechner an, um eine Rangfolge festzulegen, wenn mehrere Mailserver eingetragen sind. Je kleiner die Zahl, desto höher ist die Priorität. Null entspricht also der höchsten Stufe. Man kann z. B. 10 weitere Rechner mit niedrigerer Priorität angeben, die notfalls eingehende Mails annehmen, falls der primäre Rechner ausfällt.

`/var/lib/named/master/lokales-netz.de`

```
$TTL 2d
$GENERATE 20-127  client-$ A 192.168.1.$
@               IN SOA          boss.lokales-netz.de.
root.boss.lokales-netz.de. (
                                2008072500      ; serial
                                3h              ; refresh
                                1h              ; retry
                                1w              ; expiry
                                1d )            ; minimum

                IN NS           boss
                IN MX 0         boss

boss            IN A            192.168.1.2
www             IN CNAME        boss
www2            IN CNAME        boss
mail            IN CNAME        boss
ns              IN CNAME        boss
ftp             IN CNAME        boss
news            IN CNAME        boss

rosine          IN A            192.168.1.10
nuss            IN A            192.168.1.11
flocke          IN A            192.168.1.12
schoko          IN A            192.168.1.13
```

`Boss` ist Name-Server und Mail-Server mit höchster Priorität für die Domain `lokales-netz.de`. Weiter bestimmt die Datei die IP-Adressen für `boss`, `rosine`, `nuss`, `flocke` und `schoko`.

Mit einem Record vom Typ `A` kann man die IP-Adressen für beliebig viele Rechner angeben.

Manche Betreiber geben sich bei den Rechnernamen sehr viel Mühe und überlegen sich ein System. Namen von Bäumen (Bonsai, Erle, ...), Planeten (Mars, Venus, ...) oder Müsli-Bestandteilen (Flocke, Rosine, Nuss, ...).

Praktischer baut man aber die Namen systematisch auf. Dann kann man die Datei von einem Konfigurations-Programm erzeugen lassen und gleich für alle 255 möglichen IP-Adressen verschiedene Namen generieren lassen, z. B. nach dem System

```
client-20  IN A       192.168.1.20
client-21  IN A       192.168.1.21
client-22  IN A       192.168.1.22
...
client-127 IN A       192.168.1.127
```

Geht man so vor, braucht man bei späteren Erweiterungen des Netzes keine Einträge im Name-Server zu ändern. Genau diese Zeilen erzeugt die `$GENERATE` Direktive.

```
$GENERATE 20-127  client-$ A 192.168.1.$
```

Für die Werte von 20 bis 127 (die Werte sind willkürlich gewählt) erzeugt der Eintrag Resource Records nach dem Muster

```
client-$   IN A       192.168.1.$
```

wobei `generate` das `$`-Zeichen jeweils durch den aktuellen Wert ersetzt.

Als Alias für `Boss` sind `www`, `mail`, `ns`, `ftp` und `news` eingetragen. In einem lokalen Netz ist das praktisch. Für Rechner, die ständig mit dem Internet verbunden sind, gilt aber:

Warnung: Wenn Rechnernamen über Rechnerfunktionen informieren, freuen sich Eindringlinge. Eine einfache Verteidigungsstrategie lautet daher, keine auf die Funktion hinweisenden Namen zu vergeben.

Viele Programme adressieren den Rechner, auf dem sie laufen, über `localhost` und nicht über `boss.lokales-netz.de`. Daher gibt es für `localhost` auch `127.0.0.1` als allgemeingültige IP-Adresse.

`localhost` ordnet man `127.0.0.1` in einer eigenen Zonendatei zu.

Diese Datei hat den gleichen Aufbau wie die `privat.zone`, definiert aber nur den einzigen Namen `localhost` mit der zugehörigen IP `127.0.0.1`. Dargestellt ist hier die mitgelieferte Datei, die etwas unübersichtlich wirkt, weil OpenSUSE hier mit Platzhaltern arbeitet, um die Datei allgemeingültig zu halten.

`/var/lib/named/localhost.zone`

```
$TTL 1W
@              IN SOA  @   root (
                           42              ; serial (d. adams)
                           2D              ; refresh
                           4H              ; retry
                           6W              ; expiry
                           1W )            ; minimum

                 IN NS           @
                 IN A            127.0.0.1
```

Der Platzhalter @ steht hier für den Rechner selber, also boss.lokales-netz.de. Die Seriennummer 42 soll an das Kultbuch »Per Anhalter durch die Galaxis« von Douglas Adams erinnern. Eine derartige Seriennummer ist aber nur für Zonendateien sinnvoll, bei denen Sie keinerlei Änderungen erwarten.

13.2.4 Von der IP-Nummer zum Hostnamen: Reverse Name Server Lookup

Die bisher beschriebenen Dateien privat.zone und localhost.zone sollen dem Rechnernamen je eine IP-Adresse zuordnen. Reverse Lookup hingegen ermittelt umgekehrt zu einer IP-Adresse den Rechnernamen.

Bei dieser Namensauflösung über Zonendateien wendet man den Record-Typ PTR (Pointer) an.

Für das Reverse Lookup dient eine spezielle Domain, in-addr.arpa, vor die man die IP-Adressen in verdrehter Reihenfolge setzt. Für die Suche nach dem Namen zu 192.168.1.2 geht man mit 2.1.168.192.in-addr.arpa an eine geeignete Zonendatei und sucht dort den zugehörigen Namen.

/var/lib/named/master/1.168.192.in-addr.arpa

```
$TTL 2d
$GENERATE  20-127  $  PTR client-$.lokales-netz.de.
@   IN SOA   boss.lokales-netz.de. root.boss.lokales-netz.de. (
                                2008072501      ; serial
                                3h              ; refresh
                                1h              ; retry
                                1w              ; expiry
                                1d )            ; minimum

               IN NS            boss.lokales-netz.de.
2              IN PTR           boss.lokales-netz.de.
10             IN PTR           rosine.lokales-netz.de.
11             IN PTR           nuss.lokales-netz.de.
12             IN PTR           flocke.lokales-netz.de.
13             IN PTR           schoko.lokales-netz.de.
```

Als Name ist hier nur jeweils die letzte Zahl der IP-Adresse angegeben, da Bind 1.168.192.in-addr.arpa ergänzt.

Auch in dieser Datei erzeugt die $GENERATE Direktive einen großen Teil der Ressource Records.

Die Zuordnung 127.0.0.1 zu localhost nutzt eine eigene Pseudoadresse 1.0.0.127.in-addr.arpa und damit auch eine eigene Zonendatei.

/var/lib/named/127.0.0.zone

```
$TTL 1W
@               IN SOA          localhost.   root.localhost. (
                        42              ; serial (d. adams)
                        2D              ; refresh
                        4H              ; retry
                        6W              ; expiry
                        1W )            ; minimum

                IN NS           localhost.
1               IN PTR          localhost.
```

Auch die Zonendateien können Sie mit dem Konfigurations-Tool von YaST erstellen und bearbeiten.

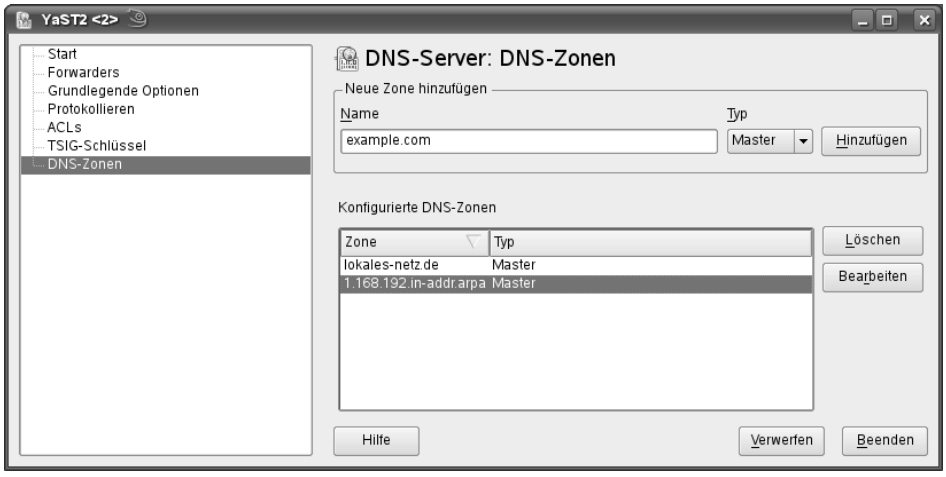

Abbildung 13.5: YaST: DNS-Zonen

Sie sehen hier alle im unteren Teil der Konfigurationsdatei eingetragenen Zonendateien. Die Zonen für `localhost` und die zugehörige rückwärtige Auflösung bietet YaST Ihnen hier nicht an, weil sie in der Konfigurationsdatei weiter oben stehen.

Sie können eine Zone bearbeiten, indem Sie sie auswählen und auf *Zone bearbeiten* klicken.

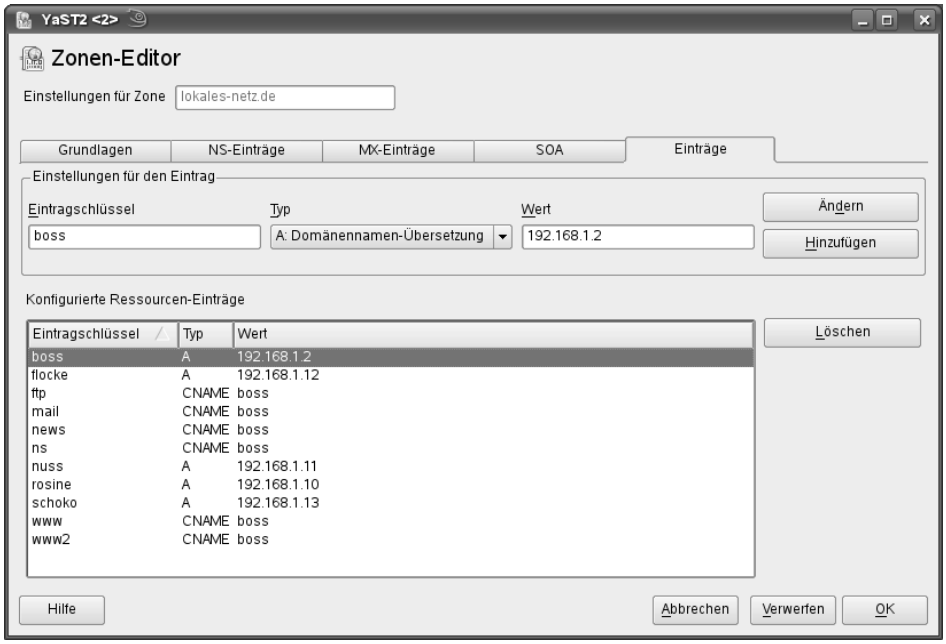

Abbildung 13.6: YaST: DNS-Zone editieren

Der Editor ist im Prinzip auch ganz nützlich, nur kennt er die `$GENERATE`-Direktive nicht, er löscht sie sogar. Wenn Sie diese hilfreiche Direktive benutzen wollen, müssen Sie auf das Konfigurations-Tool von YaST leider verzichten.

13.3 Erster Start des Name-Servers

Nach dem Start des Name-Servers mit

```
rcnamed start
```

finden Sie in der Datei `/var/log/messages` Meldungen wie:

```
Jul 25 13:17:12 boss named[15796]: starting BIND 9.4.2-P1 -t
/var/lib/named -u named
Jul 25 13:17:12 boss named[15796]: found 1 CPU, using 1 worker thread
Jul 25 13:17:12 boss named[15796]: loading configuration from
'/etc/named.conf'
Jul 25 13:17:12 boss named[15796]: listening on IPv6 interfaces, port
53
Jul 25 13:17:12 boss named[15796]: listening on IPv4 interface lo,
127.0.0.1#53
Jul 25 13:17:13 boss named[15796]: listening on IPv4 interface lo,
127.0.0.2#53
```

```
Jul 25 13:17:13 boss named[15796]: listening on IPv4 interface eth0,
192.168.1.2#53
…
Jul 25 13:17:13 boss named[15796]: command channel listening on
127.0.0.1#953
Jul 25 13:17:13 boss named[15796]: command channel listening on
::1#953
Jul 25 13:17:13 boss named[15796]: zone 0.0.127.in-addr.arpa/IN:
loaded serial 42
Jul 25 13:17:13 boss named[15796]: zone 1.168.192.in-addr.arpa/IN:
loaded serial 2008072501
Jul 25 13:17:13 boss named[15796]: zone lokales-netz.de/IN: loaded
serial 2008072500
Jul 25 13:17:13 boss named[15796]: zone localhost/IN: loaded serial
42
Jul 25 13:17:13 boss named[15796]: running
```

- Die erste Zeile ist eine allgemeine Startmeldung des Name-Servers, aus der sich vor allem die Versionsnummer, hier `9.4.2`, ergibt.
- Danach listet die Datei die IP-Adressen, auf die der Name-Server anspricht, `192.168.1.2` und `127.0.0.1` sowie jeweils `Port 53`.
- Am Ende zeigen vier Zeilen das erfolgreiche Laden der Zonendateien an.
- Die besonders wichtige letzte Zeile informiert, dass der Name-Server jetzt Anfragen beantworten kann.

13.3.1 Test und Diagnose

Wenn der Name-Server erfolgreich gestartet ist (*running*), kann man mit `host` Anfragen auf dem Linux-Server testen, ob er

- lokale Anfragen und
- weltweite Anfragen

richtig beantwortet.

Zum Testen prüft man systematisch Beispiele, die alle Zonendateien benötigen.

Der Test beginnt mit der Zone `lokales-netz.de`: die Anfrage

```
host  www
```

sollte folgende Antwort ergeben:

```
www.lokales-netz.de is an alias for boss.lokales-netz.de.
boss.lokales-netz.de has address 192.168.1.2
```

Der Nameserver antwortet mit dem Namen des Rechners, seiner IP, sowie dem vollständigen Alias.

Als Zweites ist `localhost.zone` dran:

```
host  localhost
```

muss ergeben:

```
localhost has address 127.0.0.1
```

Dann folgt die Auflösung gemäß Zone `1.168.192.in-addr.arpa`:

```
host  192.168.1.12
```

löst der Name-Server auf zu:

```
12.1.168.192.in-addr.arpa domain name pointer flocke.lokales-netz.de.
```

Abschließend folgt `127.0.0.zone`:

```
host  127.0.0.1
```

löst er auf zu

```
1.0.0.127.in-addr.arpa domain name pointer localhost.
```

Wenn die bisherigen Tests erfolgreich verlaufen sind und eine Verbindung ins Internet besteht, sollte man auch externe Adressen abfragen können:

```
host  ns.suse.de
```

Hier sucht `host` den Name-Server von SUSE. Als Antwort erhält man

```
ns.suse.de has address 195.135.220.2
```

Diese Antwort hat der eigene Name-Server natürlich nicht selber geben können, er hat sich aber eine Auskunft bei den unter `forwarders` eingetragenen Name-Servern besorgt.

Mit

```
host  www.suse.de ns.suse.de
```

kann man direkt einen bestimmten Name-Server, hier den immer noch existierenden SUSE-Name-Server, abfragen:

```
Using domain server:
Name: ns.suse.de
Address: 195.135.220.2#53
Aliases:

www.suse.de is an alias for turing.suse.de.
turing.suse.de has address 195.135.220.3
```

Die Antwort ist etwas umfangreicher, da auch Informationen über den befragten Name-Server auftauchen.

Wenn alle Tests erfolgreich verlaufen sind, braucht man nur noch zu veranlassen, dass der Name-Server zukünftig beim Hochfahren des Systems automatisch startet. Dazu

geht man in YaST-Kontrollzentrum unter *System* • *Runlevel-Editor* auf *Runlevel-Eigenschaften* und sucht in der Liste die Zeile für den *named*.

Bringen Sie den Rollbalken auf diese Zeile und klicken Sie dann nacheinander auf die mit *3* bzw. *5* beschrifteten Checkboxen unterhalb der Auswahlliste. Der Nameserver startet dann zukünftig in diesen Runleveln automatisch.

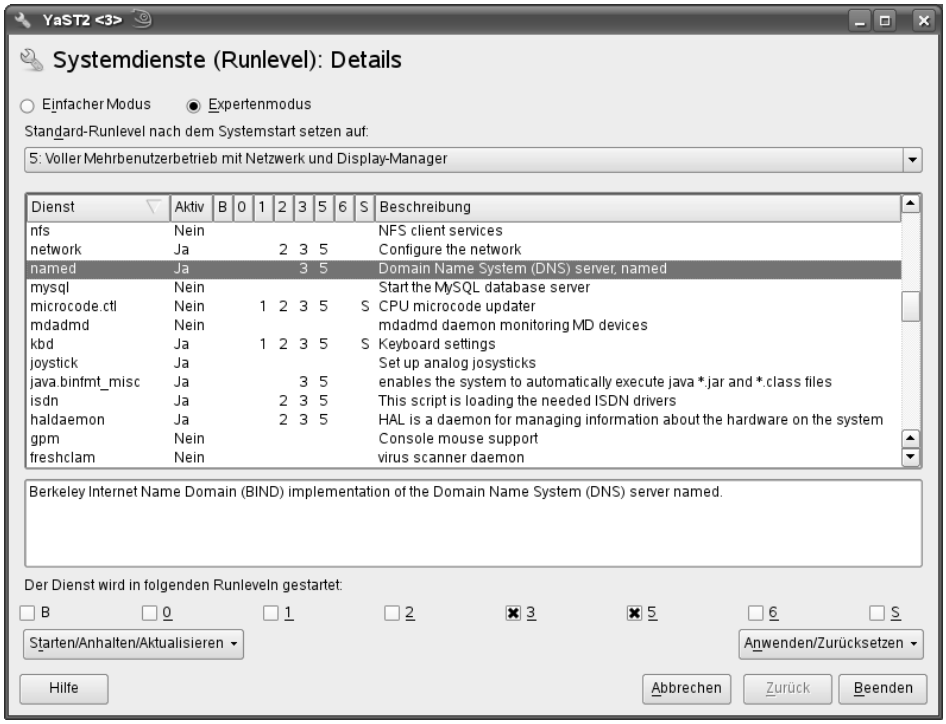

Abbildung 13.7: Runlevel-Editor: named

13.3.2 Troubleshooting

Die Konfiguration des Name-Servers ist eine der wenigen Konfigurationen, bei denen OpenSUSE bzw. YaST wenig helfen bzw. vorkonfigurieren können.

Sollte der Name-Server nicht richtig starten, so gibt er seine Fehlermeldungen in der Datei `/var/log/messages` aus.

Syntaxfehler in der Datei `/etc/named.conf` gibt `Bind` mit der zugehörigen Zeilennummer an. Diese Fehler führen meist dazu, dass der Name-Server überhaupt nicht startet.

Der Name-Server vermerkt außerdem Fehler in einer der Zonendateien. Diese führen zu einer Teilfunktion des Name-Servers, er arbeitet dann nur mit den Informationen aus den fehlerfreien Dateien.

Der Name- Server muss alle Anfragen der Art

```
host  boss
host 192.168.1.2
host  localhost
host 127.0.0.1
```

erfolgreich auflösen können. Sollten einzelne dieser Anfragen fehlschlagen, ist die zugehörige Zonendatei fehlerhaft.

Bei fehlerhaften Zonendateien spielt oft der abschließende Punkt eine Rolle. Immer dann, wenn nichts mehr ergänzt werden darf, weil eine Adresse vollständig ist, muss am Ende ein Punkt stehen. Bei unvollständigen Angaben, die noch ergänzt werden sollen, darf am Ende jedoch kein Punkt stehen.

13.4 Dynamische Updates

Wenn Sie in Ihrem Netz mit Windows-Clients arbeiten, haben Sie das Problem zweier unterschiedlicher Namensauflösungen. Sie haben einerseits die Wins-Namen und andererseits einen Namen innerhalb der lokalen Domain. Bisher war es kaum möglich, beide Namensräume zu vereinheitlichen.

Im Zusammenspiel mit dem DHCP-Server (siehe Kapitel 2.6) können Sie eine interessante Funktionalität erreichen. Wenn sich ein Windows-Client im Netz anmeldet, versucht er per DHCP eine IP-Adresse zu bekommen. Dazu übermittelt er dem DHCP-Server seine MAC-Adresse und seinen Wins-Namen.

```
Jan  4 17:42:55 boss dhcpd: DHCPDISCOVER from 00:50:bf:58:56:fd
(OEMComputer) via eth0
```

Mit diesem Namen kann der DHCPD den Nameserver aktualisieren, wenn Sie die Konfigurationen entsprechend anpassen.

In der Datei `/etc/named.conf` müssen Sie die Zonen-Statements etwas erweitern, um das Update zu erlauben. Außerdem müssen die Zonen-Dateien im Verzeichnis `dyn` abgelegt sein und nicht mehr im Verzeichnis `master`.

```
# You can insert further zone records for your own domains below.


zone "lokales-netz.de" in {
   type master;
   file "dyn/lokales-netz.de";
   allow-update  {127.0/16; 192.168/16; };
```

```
};

zone "1.168.192.in-addr.arpa" in {
   type master;
   file "dyn/1.168.192.in-addr.arpa ";
   allow-update  {127.0/16; 192.168/16; };
};
```

Mit der Zeile

```
allow-update  {127.0/16; 192.168/16; };
```

erlauben Sie dem Server selbst und den Rechnern in Ihrem lokalen Netz, die Zonendateien zu aktualisieren.

Nun müssen Sie noch die `dhcpd.conf` Ihres Linux-Servers so ändern, dass der DHCPD die Zonendateien auch tatsächlich ändert.

```
# dhcpd.conf
#
# a minimal /etc/dhcpd.conf example
# modified for www.linuxbu.ch

# this statement is needed by dhcpd-3 needs at least this statement.
# you have to delete it for dhcpd-2, because it does not know it.
ddns-update-style ad-hoc; ddns-updates on;
```

In der Beispieldatei aus Kapitel 2 stand an dieser Stelle

```
ddns-update-style  none; ddns-updates off;
```

was das Aktualisieren unterbunden hatte. Die Aktualisierung ist ja auch erst sinnvoll, wenn Sie einen eigenen Nameserver eingerichtet haben und betreiben.

Der DHCPD ändert Zonendateien des Nameservers nicht nur virtuell, sondern dauerhaft. Dabei setzt er z. B. auch die Generate-Zeile in die entsprechenden einzelnen Zeilen um, was die Dateien erheblich vergrößert.

Sie sollten daher zuvor Kopien aller Zonendateien anfertigen, um sie leichter ändern zu können.