Aufbau und Konfiguration eines Name-Servers unter Linux Suse 9.2

Eine Projektarbeit
von
Jörg Arnold u. Frank Merkel

Im Rahmen der Weiterbildung zum staatlich gepr. Betriebswirt für Informationsverarbeitung u. –management
In der BBS Neustadt/Weinstraße

Betreuer: Alexander Scheib

Inhaltsverzeichnis

Prolog

Zu Beginn der sechziger Jahre, als die USA und die damalige UdSSR sich mitten im Kalten Krieg befanden und das Internet (bzw. *ARPAnet*) als solches noch gar nicht existierte, gab das amerikanische Verteidigungsministerium (*Department of Defense, DoD*) den Auftrag, ein dezentrales, ausfallsicheres und technisch fortschrittliches Netzwerk zu entwickeln und zu implementieren, um die vermeintliche Bedrohung durch die UdSSR möglichst gering zu halten.

Einige der renommiertesten Universitäten, Organisationen und Unternehmen arbeiteten an der Umsetzung und 1969 war es schließlich soweit. Vier Universitäts-Rechner in Kalifornien (*University of California*, Los Angeles, *University of California Santa Barbara*, *Stanford Research Institute*) und Utah (*University of Utah*) bildeten die Grundlage des neuen ARPAnets und konnten miteinander kommunizieren. In diesem Stadium der Entwicklung wurden die Informationen der verschiedenen Systeme noch in einer einzigen Datei, der »HOSTS.TXT«, verwaltet, von der alle angeschlossenen Systeme über identische Kopien verfügten. Diese Datei ist quasi ein Vorläufer der heutigen /etc/host-Datei.

Von der Idee und des dahinter steckenden Aufwandes war dies eine außerordentlich clevere und einfach zu handhabende Administrationsangelegenheit. Doch mit der Umstellung der Netzwerkprotokolle auf TCP/IP Mitte der siebziger Jahre stieg die Anzahl der zu verwaltenden Systeme immens an, womit die Aktualität der Datei »HOSTS.TXT« kaum mehr gewährleistet werden konnte und deren Umfang unpraktikable Ausmaße annahm.

Der ursprüngliche Ansatz, eine dezentrale Verwaltung zu erschaffen, wurde aufgrund des benötigten Überblicks bei der Vergabe neuer Rechnernamen und -adressen »ad absurdum« geführt.
So war es unumgänglich, dass in den achtziger Jahren - die Anzahl der verbundenen Rechner betrug mittlerweile über 1000 - ein neues System, das Domain Name System (DNS), entwickelt und eingeführt wurde

1 Geschichte

BIND (**B**erkeley **I**nternet **N**ame **D**omain) ist ein Open Source Software-Paket, mit dem auf Rechnern mit Standard-Betriebssystemen (z.B. UNIX, Linux, Windows NT) ein Domain Name System Server implementiert werden kann.

BIND kann kostenlos bezogen werden, der Source-Code ist veröffentlicht. Aufgrund seiner weiten Verbreitung und der zeitnahen Umsetzung der aktuellen DNS-RFCs gilt BIND seit Jahren als DNS-Referenz-Software.

Bevor es DNS gab, wurde die Auflösung von Namen in IP-Adressen über Listen (/etc/hosts.txt, vgl. /etc/hosts auf heutigen Unix-Systemen) vorgenommen, die auf jedem Rechner im Internet vorhanden sein mussten. Änderungen wurden zunächst manuell auf einem Masterserver durchgeführt und dann per Datei-Download an die einzelnen Rechner verteilt. Mit steigender Anzahl von IP-Teilnehmern wurde dieses Verfahren zunehmend unhandlicher.

1983 wurde von Paul Mockapetris das Domain Name System (DNS) spezifiziert. Im gleichen Jahr wurde die erste DNS-Software - *JEEVES* - auf einem DEC-Rechner implementiert. Wenig später gingen die ersten drei Internet Root-Server in Betrieb.

Anfang der 80er Jahre wurden an der Universität Berkeley an der Weiterentwicklung von UNIX gearbeitet. Einige Studenten begannen, für UNIX eine DNS-Software zu schreiben, die sie BIND (Berkeley Internet Domain System) tauften. BIND wurde ständig weiterentwickelt und die Version 4 wurde zum weltweiten Standard. Nachdem die Berkeley Universität die Weiterentwicklung der Software eingestellt hatte, wurde die Verantwortung für kurze Zeit von der Firma DEC und anschließend von *Vixie Enterprises* übernommen. Paul Vixie war zu dieser Zeit treibende Kraft hinter dem Projekt.

Ab der Version 4.9.3 ging BIND in die Verantwortung des Hersteller-unabhängigen ISC (Internet Software Consortium – ab 2004: Internet Systems Consortium) über. Die Version 8 wurde 1997 fertiggestellt. 1999 beauftragte ISC die Firma Nominum Inc., die Version 9 zu entwickeln. BIND 9 ist heute (Anfang 2004) Standard. Die Version 8 ist noch weitverbreitet, Version 4 gilt als veraltet.

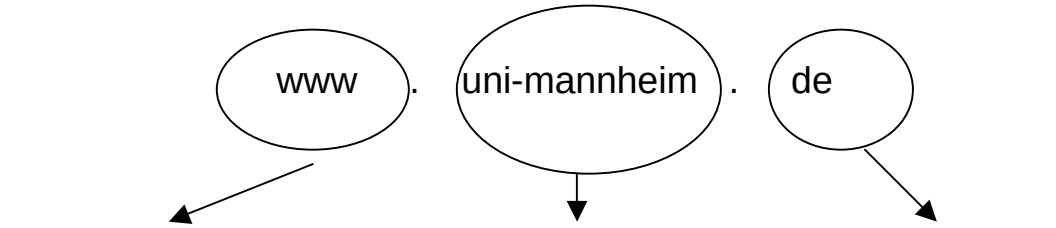
2 Ziele und Funktionen des DNS

Wie bereits erwähnt, übernimmt DNS die Aufgabe der Umwandlung von Internetnamen in numerische Internetadressen und umgekehrt. Diese Auflösung basiert auf der Verwendung des TCP/IP-Protokoll-Stacks.
Hierbei wird jedem öffentlich zugänglichen System (Host, Router, Gateway, ...) eine eindeutige 32-Bit große, binäre Zahl zugeordnet (IP-Adresse), über die es angesprochen werden kann. Das zurzeit verwendete IP-Protokoll ist das IPv4. Hier besteht jede IP-Adresse aus 32 Bits. Da solche Zahlenkolonnen in der Praxis von uns Menschen nur schwer zu benutzen sind, verwendet man hier meist die Punktierte-Dezimale Schreibweise, wobei die 32 Bits geteilt werden und diese jeweils dezimal notiert werden. Auf diese Weise besteht eine IP-Adresse aus vier sogenannten Quads:

Zur Vereinfachung des Umgangs mit IP-Adressen werden diese in vier Oktetts (Quad) unterteilt und in Form von durch Punkte getrennten *Dezimalzahlen* dargestellt:
(»11111111 11111111 11111111 11111111« ➔*255.255.255.255*«
»11000010 01001101 01111100 00100011« ➔ *194.77.124.35*«).

Jede Oktette repräsentiert (da als 1 Byte definiert) somit einen Werteumfang von 0..255 (2^8 = 256). Seltener anzutreffen ist die hexadezimale Notation (»FF.FF.FF.FF«), wobei diese vorrangig beim Nachfolger des jetzigen IP-Adressformats angewandt wird (IPv6). Da das menschliche Gehirn bekanntlich besser mit Namen als mit Zahlen umgehen kann, wird jeder dieser Adressen ein eindeutiger Namen zugeordnet. Ein Beispiel für eine Adresse nach dem DNS-System ist : www.uni-mannheim.de . Nach diesem System vergebene Adressen, werden als Uniform Resource Locator (URL) bezeichnet.

Beispiel:



So lässt sich grundsätzlich mit der IP-Adresse »216.239.39.101« als auch mit dem Namen »www.google.de« die gleiche Web-Seite auf dem gleichen System erreichen. Da die Vermittlung im Internet aber einzig auf IP-Adressen basiert, ist bei Verwendung von symbolischen Rechnernamen vorab eine Adressauflösung durchzuführen.

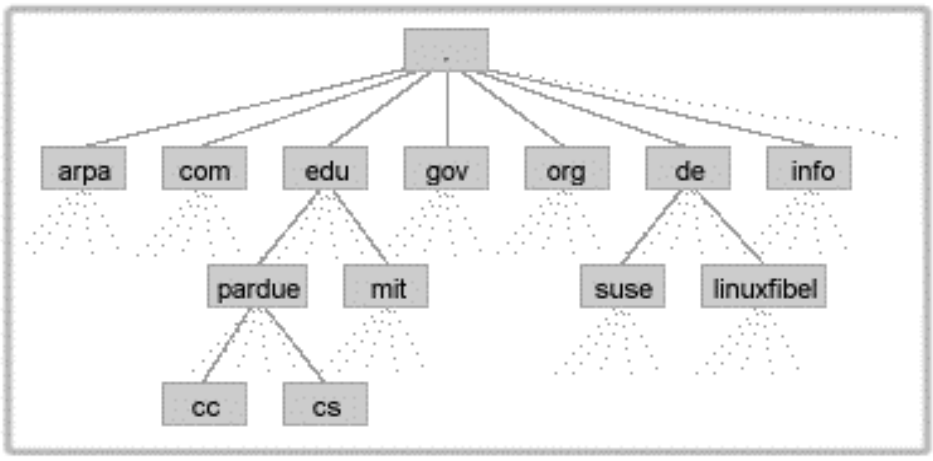


Abbildung 1: DNS-Namensraum

Um diese Auflösung zu verstehen, sollten Sie den Aufbau des DNS-Namensraumes kennen (Abbildung 1). Ausgehend von einer Wurzel (root), die die Bezeichnung ».« trägt, sind die Domains in einer baumartigen Struktur organisiert.

Die der Wurzel folgende Ebene umfasst die so genannten *Top Level Domains (TLD's)*. Eine Ebene tiefer folgen die Subdomains, denen entweder unmittelbar die Rechnernamen folgen oder aber eine weitere Ebene lokaler Domains, unterhalb derer dann die Rechnernamen liegen.

3 Top-Level-Domains

Die Top-Level-Domains (TLD) gliedern sich in zwei Kategorien ein. Zur Kategorie der so genannten *Generic TLD's* zählen »org« (Non-Profit Organisationen), »mil« (militärische Einrichtungen), »com« (Kommerzielle Unternehmen), »edu« (Bildungseinrichtungen), »net« (Netzorganisationen) und »gov« (Regierungsbehörden). Im Jahre 2000 erfuhren die generischen TLD's eine Erweiterung um »biz«, »info«, »coop«, »aero«, »name«, »pro« und »museum«.

Die zweite Kategorie beinhaltet die *länderspezifischen cTLD's*, wie bspw. »de« (Deutschland), »uk« (Grossbritannien), »aw« (Aruba), »cc« (Cocos Islands), »jp« (Japan) oder »ch« (Schweiz). Eine vollständige Liste aller TLD's wird von der *Internet Assigned Numbers Authority* (IANA), der Dachorganisation des Internets, verwaltet.
Die Abarbeitung eines Internetnamens, bspw. »www.linuxfibel.de«, erfolgt hierarchisch von rechts nach links, d.h. (vergleiche Abbildung 1) beginnend in der Wurzel führt der Zweig »de« (TLD) zum Zweig »linuxfibel« (registrierte Domain), der wiederum im Punkt »www« (Name eines Rechners der Domain »linuxfibel.de«) endet. Die Suche nach einem entsprechenden System erfolgt also stets baumabwärts beginnend im Root-Verzeichnis.

Korrekterweise müsste eine Domain immer mit einem abschliessenden Punkt, der das Root-Verzeichnis repräsentiert, geschrieben werden. Allerdings kann dies vernachlässigt werden, da inzwischen alle DNS-Werkzeuge hinreichend intelligent und tolerant sind, um trotzdem ein positives Ergebnis zurück zu melden. Bei einer Bind-Konfiguration wäre jedoch ein fehlender Punkt am Ende fatal, wie Sie später noch erfahren werden.

4 Forward- und Reverse-Lookup

Bei einer Namensauflösung wird i.d.R. das Äquivalent zu einem Internetnamen in Form einer Internetadresse, also eine Umsetzung von Namen nach Adresse, gesucht

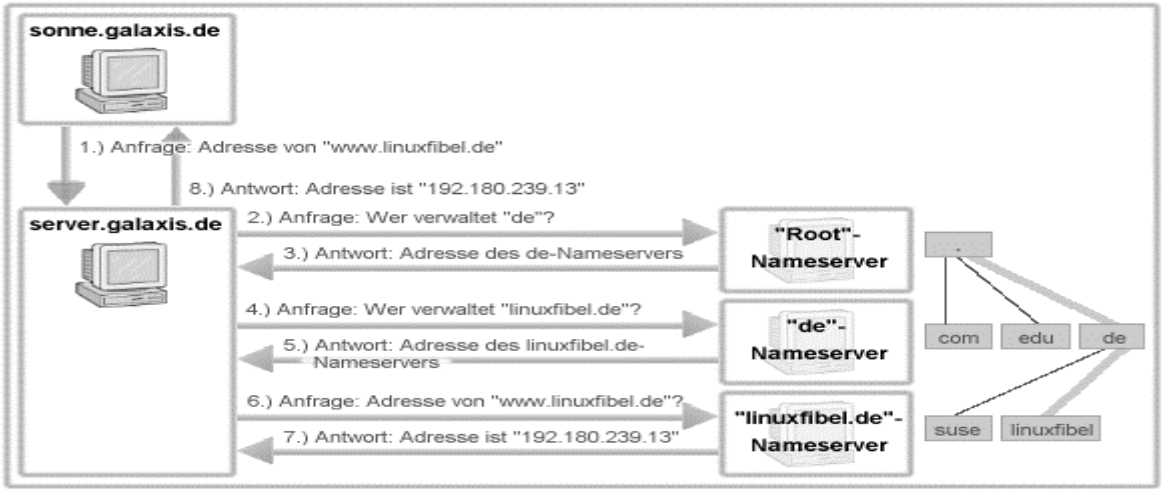


Abbildung 2: Auflösung der IP-Adresse eines Rechnernamens

Bedarf eine Anwendung der IP-Adresse zu einem Rechnernamen (»www.linuxfibel.de« vergleichen Sie das Beispiel in Abbildung 2), so

übernimmt der so genannte Resolver, eine Sammlung von Funktionen aus der C-Bibliothek, die weitere Recherche.
In typischen Konfigurationen wird der Resolver eines Clients (»sonne.galaxis.de«) zunächst die lokale Datei »/etc/hosts« nach entsprechenden Einträgen durchforsten. Wird er dort nicht fündig, reicht er die Anfrage an den zuständigen Nameserver weiter (in Abbildung 2 als »server.galaxis.de« bezeichnet).
Jeder Nameserver verfügt über einen Cache, indem er die Daten der zuletzt recherchierten Anfragen eine Zeit lang zwischenspeichert. Erst wenn der lokale Nameserver die Adresse nicht in seinem Cache hat, beginnt der mit der Auflösung des Namens von $raquo;hinter her«. D.h. er fordert einen der Rootserver an (eine Liste solcher hält der Nameserver in einer Konfigurationsdatei), ihm die Adresse des für die Domain »de« zuständigen Nameservers mitzuteilen. An die gelieferte Adresse sendet er die folgende Anfrage nach der Adresses des für »linuxfibel.de« zuständigen Nameservers.
Bei letzterem Server erhält er schließlich die gewünschte Informationen, die er sowohl in seinen Cache einträgt, als auch zum anfragenden Clientrechner weiterleitet.

Zur Auflösung in die Gegenrichtung (Adresse in Namen), das sogenannte Reverse Lookup, wurde eine neue Domain »in-addr.arpa« (***A****ddress and* ***R****outing* ***P****arameter* ***A****rea domain*) eingeführt. Unterhalb dieser speziellen Domain existieren 256 Subdomains (0..255). Insgesamt wiederholt sich diese Unterteilung viermal, bis die 32-Bit-Adresse vollständig dargestellt ist.

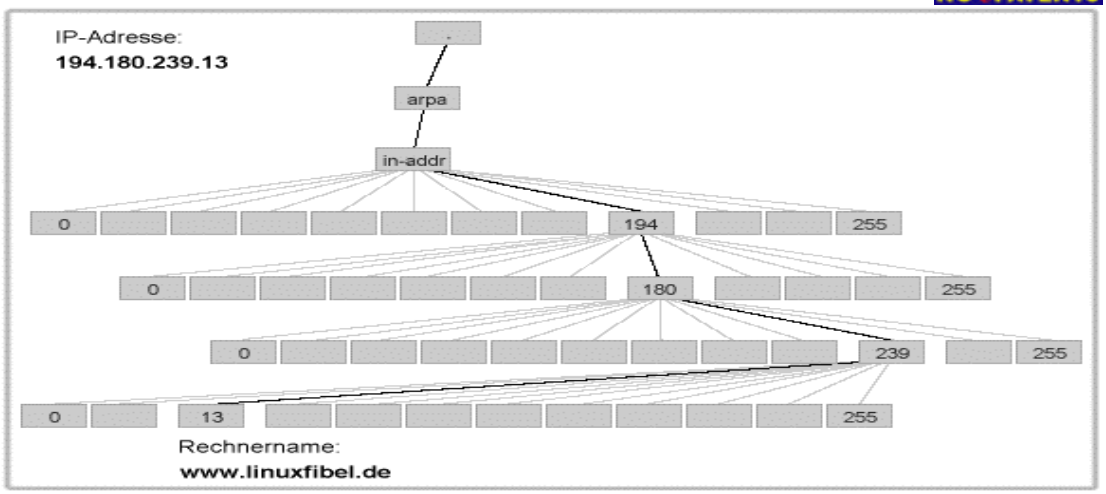


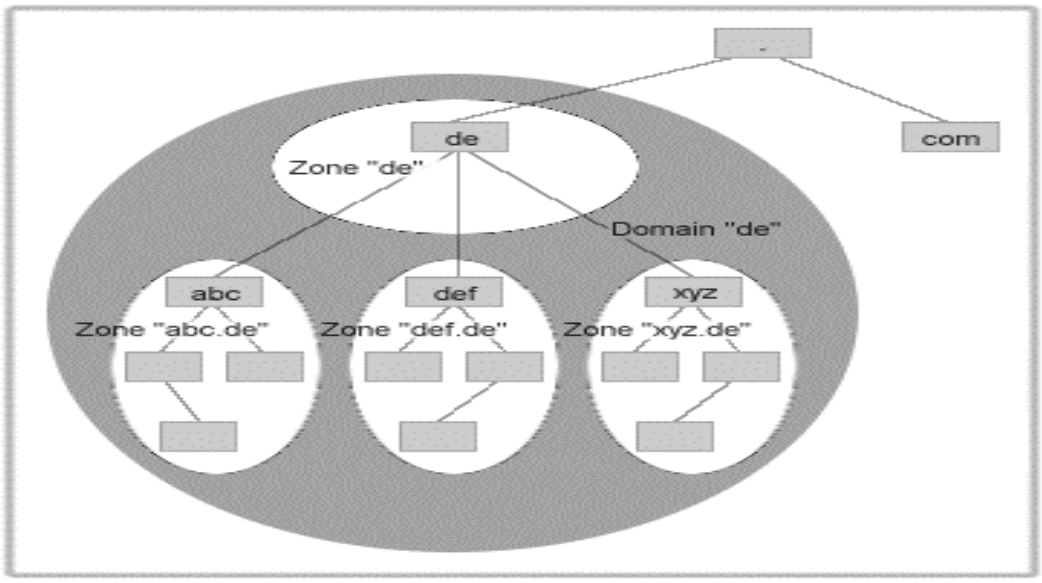
Abbildung 3: Reverse-Lookup mit Hilfe der Pseudodomain »in-addr.arpa«

Für das kommende IP-Adressformat IPv6 wurde die Domain »ip6.arpa« eingeführt, die nach einem ähnlichen Prinzip funktioniert, wie »in-addr.arpa« für IPv4 (offizielle Bezeichnung des IP für 32 Bit Adressen).
Anmerkung: Eine weitere Pseudodomain »e164.arpa« dient der Recherche von *whois*-Anfragen.

5 Unterschiede zwischen Domains und Zonen

Eine sehr wichtige Unterteilung in diesem Schema ist, um Verwirrungen auszuschließen, der Unterschied zwischen Domain und Zone.
Leider ist dieser Unterschied nicht allzu groß und daher auch nicht allzu offensichtlich, aber immerhin vorhanden.
Eine Domain beinhaltet alle unter ihm liegenden Domains, Zonen und Systeme (Hosts), d.h. alle baumabwärts liegenden Verwaltungseinheiten gehören zu dieser Domain. Da dies aber keinerlei Sinn macht alles auf dieser Ebene zu verwalten, wurden sogenannte Zonen eingerichtet. Diese Zonen konnten nun an die verschiedenen Einrichtungen und Unternehmen abgegeben werden, welche die Verwaltung übernehmen und neue Knoten

zeitnaher und schneller einbinden konnten. Jede Zone ist in sich abgeschlossen und beinhaltet nur die Knoten, die direkt zu ihr gehören.
Sollten weitere Unterteilungen benötigt werden, so werden neue Zonen errichtet, die jedoch völlig unabhängig voneinander sind.



An dieser Stelle werden Namensserver (Nameserver) eingesetzt um die Verwaltung der Zonen zu übernehmen. Hierbei kann jedem Nameserver eine oder auch mehrere Zonen delegiert werden, für die er zuständig ist. Man spricht auch von "Autorität" einer Zone.
Generell besitzt jede Zone einen primären Nameserver ("primary Master", "Primary") und evtl. mehrere sekundäre Nameserver ("secondary Master", "Secondary"), die die Zonendaten untereinander austauschen, um auf dem neuesten Stand zu bleiben.
Eine Domain ist also ein logischer Oberbegriff, der zwar im täglichen Sprachgebrauch sehr häufig verwendet wird, im DNS wird jedoch lediglich mit Zonen gearbeitet. Oftmals findet auch keinerlei Unterscheidung statt, so dass die beiden Begriffe äquivalent behandelt werden.

6 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).
`/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.

7 Die Dateien

Bearbeiten können Sie die Datei entweder direkt mit einem Texteditor (joe) oder die Funktion im YaST-Kontrollzentrum, die Sie unter Netwerkdienste – Hostnamen finden

7.1 /etc/hosts

In dieser Datei existiert neben dem localhost Eintrag, der stets eingetragen sein sollte, noch die IP-Adresse und Bezeichnung für das Netz, welches ebenfalls einzutragen ist.
So kann der Server zumindest seine eigenen Adressen immer auflösen.
Im YaST-Kontrollzentrum sollte man jetzt die IP- Adresse (172.16.111.113), sowie den Domain-Namen (linux-fsinf03.de) eintragen. Wichtig ist, dass die Checkboxen für DHCP deaktiviert sind, der der Server seine Daten ja nicht von einem anderen System beziehen soll. YaST verändert dann die Datei.

7.2 /etc/host.conf

Diese Datei legt über den Befehl order, die Reihenfolge fest, mit der die Namensauflösung erfolgt.
Die Angabe order hosts, bind bedeutet, der Resolver fragt erst
die Datei /etc/hosts ab, bevor er die Namensauflösung über DNS probiert.
Der Eintrag multi on bewirkt, dass man zu einem Rechnernamen in der /etc/hosts mehrere IP-Adressen angeben darf.

7.3 /etc/resolv.conf

```
    I    /etc/resolv.conf
nameserver 172.16.111.113
search linux.fsinf03
```

Die beiden Zeilen in dieser Datei bewirken, dass für die Suche nach Rechnern der Domain *linux-fsinf03.de* der Nameserver *172.16.111.113* 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 Konfiguartionsdateien ein.

Die von SuSE installierten Musterdateien können Sie für Ihre Bedürfnisse anpassen. Eine umfangreiche Dokumentation zum Nameserver Bind findet sich in Ordner */usr/share/doc /packages/bind9*.
Für jeden zu befragenden Nameserver bedarf es eines eigenen Nameserver-Eintrags in der Datei »/etc/resolv.conf«. Entsprechend der Reihenfolge der Einträge werden die Nameserver nacheinander kontaktiert, bis der erste erfolgreich eine Anfrage beantwortet.
In den gängigen Implementierungen sind maximal drei Nameserver-Angaben zulässig. Fehlt ein Nameserver-Eintrag, so wird die Anfrage an den lokalen Rechner gerichtet. Dieser muss zumindest als Caching-only-Nameserver konfiguriert sein (sonst schlägt jede Anfrage fehl).
In unserem Beispiel haben wir nur einen Nameserver (172.16.111.113) und eine Suchliste (linux-fsinf03.de) eingetragen.
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

7.4 /etc/named.conf

Diese Datei stellt die Hauptkonfigurationsdatei dar.
Am Anfang dieser Datei wird betont das es sich um eine Konfigurationsdatei für das aktuelle Bind9 und nicht für ältere Versionen handelt.
Zeilen, die mit dem Lattenzaun „#“ beginnen, sind Kommentare.

Das Options-Statement gibt zuerst den Pfad zu den weiteren Konfigurationsdateien an. „directory „/var/lib/named“; Anfragen die der Nameserver nicht beantworten kann, werden an die Nameserver weiter gegeben, die im forwarders-Statement aufgeführt sind.
Das Statement „allow-query“ gibt an von wo aus auf dem
In diesem Fall ist der Zugriff auf dem Nameserver nur aus dem lokalen Netz heraus und vom Server selber zugelassen.
Äußerst wichtig sind die Zonen-Statements an Ende der Datei /etc/named.conf

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

Dieses erste Zonen-Statement enthält die IP-Adressen der Root-Nameserver und den Verweis auf die root.hint Datei.

7.5. /var/lib/named/root.hint

Die Adressen der Root-Nameservers finden sich in der Datei /var/lib/named/root.hint. Die mitgelieferte Datei braucht man normalerweise nicht zu ändern. Dessen Adresse muss jeder Nameserver zwingend kennen. Deshalb liegt jedem Bind-Paket eine Datei bei, die die Liste der Root-Server enthält. Die Adressen der Root-Servers werden i.d.R. niemals geändert, sodass Sie von der Aktualität der Liste ausgehen können.


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

Die Localhost-Zone ist notwendig, damit der Server auch den Namen localhost zu 127.0.0.1 auflösen kann.

```
zone "0.0.127.in-addr.arpa" in {
  type master;
  file "127.0.0.zone";
};
zone "113.111.16.172.in-addr.arpa" in {
  type master;
  file "tavirp.zone";
};
```

Die Reverse-Zone für den lokalen Host, ist für die Rückwärtsauflösung von 127.0.0.1 zu localhost zuständig. Zur Auflösung in die Gegenrichtung (Adresse in Namen), das sogenannte *Reverse Lookup*, wurde eine neue Domain »in-addr.arpa« (***A****ddress and* ***R****outing* ***P****arameter* ***A****rea domain*) eingeführt.
Im vorliegenden Beispiel hat *linux-fsinf03.de* die IP-Adresse *113.111.16.172*, diese Zuordnung ergibt sich aus der Zonen-Datei *private.zone*. Für die Rückwärtsauflösung von *172.16.111.113* zu *linux-fsinf03.de* ist diese Datei zuständig. Die Rückwärtsauflösung soll auch das *tavirp* (privat rückwärts gelesen) andeuten.


```
zone "linux-fsinf03.de" in {
  type master;
  file "privat.zone";
};
```

Mit dem Zone-Statement bekommt der Nameserver die Zuständigkeit für *linux-fsinf03.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/private.zone* (s. u.).

8 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 unvollstädige 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 Standart-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 *private-zone*.
Alle weiteren Zeilen sind dann Ressource Records mit folgendem Aufbau:

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 Zeitangabe 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
linux.linux-fsinf03.de. (beachten Sie den Punkt am Ende) oder
linux (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 primär Rechner ausfällt.

8.1 /var/lib/named/privat.zone

```
/var/lib/named/privat.zone
$TTL 2d
@                   IN SOA          linux.linux.fsinf03.    postmaster.linux.fsinf03. (
                                    2024070400      ; serial
                                    3h              ; refresh
                                    15m             ; retry
                                    1w              ; expiry
                                    1d )            ; minimum

linux.fsinf03.      IN NS           linux
linux.fsinf03.      IN MX 0         linux
linux               IN A            172.16.111.113
www                 IN CNAME        linux
www2                IN CNAME        linux
mail                IN CNAME        linux
ns                  IN CNAME        linux
ftp                 IN CNAME        linux
news                IN CNAME        linux
;
windo               IN A            172.16.111.112
wind                IN A            172.16.111.114
```

Der Platzhalter „@“ steht hier für den Rechner selber, also für linux.

linux ist Nameserver und Mailserver mit höchster Priorität für die Domain *linux-fsinf03.de*. Weiter bestimmt die Datei die IP-Adressen für *windo, wind*.

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

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. Geht man so vor, braucht man bei späteren Erweiterungen des Netzes keine

Einträge im Nameserver zu ändern. Genau diese Zeilen erzeugt die $GENERATE-Direktive.

Als Alias für linux sind *www, mails, ns, ftp* und *news* eingetragen. In einem lokalen Netz ist das praktisch. Für Rechner, die ständig mit dem Internet verbunden sind, besteht ein Sicherheitsrisiko, denn wenn Rechnernamen über Rechnerfunktionen informieren freuen sich Eindringlinge.

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.

8.2 /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
```

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 Dabei, die dadurch etwas unübersichtlich wirkt, da SuSE hier mit Platzhaltern arbeitet, um die Datei allgemeingültig zu halten. Der Platzhalter „@“ steht hier für den Rechner selbst, also chef.linux-fsinf03.de.

9 Von der IP-Nummer zum Hostnamen

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 Reccord-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.

9.1 /var/lib/named/tavrip.zone

```
$TTL 2D
$GENERATE 115-127 $ PTR client-$.linux-fsinf03.de.
@               IN SOA          chef.linux-fsinf03.de.  postmaster.linux-fsinf0
                                2024070400      ; serial (12.07.2003 Version 03
                                3H              ; refresh
                                15M             ; retry
                                1W              ; expiry
                                1D )            ; minimum

                IN NS           chef.linux-fsinf03.de
2               IN PTR          chef.linux-fsinf03.de
112             IN PTR          windo.linux-fsinf03.de
114             IN PTR          wind.linux-fsinf03.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 Zurordnung *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.

9.2 /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.
```

10 Erster Start des Nameservers

Durch die Eingabe »*rcnamed start*« wird der Nameserver gestartet.
Sie finden in der Datei */var/log/messages* Meldungen wie:


```
Oct 11 21:23:03 chef named[5024]: shutting down: flushing changes
Oct 11 21:23:03 chef named[5024]: stopping command channel on 127.0.0.1#953
Oct 11 21:23:03 chef named[5024]: stopping command channel on ::1#953
Oct 11 21:23:03 chef named[5024]: no longer listening on 127.0.0.1#53
Oct 11 21:23:03 chef named[5024]: no longer listening on 172.16.111.113#53
Oct 11 21:23:03 chef named[5024]: exiting
Oct 11 21:23:03 chef named[5141]: starting BIND 9.2.4 -t /var/lib/named -u named
Oct 11 21:23:03 chef named[5141]: using 1 CPU
Oct 11 21:23:03 chef named[5141]: loading configuration from '/etc/named.conf'
Oct 11 21:23:03 chef named[5141]: listening on IPv4 interface lo, 127.0.0.1#53
Oct 11 21:23:03 chef named[5141]: listening on IPv4 interface eth0, 172.16.111.113#53
Oct 11 21:23:03 chef named[5141]: command channel listening on 127.0.0.1#953
Oct 11 21:23:03 chef named[5141]: command channel listening on ::1#953
Oct 11 21:23:03 chef named[5141]: zone 0.0.127.in-addr.arpa/IN: loaded serial 42
Oct 11 21:23:03 chef named[5141]: zone 111.16.172.in-addr.arpa/IN: loaded serial
2024070400
Oct 11 21:23:03 chef named[5141]: zone linux-fsinf03.de/IN: loaded serial 2024070400
Oct 11 21:23:03 chef named[5141]: zone localhost/IN: loaded serial 42
Oct 11 21:23:03 chef named[5141]: running
```

- Die erste Zeile ist eine allgemeine Start-Meldung des Nameservers, aus der sich vor allem die Versionsnummer, hier *9.2.4*, ergibt.

- Danach listet die Datei die IP-Adressen, auf die der Nameserver anspricht, *172.16.111.113* 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.

Sollte der Nameserver nicht richtig starten, so gibt er seine Fehlermeldungen ebenfalls 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.
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.
Sollten jetzt dennoch Probleme bestehen, dann kontaktieren Sie einfach den Mann der seines gleichen sucht, Dipl.Ing. Alexander Scheib www.scheib-info.de selbstlose Hilfe ist ihnen sicher.

11 Test des DNS-Servers

Der Nameserver wurde erfolgreich gestartet (running), wenn man mit „host“ Anfragen auf dem Linux-Server testet und diese lokalen Anfragen keine Fehlermeldung ausgeben.

Diese „Host“-Tests können wie folgt aussehen:

```
chef:/ # host www
www.linux-fsinf03.de is an alias for chef.linux-fsinf03.de.
chef.linux-fsinf03.de has address 172.16.111.113
```

```
chef:/ # host localhost
localhost has address 127.0.0.1
```

```
chef:/ # host 172.16.111.112
112.111.16.172.in-addr.arpa domain name pointer windo.linux-
fsinf03.de.111.16.172.in-addr.arpa.
```

```
chef:/ # host 127.0.0.1
```

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

Nach erfolgreichen Tests braucht man nur noch im 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 beim Hochfahren des Systems in diesen Runleveln automatisch.

Literaturverzeichnis

- http://www.linuxfibel.de/dns_srv.htm (Stand 01.09.2005)
- http://www.afokken.de/linux/lxportf.htm (Stand 30.08.2005)
- http://www.wolfgarten.com/downloads/bind.pdf (Stand 23.08.2005)
- http://de.wikipedia.org/wiki/Nameserver#Resolver (Stand 16.08.2005)
- http://www.ag-intra.net/linux-al-dns8.html (Stand 16.08.2005)
- http://www.afokken.de/linux/lxnames.htm (Stand 13.08.2005)
- http://www.klaus.franken.de/DE-ISDN-HOWTO/html/DE-ISDN-HOWTO-8.html (Stand 13.08.2005)
- http://www.selflinux.org/selflinux/html/dns02.html (Stand 09.08.2005)
- Matt Welsh, Lar Kaufman. Running LINUX. O'Reilly & Associates. 1996. A classic LINUX book. Covers LINUX system administration in-depth. On-line go to sunsite.unc.edu, redhat.com
- Greg Lehey. The Complete FreeBSD. Walnut Creek. 1998. Big and new FreeBSD book. On-line use freebsd.org.
- Liu, Cricket, and Albitz, Paul. DNS and BIND. O'Reilly & Associates. 1998 Great DNS book. You may need it to run your own DNS server. Covers BIND 8.