15 Domain Name-Server einrichten

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

Mehr zum Aufbau von Domain-Namen finden Sie in Internetbüchern 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 NT/XP unter `winnt\system32\drivers\etc`, bei Linux im Verzeichnis `/etc`. Zunächst 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.

15.1 Wann Sie einen eigenen Nameserver brauchen

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

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

15.2 So funktionieren das Domain Name System und Internet-Domains

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

Zur Zeit gibt es 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.fi`) – verwalten die Landesdomains wie `de` (Deutschland), `at` (Österreich), `ch` (Schweiz) und `fr` (Frankreich).

Inzwischen verwalten zahlreiche konkurrierende Firmen die Drei-Buchstaben-Domains aus der Anfangszeit des Internets (`com`, `edu`, `gov`, `mil`, `net`, `org`, `int`). Hier kommt es immer häufiger zu Pannen wie Doppelvergabe.

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 geworden ist, ist die eindeutige Zuständigkeit geklärt.

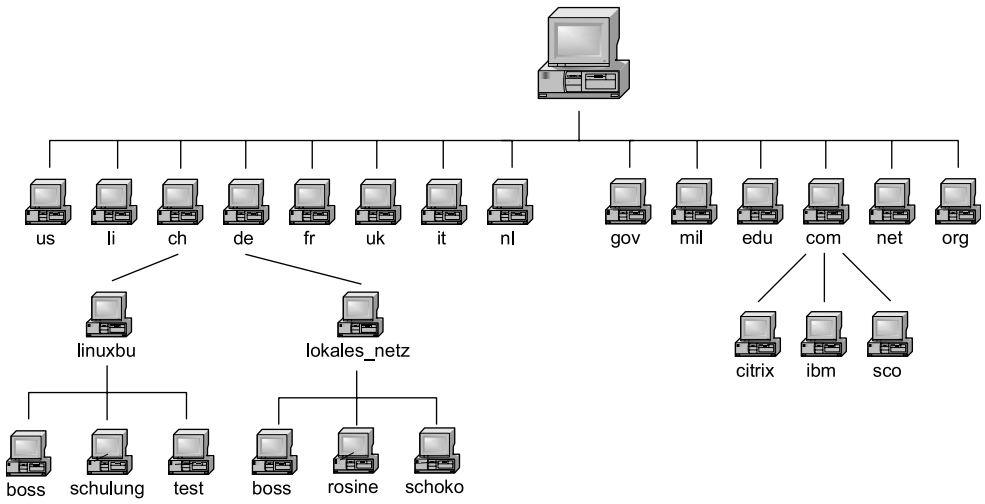

Abbildung 15.1: Baumstruktur

Der Ablauf einer Namens-Anfrage ist folgendermaßen:

1. Ruft jemand in den USA die Web-Adresse `www.linuxbu.ch` auf, so landet dessen Nameserver-Anfrage über Zwischenschritte beim zentralen Nameserver des NIC, dem Root-Server.
2. Dieser gibt die Anfrage an den Nameserver des Ch-NIC, der Sie dann an den für *linuxbu.ch* zuständigen Nameserver (`nameserv.deltaweb.de`) weitergibt, von wo er nun endgültig die IP-Adresse (*62.134.48.2*) bekommt.
3. Diese IP-Adresse geht dann auf dem langen Weg über die beteiligten Nameserver an den anfragenden Rechner weiter.

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

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

Auch die Zuordnung eines zuständigen Mailservers für einen Adressbereich erfolgt über den Nameserver, dafür sind die *mx-records* (Mail Exchanger-Einträge) zuständig.

15.2.1 Die Hosts-Datei

In kleineren Netzen ist ein eigener Nameserver 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 die 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-Adressformats auf 6Byte bedeutsam.

15.2.2 Nameserver installieren und konfigurieren

Der Nameserver befindet sich bei SuSE im Paket `bind9` der Selektion `Netzwerk/Server` bzw. der zugehörigen rpm-Datei auf *CD2*. Die Standardinstallation richtet das Paket nicht ein, man muss dies also gegebenenfalls nachholen, bevor man den DNS konfiguriert.

Folgende Dateien sind für die beschriebene Konfiguration wichtig:

Datei	*Bedeutung*
`/usr/sbin/named`	Die Binärdatei, die den Nameserver 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-Nameservern (Standard-Nameserver).

Tabelle 15.1: Konfigurationsdateien des Nameservers

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

Tabelle 15.1: Konfigurationsdateien des Nameservers (Forts.)

Hinweis: Sie können den Nameserver erst starten, wenn sie alle Konfigurationsdateien angelegt haben.

Damit der Rechner selber später auch auf den Nameserver zugreifen kann, sollte man zuerst das YaST-Kontollzentrum starten und dort unter *Netzwerkdienste* • *DNS- und Hostname* die notwendigen Angaben machen. Hier gibt man die IP-Adresse (`192.168.1.2`) bzw. die IP-Adressen für den oder die Nameserver, sowie den Domain-Namen (`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.

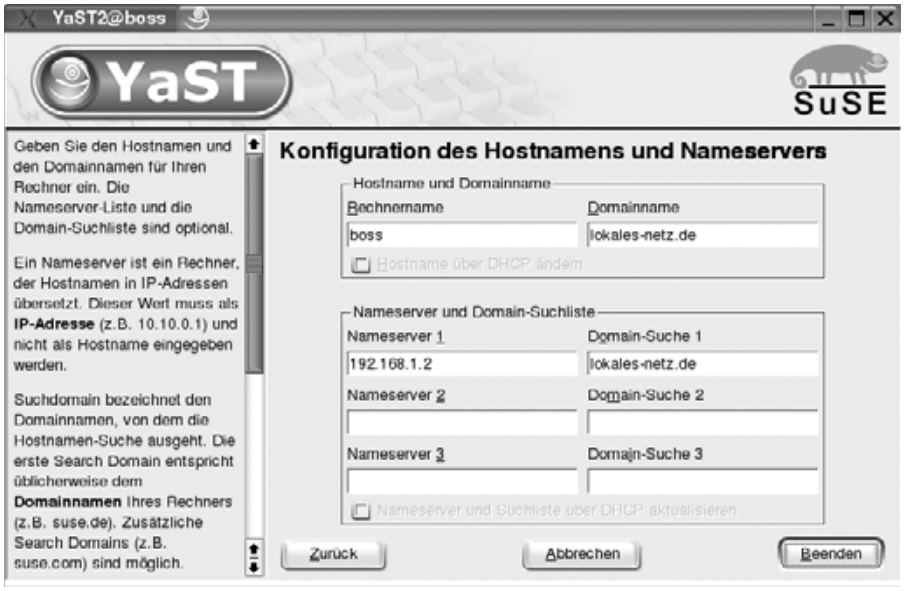

Abbildung 15.2: Konfiguration des Nameservers

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 Nameserver `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 Nameserver `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.

Die von SuSE installierten Musterdateien können Sie für Ihre Bedürfnisse anpassen. Eine umfangreiche Dokumentation zum Nameserver *Bind* findet sich im Ordner `/usr/share/doc /packages/bind9`.

`/etc/named.conf`

```
# Copyright (c) 2001-2003 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. 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/bind9/sample-config.
#
# A description of all available options can be found in
# /usr/share/doc/packages/bind9/misc/options.

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 { 194.25.2.129; };

        # Enable the next entry to prefer usage of the name
        # server declared in the forwarders section.

        #forward first;

        # The listen-on record contains a list of local network
        # interfaces to listen on. Optionally the port can be
        # specified. Default is to listen on all interfaces
        # found on your system. The default port is 53.

        #listen-on port 53 { 127.0.0.1; };

        # The listen-on-v6 record enables or disables listening
        # on IPV6 interfaces. Allowed values are 'any' and
        # 'none' or a list of addresses. IPv6 can only be
        # used with kernel 2.4 in this release.

        listen-on-v6 { any; };

        # The next three statements may be needed if a firewall
        # stands between the local server and the internet.

        #query-source address * port 53;
        #transfer-source * port 53;
        #notify-source * port 53;
```

```
        # 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; };

        # 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;
};

# The following zone definitions don't need any modification.
# The first one is the definition of the root name servers.
# The second one defines localhost while the third defines
# the reverse lookup for localhost.

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";
};

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

zone "lokales-netz.de" in {
        type master;
        file "privat.zone";
};
zone "1.168.192.in-addr.arpa" in {
        type master;
        file "tavirp.zone";
};
```

Zu den einzelnen Abschnitten dieser Datei:

```
# Copyright (c) 2001-2003 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 { 194.25.2.129; };
...
        # 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.

Anfragen, die der Nameserver nicht beantworten kann, gibt er an den oder die Nameserver weiter, die im `forwarders`-Statement aufgeführt sind. Als `forwarders` sollten Sie hier den oder die Nameserver Ihres Providers eintragen.

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

Sehr wichtig sind die Zone-Statemens an Ende der Datei.

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

Mit dem Zone-Statement bekommt der Nameserver die Zuständigkeit für `lokales-netz.de`. Er ist *primärer Nameserver* (master) für diese Domain. Neben einem primären Nameserver könnten Sie auch einen *Slave Nameserver* einrichten, der beim Ausfall des Masters dessen Aufgabe übernehmen kann. Die eigentlichen Adressen finden sich in der Datei `/var/lib/named/privat.zone` (s. u.).

```
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.

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

Im vorliegenden Beispiel hat `boss.lokales-netz.de` die IP-Adresse `192.168.1.2`, diese Zuordnung ergibt sich aus der Zonen-Datei `privat.zone`. Für die Rückwärtsauflösung von `192.168.1.2` zu `boss.lokales-netz.de` ist diese Datei zuständig. Die Rückwärtsauflösung soll auch das `tavirp` (*privat* rückwärts gelesen) 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 Zonen-Datei notwendig.

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

Diese fünfte Zonen-Datei enthält die IP-Adressen der Root-Nameserver. Die mitgelieferte Datei braucht man normalerweise nicht zu ändern.

15.2.3 DNS-Zonen konfigurieren

Wichtigster Inhalt der Zonen-Dateien (Master Files) sind die Ressource 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 Zonen-Namen aus der `/etc/named.conf`. In den folgenden Beispielen findet sich diese Direktive daher nicht.

`$TTL` (Time To Live) gibt eine Standard-Gültigkeitsdauer für die Ressource Records vor, hier zwei Tage (`2D`).

`$GENERATE` ist eine Bind8/Bind9-spezifische, nicht standardisierte Direktive, mit der man viele gleichartige Ressource Records erzeugen kann. Eine genauere Beschreibung findet sich im Beispiel `privat.zone`.

Alle weiteren Zeilen sind dann Ressource 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 Nameservers 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-Nameserver (sekundäre Nameserver) 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. `2000031203`, für die dritte Version vom 12. März 2000.

Nun folgen einige Adressangaben. Vollständige DNS-Namen bekommen noch einen Punkt dahinter, alle Namen ohne Punkt am Ende bekommen den betreffenden Domain-Namen angehängt.

Für die Datei `privat.zone` 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 Nameserver. Ein Netz mit ständiger Internetverbindung muss zwei Nameserver besitzen, damit beim Ausfall eines Nameservers der andere einspringen kann.

Für den Austausch von Mails 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 Priorität. Man kann z. B. 10 weitere Rechner mit niedrigerer Priorität angeben, die notfalls eingehende Mails annehmen, falls der primäre Rechner ausfällt.

/var/lib/named/privat.zone

```
$TTL      2D
$GENERATE 20-127  client-$ A 192.168.1.$
@  IN SOA  boss.lokales-netz.de. postmaster.lokales-netz.de. (
           2003071203     ; serial (12.07.2003 Version 03)
           3H             ; refresh
           15M            ; 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 Nameserver und Mailserver 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,...).

Das ist zwar nett, praktischer ist es aber, die Namen einfach systematisch aufzubauen, 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 Nameserver 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 Ressourcen Records erzeugt 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 ist, nicht auf die Funktion hinweisende Namen zu vergeben.

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

`localhost` ordnet man `127.0.0.1` in einer eigenen Zonen-Datei 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 von SuSE mitgelieferte Datei, die dadurch etwas unübersichtlich wirkt, da SuSE 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 D. Adams erinnern. Eine derartige Seriennummer ist aber nur für Zonen-Dateien sinnvoll, bei denen Sie keinerlei Änderungen erwarten.

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

Die bisher beschriebenen Dateien `privat.zone` und `localhost.zone` sollen Rechnernamen je eine IP-Adresse zuordnen. Manchmal will man umgekehrt zu einer IP-Adresse den Rechnernamen ermitteln. Dies bezeichnet man als Reverse Lookup.

Bei dieser Namensauflösung über Zonen-Dateien wendet man den neuen 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 davor 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 Zonen-Datei und sucht dort den zugehörigen Namen.

/var/lib/named/tavirp.zone

```
$TTL  2D
$GENERATE 20-127 $ PTR client-$.lokales-netz.de.
@  IN SOA  boss.lokales-netz.de. postmaster.lokales-netz.de. (
             2003071203  ; serial (12.07.2003 Version 03)
             3H          ; refresh
             15M         ; 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 Pseudo-Adresse `1.0.0.127.in-addr.arpa` und damit auch eine eigene Zonen-Datei.

`/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.
```

15.3 Erster Start des Nameservers

Nach dem Start des Nameservers mit

```
rcnamed start
```

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

```
Jul 23 09:32:46 boss named[665]: starting BIND 9.2.2
↵ -t /var/lib/named -u named
Jul 23 09:32:47 boss named[665]: using 1 CPU
Jul 23 09:32:48 boss named[667]: loading configuration from
↵ '/etc/named.conf'
Jul 23 09:32:49 boss kernel: IPv6 v0.8 for NET4.0
Jul 23 09:32:49 boss kernel: IPv6 over IPv4 tunneling driver
Jul 23 09:32:49 boss named[667]: listening on IPv6 interfaces,
↵ port 53
Jul 23 09:32:49 boss named[667]: listening on IPv4 interface
↵ lo, 127.0.0.1#53
Jul 23 09:32:49 boss named[667]: binding TCP socket: address
↵ in use
```

```
Jul 23 09:32:49 boss named[667]: listening on IPv4 interface
↵ eth0, 192.168.1.2#53
Jul 23 09:32:49 boss named[667]: binding TCP socket: address
↵ in use
Jul 23 09:32:50 boss named[667]: command channel listening on
↵ 127.0.0.1#953
Jul 23 09:32:50 boss named[667]: command channel listening on
↵ ::1#953
Jul 23 09:32:50 boss named[667]: zone 0.0.127.in-addr.arpa/IN:
↵ loaded serial 42
Jul 23 09:32:50 boss named[667]: zone 1.168.192.in-addr.arpa/IN:
↵ loaded serial 2003071203
Jul 23 09:32:50 boss named[667]: zone lokales-netz.de/IN:
↵ loaded serial 2003071203
Jul 23 09:32:50 boss named[667]: zone localhost/IN:
↵ loaded serial 42
Jul 23 09:32:50 boss named[667]: running
```

- Die erste Zeile ist eine allgemeine Start-Meldung des Nameservers, aus der sich vor allem die Versionsnummer, hier `9.2.2`, ergibt.
- Danach listet die Datei die IP-Adressen, auf die der Nameserver anspricht, `192.168.1.2` und `127.0.0.1` sowie jeweils `Port 53`.
- Die folgenden vier Zeilen zeigen das erfolgreiche Laden der Zonen-Dateien an.
- Die besonders wichtige letzte Zeile informiert, dass der Nameserver jetzt Anfragen beantworten kann.

15.3.1 Test und Diagnose

Wenn der Nameserver 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 Zonen-Dateien benötigen.

Der Test beginng mit `privat.zone`:

```
host  www
```

sollte folgende Antworte 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 `tavirp.zone`:

```
host  192.168.1.12
```

löst der Nameserver 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 Nameserver von SuSE. Als Antwort erhält man

```
ns.suse.de has address 213.95.15.193
```

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

Mit

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

kann man direkt einen bestimmten Nameserver, hier den SuSE-Name-Server, abfragen:

```
Using domain server:
Name: ns.suse.de
Address: 213.95.15.193#53
```

```
Aliases:

www.suse.de is an alias for Turing.suse.de.
Turing.suse.de has address 213.95.15.200
```

Die Antwort ist etwas umfangreicher, da auch Informationen über den befragten Nameserver auftauchen.

Wenn alle Tests erfolgreich verlaufen sind, braucht man nur noch zu veranlassen, dass der Nameserver 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.

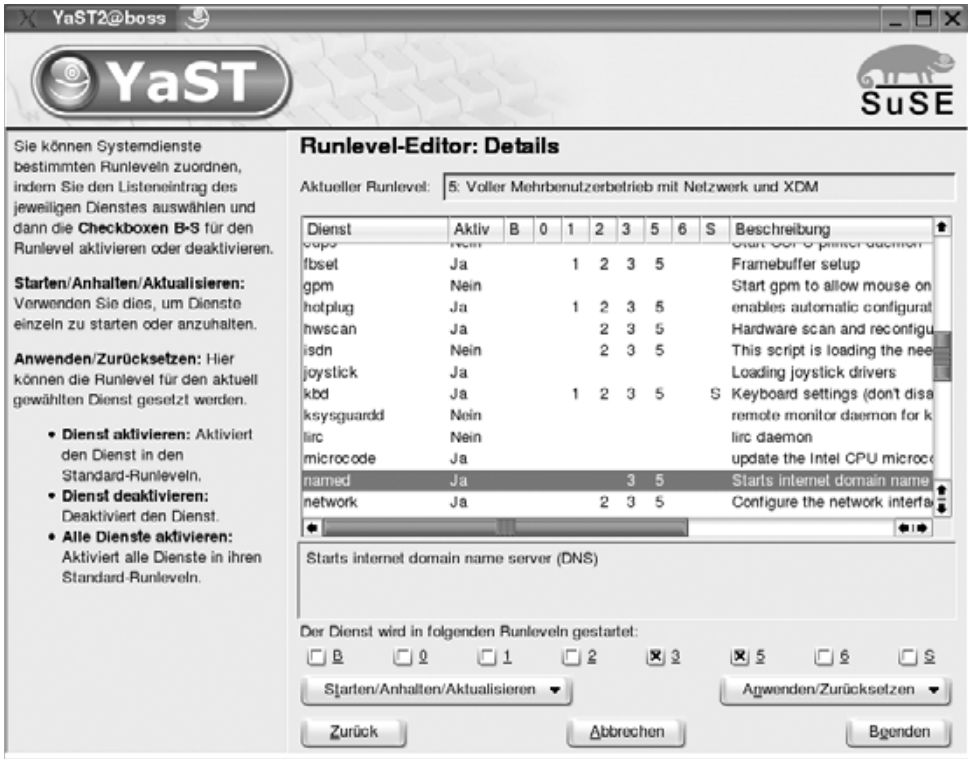

Abbildung 15.3: Runlevel-Editor: named

15.3.2 Troubleshooting

Die Konfiguration des Nameservers ist eine der wenigen Konfigurationen, bei denen SuSE bzw. YaST wenig helfen bzw. vorkonfigurieren können.

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

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

Der Nameserver vermerkt außerdem Fehler in einer der Zonen-Dateien. Diese führen zu einer Teilfunktion des Nameservers, er arbeitet dann nur mit den Informationen aus den fehlerfreien Dateien.

Der Nameserver 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 Zonen-Datei fehlerhaft.

Bei fehlerhaften Zonen-Dateien 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 kein Punkt stehen.

15.4 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.

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

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

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

Mit der Zeile

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

erlauben Sie dem Server selber und den Rechnern in Ihrem lokalen Netz, die Zonen-Dateien zu aktualisieren.

Nun müssen Sie noch die *dhcpd.conf* Ihres Linux-Servers so ändern, dass der DHCPD die Zonen-Dateien auch wirklich 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. Das Aktualisieren ist ja auch erst sinnvoll, wenn Sie einen eigenen Nameserver eingerichtet haben und betreiben.

Die Veränderungen am Nameserver erfolgen nicht nur virtuell, sondern dauerhaft, der Nameserver verändert dabei die Zonen-Dateien. Dabei setzt er z. B. auch die Generate-Zeile in die entsprechenden einzelnen Zeilen um, was die Dateien erheblich vergrößert. Sie sollten sich daher Kopien aller Zonen-Dateien anfertigen, um eventuelle Veränderungen leichter vornehmen zu können.