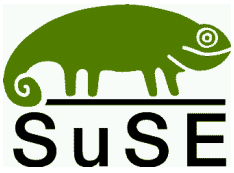
Daniel Bischof, Frank Bodammer, Olaf Donjak, Roman Drahtmüller, Torsten Dubiel, Karl Eichwalder, Viviane Glanz, Carsten Groß, Roland Haidl, Dirk Lerner, Lars Müller, Jordi Jaen Pallares, Edith Parzefall, Manuela Piotrowski, Peter Pöml, Peter Reinhart, Thomas Schraitle.

# SuSE Linux

# Netzwerk




SuSE GmbH
Schanzäckerstr. 10
D-90443 Nürnberg
Tel.: (09 11) 7 40 53 31 (Vertrieb)
Fax.: (09 11) 7 41 77 55 (Vertrieb)
E-Mail: suse@suse.de
WWW: http://www.suse.de

**Daniel Bischof, Frank Bodammer, Olaf Donjak, Roman Drahtmüller, Torsten Dubiel, Karl Eichwalder, Viviane Glanz, Carsten Groß, Roland Haidl, Dirk Lerner, Lars Müller, Jordi Jaen Pallares, Edith Parzefall, Manuela Piotrowski, Peter Pöml, Peter Reinhart, Thomas Schraitle.**

SuSE Linux
Netzwerk

2. Auflage 2001
SuSE GmbH



Satz: LaTeX
Geeko-Icons von Rolf Vogt.

# Inhaltsverzeichnis

# 1 Vorwort

Das SuSE Linux Netzwerk-Handbuch bietet Ihnen architekturunabhängiges Wissen rund um SuSE Linux im Netzwerk. Mit diesem Buch wenden wir uns sowohl an Linux-Anwender, die noch keine Erfahrung mit dem Einrichten von Netzwerken haben, als auch an erfahrene System-Administratoren, die sich über Teilbereiche oder Besonderheiten von SuSE Linux im Netzwerk informieren wollen. Der erste Teil beschäftigt sich mit der Konfiguration des Netzwerks, anschließend zeigen wir Ihnen den Weg ins Internet. Danach werden verschiedene Anwendungen wie FTP, der Webserver Apache und der Proxy-Server Squid vorgestellt. Das letzte Kapitel dieses Handbuchs beschäftigt sich mit dem wichtigen Thema Sicherheit, Firewalls und Masquerading.

Wir legen den Begriff Installationssupport großzügig aus, aber Sie werden sicher verstehen, dass wir zum Preis eines SuSE Linux-Pakets nicht die Administration Ihres Firmennetzes übernehmen können. Wir lassen Sie aber trotzdem mit Ihren Aufgaben nicht allein und bieten Ihnen unter

`http://support.suse.de`

reichlich Informationen und Hilfestellungen rund um SuSE Linux. Auf dieser Seite finden Sie auch die Angebote unseres kostenpflichtigen Business-Supports.

Have a lot of fun

Ihr SuSE Team

# 1 Vorwort

# 2 Linux im Netzwerk

Heute ist die Anzahl miteinander vernetzter Computer bereits so groß, dass ein einzelner, nicht zumindest zeitweise vernetzter Computer selten geworden ist. Linux bietet Ihnen hier alle Voraussetzungen und notwendigen Netzwerktools zur Einbindung in diverse Netzwerkstrukturen.

Nachfolgend wird eine Einführung in das normalerweise von Linux verwendete Protokoll TCP/IP gegeben. Dann werden die verschiedenen Dienstleistungen und auch besonderen Eigenschaften des Protokolls vorgestellt. Den Abschluss bildet eine Übersicht über den zukünftigen Standard IPv6 und die Unterschiede und Vorzüge gegenüber dem bestehenden IPv4.

Nach der Vorstellung der Grundlagen zeigen wir Ihnen die Einrichtung eines Netzwerkzugangs mit einer Netzwerkkarte unter SuSE Linux mit Hilfe von YaST und YaST2. Es werden die zentralen Konfigurationsdateien besprochen und einige der wichtigsten Tools aufgeführt.

Praktisch die gesamte Netzwerkkonfiguration können Sie mit YaST2 oder mit YaST durchführen (siehe Abschnitt 2.3.1 auf Seite 18 und Abschnitt 2.3.2 auf Seite 20); da jedoch gerade die Konfiguration eines Netzwerks beliebig komplex sein kann, werden in diesem Kapitel nur die grundlegenden Mechanismen und die für die Konfiguration des Netzwerks relevanten Dateien beschrieben.

## 2.1 TCP/IP - Das von Linux verwendete Protokoll

Linux und andere Unix-Betriebssysteme verwenden das TCP/IP-Protokoll. Genau genommen handelt es sich um eine Protokollfamilie, die ganz unterschiedliche Dienstleistungen bietet. TCP/IP wurde aus einer militärischen Anwendung heraus entwickelt und in der heute verwendeten Form ca. 1981 in einem so genannten RFC festgelegt. Bei RFC (engl. *Request for comments*) handelt es sich um Dokumente, die die verschiedenen Internetprotokolle und die Vorgehensweise bei der Implementierung des Betriebssystems und von Applikationen beschreiben. Auf diese RFC-Dokumente können Sie direkt über das Web zugreifen, die URL lautet `http://www.ietf.org/`. Sie finden diese Dokumente auch in der SuSE Linux Distribution im Paket `rfc`, Serie `doc` (Dokumentation). In der Zwischenzeit sind einige Verfeinerungen am TCP/IP Protokoll vorgenommen worden, am grundlegenden Protokoll hat sich seit 1981 aber nichts geändert.

**Tipp**

Die RFC Dokumente beschreiben den Aufbau der Internet Protokolle. Falls Sie Ihr Know-how über ein bestimmtes Protokoll vertiefen wollen, ist das passende RFC Dokument die richtige Anlaufstelle:
`http://www.ietf.org/rfc.html`

Die in Tabelle 2.1 genannten Dienste stehen zur Verfügung, um Daten zwischen zwei Linuxrechnern über TCP/IP auszutauschen:

| | |
|---|---|
| TCP | (engl. *Transmission control protocol*) Ein verbindungsorientiertes, gesichertes Protokoll. Die zu übertragenden Daten werden aus der Sicht der Applikation als Datenstrom verschickt und vom Betriebssystem selbst in das passende Übertragungsformat gebracht. Die Daten kommen bei der Zielapplikation auf dem Zielrechner als exakt der Datenstrom an, als der sie abgeschickt wurden. TCP stellt sicher, dass unterwegs keine Daten verloren gehen und nichts durcheinander kommt. TCP wird dort verwendet, wo die Reihenfolge der Daten wichtig ist und der Begriff Verbindung Sinn macht. |
| UDP | (engl. *User Datagram protocol*) Ein verbindungsloses, ungesichertes Protokoll. Die zu übertragenden Daten werden paketorientiert verschickt, die Datenpakete werden dabei schon von der Applikation erzeugt. Die Reihenfolge der Daten beim Empfänger ist nicht garantiert, ebenso kann es passieren, dass einzelne Datenpakete verloren gehen. UDP eignet sich für datensatzorientierte Applikationen und bietet kleinere Latenzzeiten als TCP. |
| ICMP | (engl. *Internet control message protocol*) Im Wesentlichen ist das kein für den Benutzer verwendbares Protokoll, sondern ein spezielles Steuerprotokoll, das Fehlerzustände übermittelt und das Verhalten der an der TCP/IP-Datenübertragung beteiligten Rechner steuern kann. Zusätzlich wird durch ICMP noch ein spezieller Echo-Modus bereitgestellt, den man mit dem Programm ping prüfen kann. |
| IGMP | (engl. *Internet group management protocol*) Dieses Protokoll steuert das Verhalten von Rechnern bei der Verwendung von IP-Multicast. Leider kann IP-Multicasting in diesem Rahmen nicht vorgestellt werden. |

Tabelle 2.1: Verschiedene Protokolle der TCP/IP Protokollfamilie

Fast alle Hardwareprotokolle arbeiten paketorientiert. Die zu übertragenden Daten müssen in kleine „Päckchen“ gepackt werden und können nicht „in einem Rutsch“ verschickt werden. Deshalb arbeitet auch TCP/IP mit kleinen Datenpaketen. Die Maximalgröße eines TCP/IP Paketes ist knapp 64 Kilobyte. In der Praxis sind die Pakete normalerweise viel kleiner, da die Netzwerkhardware der limitierende Faktor ist. So ist die zulässige Maximalgröße eines Datenpaketes auf dem Ethernet ca. 1500 Byte. Dementsprechend wird die Paketgröße

des TCP/IP Pakets begrenzt, wenn die Daten über ein Ethernet geschickt werden. Will man mehr Daten übertragen, müssen vom Betriebssystem entsprechend mehr Datenpakete verschickt werden.

Wenn man ganz genau ist, dann sollte es eigentlich nicht TCP/IP-Protokoll heißen, sondern nur IP-Protokoll. Über IP (engl. *Internet protocol*) findet eine ungesicherte Datenübertragung statt. TCP (engl. *Transmission control protocol*) ist gewissermaßen nur ein Aufsatz auf das darunter liegende IP, um eine gesicherte Übertragung der Daten zu garantieren. IP selbst ist wiederum ein Aufsatz auf das darunter liegende, hardwareabhängige Protokoll, zum Beispiel Ethernet.

### 2.1.1 Schichtenmodell

Diese Sache mit den „Aufsätzen“ hat auch einen richtigen Namen: Kenner sprechen hier vom „Schichtenmodell“. Eine Netzwerkarchitektur kann man in verschiedene Schichten (engl. *Layer*) unterteilen. Vergleichen Sie hierzu die Abbildung 2.1.

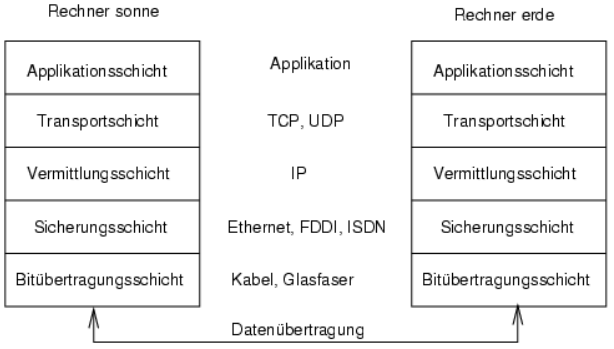



Abbildung 2.1: Vereinfachtes Schichtenmodell für TCP/IP

In der Abbildung sind jeweils ein oder zwei Beispiele für die jeweilige Schicht erwähnt. Wie Sie sehen, sind die Schichten nach „Abstraktionsebenen“ geordnet, die unterste Schicht ist sehr nah an der Hardware. Die oberste Schicht hingegen abstrahiert die darunter liegende Hardware nahezu vollständig. Jede der Schichten hat eine ganz spezielle Funktion, die zum Großteil schon aus der Bezeichnung hervorgeht. So wird das verwendete Netzwerk (z. B. Ethernet) durch die Bitübertragungsschicht und die Sicherungsschicht verkörpert.

- Während sich Schicht 1 mit solchen Dingen wie Kabeltypen, Signalformen, Signalkodierung und ähnlichem beschäftigt ist Schicht 2 für das Zugriffsverfahren (Welcher Rechner darf wann Daten schicken?) und eine Fehlerkorrektur (Datensicherung - deshalb **`Sicherungsschicht`**) zuständig. Die Schicht 1 nennt man die **`Bitübertragungsschicht`**.
- Schicht 3 wiederum, die **`Vermittlungsschicht`** ist für die Datenübertragung über weite Strecken verantwortlich. Die Vermittlungsschicht stellt sicher, dass die Daten auch über weite Strecken beim richtigen Empfänger ankommen und zugestellt werden können.

- Schicht 4, die **`Transportschicht`**, ist für die Daten der Applikation verantwortlich und stellt sicher, dass die Daten in der richtigen Reihenfolge ankommen und nicht verloren gehen. Die Sicherungsschicht ist nur dafür verantwortlich, dass die ankommenden Daten korrekt sind. Gegen das „Verlieren“ von Daten schützt die **`Transportschicht`**.
- Schicht 5 schließlich ist die Datenverarbeitung durch die Applikation selbst.

Damit jede der Schichten die ihr zugeteilte Aufgabe erfüllen kann, müssen zusätzliche Informationen der jeweiligen Schicht im Datenpaket im **`Header`**, dem Kopf des Datenpakets, gespeichert werden. Jede der Schichten fügt einen kleinen Datenblock, den sog. „Protokollkopf“ (engl. *Protocol header*), an das im Entstehen begriffene Paket vorne dran. Schauen wir uns also einmal ein beliebiges TCP/IP-Datenpaket an, das auf einem Ethernetkabel unterwegs ist, so setzt sich dieses wie in Bild 2.2 abgebildet zusammen.

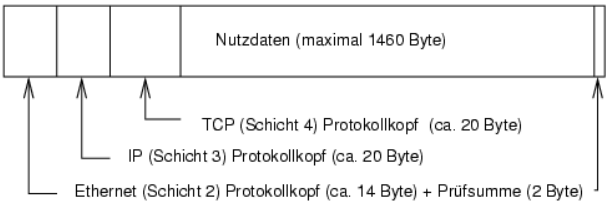



Abbildung 2.2: TCP/IP Paket im Ethernet

Wie Sie sehen, ist die Welt nicht perfekt und ohne Ausnahme. Die Prüfsumme der Sicherungsschicht befindet sich am Ende des Pakets und nicht am Anfang. Dies bringt aber für die Netzwerkhardware eine Vereinfachung. Die maximal mögliche Menge der Nutzdaten in einem Paket beträgt im Ethernet-Netzwerk 1460 Byte.

Möchte eine Applikation also Daten über das Netzwerk verschicken, durchlaufen die Daten die einzelnen Schichtebenen, die alle im Linuxkernel (Ausnahme Schicht 1: Netzwerkkarte) implementiert sind. Jede der Schichten ist dafür verantwortlich, die Daten so aufzubereiten, dass sie an die jeweils darunter liegende Schicht weitergereicht werden können. Die unterste Schicht ist schließlich für den eigentlichen Datenversand zuständig. Beim Empfang läuft das ganze nun umgekehrt ab. Wie bei den Schalen einer Zwiebel werden von jeder Schicht die Protokollköpfe von den Nutzdaten entfernt. Schicht 4 ist dann letztendlich dafür verantwortlich, die Daten für die Applikation auf dem Zielrechner bereitzustellen. Dabei kommuniziert eine Schicht immer nur mit der Schicht direkt über oder unter ihr. Für eine Applikation ist es also irrelevant, ob die Daten über ein 100-MBit/s-FDDI-Netzwerk oder über eine 56-KBit/s-Modemleitung übertragen werden. Umgekehrt ist es für die Datenübertragungsleitung egal, welche Daten eigentlich verschickt werden, solange sie richtig verpackt sind.

Das Schöne an dem Schichtenmodell ist also, dass eine Schicht praktisch unabhängig von der jeweils anderen ist. Wenn man zum Beispiel einen Verstärker für die elektrischen Signale auf der Leitung einsetzt (so etwas gibt es wirklich und nennt sich **`Repeater`**), so muss er nur etwas von Schicht 1 und 2 verstehen, um das Paket richtig zu regenerieren.

Um die Pakete richtig weiterzuleiten, muss ein so genanntes „Gateway“ nur die Protokolle bis Schicht 3 verstehen, also IP. Ein Zusammenhang der Pakete o.ä. ist für die korrekte Weiterleitung der Pakete nicht wichtig.

### 2.1.2 IP-Adressen und Routing

#### IP-Adressen

Jeder Computer im Internet hat eine eindeutige 32-Bit-Adresse. Diese 32 Bit beziehungsweise 4 Byte werden normalerweise wie in Tabelle 2.2 in der zweiten Zeile abgebildet geschrieben:

| | | | | |
|---|---|---|---|---|
| IP-Adresse (binär): | 11000000 | 10101000 | 00000000 | 00010100 |
| IP-Adresse (dezimal): | 192. | 168. | 0. | 20 |

Tabelle 2.2: Schreibweise IP-Adresse

Die vier Bytes werden also im dezimalen Zahlensystem durch einen Punkt getrennt nebeneinander geschrieben. Genauer gesagt bekommt nicht nur ein Rechner eine IP-Adresse, sondern sogar jede Netzwerkschnittstelle in diesem Rechner. Die IP-Adresse ist einem Rechner bzw. einer Netzwerkschnittstelle zugeordnet, sie kann also nicht woanders auf der Welt nochmals verwendet werden. Ausnahmen von diesen Regeln gibt es zwar, spielen aber bei der folgenden Betrachtung erst einmal keine Rolle.

Auch die Ethernetkarte besitzt selbst eine eindeutige Adresse, die so genannte **MAC** (engl. *Media access control*) Adresse. Diese ist 48 Bit lang, weltweit eindeutig und wird vom Hersteller der Netzwerkkarte fest in der Hardware gespeichert. Durch die Vergabe der Adresse vom Hersteller ergibt sich aber ein fataler Nachteil: Die **MAC**-Adressen bilden kein hierarchisches System, sondern sind mehr oder weniger zufällig verteilt. Sie können daher nicht zur Adressierung eines weit entfernten Rechners verwendet werden. Die **MAC**-Adresse spielt aber bei der Kommunikation von Rechnern in einem lokalen Netz eine entscheidende Rolle (und ist der Hauptbestandteil des Protokollkopfes von Schicht 2).

Zurück zu den IP-Adressen: Die Punkte deuten schon an, dass die IP-Adressen ein hierarchisches System bilden. Bis Mitte der 90er Jahre waren die IP-Adressen fest in Klassen eingeteilt: Je nach Klasse bildeten das erste (Klasse A: `1.x.x.x` – `127.x.x.x`), die ersten beiden (Klasse B: `128.x.x.x` – `191.255.x.x`) oder sogar die ersten drei Bytes (Klasse C: `192.0.0.x` – `223.255.255.x`) die Netzwerkadresse, die restlichen Bytes nummerierten die Rechner in diesem Netzwerk. Dieses System erwies sich aber als zu unflexibel und daher wurde diese Aufteilung aufgegeben. Man verwendet nun „klassenloses Routing“ (CIDR (engl. *classless inter domain routing*)).
Da der Rechner mit der IP-Adresse `192.168.0.20` ja erstmal nicht wissen kann, wo sich der Rechner mit der IP-Adresse `192.168.0.1` befindet, wurden die Netzmasken erdacht.

### Netzmasken und Routing

Vereinfacht gesagt definiert die (Sub-)Netzmaske auf einem Rechner mit IP-Adresse, was „drinnen“ und was „draußen“ ist. Rechner, die sich „drinnen“ (Profis sagen: „im gleichen Subnetz“) befinden, können direkt angesprochen werden. Rechner, die sich „draußen“ („nicht im gleichen Subnetz“) befinden, müssen über ein so genanntes Gateway oder Router angesprochen werden. Da jedes Netzwerkinterface eine eigene IP-Adresse bekommen kann, ahnen Sie schon, dass es schnell beliebig kompliziert wird.

Bevor ein Netzwerkpaket auf die Reise geschickt wird, läuft folgendes im Rechner ab: Die Zieladresse wird mit der Netzmaske bitweise UND verknüpft. Daraufhin wird auch die Absendeadresse bitweise mit der Netzmaske UND verknüpft. Stehen mehrere Netzwerkinterfaces zur Verfügung, werden in der Regel alle möglichen Absendeadressen überprüft.

Die Ergebnisse der UND-Verknüpfungen werden verglichen. Ergibt sich zwischen den Ergebnissen eine exakte Übereinstimmung, so befindet sich der Zielrechner im gleichen Subnetz. Ansonsten muss er über ein Gateway angesprochen werden. Das heißt, je mehr „1“ Bits sich in der Netzmaske befinden, desto weniger Rechner können direkt, sondern nur über ein Gateway angesprochen werden. Zur Veranschaulichung sind in Tabelle 2.3 mehrere Beispiele aufgeführt.

| | binäre Darstellung | | | |
|---|---|---|---|---|
| IP-Adresse:192.168.0.20 | 11000000 | 10101000 | 00000000 | 00010100 |
| Netzmaske: 255.255.255.0 | 11111111 | 11111111 | 11111111 | 00000000 |
| Ergebnis der Verknüpfung | 11000000 | 10101000 | 00000000 | 00000000 |
| In Dezimalschreibweise | 192. | 168. | 0. | 0 |
| | | | | |
| IP-Adresse: 192.168.0.22 | 11000000 | 10101000 | 00000000 | 00010110 |
| Netzmaske: 255.255.255.0 | 11111111 | 11111111 | 11111111 | 00000000 |
| Ergebnis der Verknüpfung | 11000000 | 10101000 | 00000000 | 00000000 |
| In Dezimalschreibweise | 192. | 168. | 0. | 0 |
| | | | | |
| IP-Adresse: 213.95.15.200 | 11010101 | 10111111 | 00001111 | 11001000 |
| Netzmaske: 255.255.255.0 | 11111111 | 11111111 | 11111111 | 00000000 |
| Ergebnis der Verknüpfung | 11010101 | 10111111 | 00001111 | 00000000 |
| In Dezimalschreibweise | 213. | 95. | 15. | 0 |

Tabelle 2.3: Verknüpfung der IP-Adressen mit der Netzmaske

Die Netzmaske wird wieder – wie schon die IP-Adresse – in Form von durch Punkte getrennten Dezimalzahlen geschrieben. Da die Netzmaske auch ein 32-Bit-Wert ist, werden vier Zahlenwerte nebeneinander geschrieben. Welche Rechner Gateway sind oder welche Adressbereiche über welche Netzwerkschnittstelle erreichbar sind, muss vom Benutzer konfiguriert werden.

Um wieder ein Beispiel zu geben: Alle Rechner, die am gleichen Ethernetkabel angeschlossen sind, befinden sich in der Regel *im gleichen Subnetz* und sind

direkt erreichbar. Auch wenn das Ethernet über Switches oder Bridges unterteilt ist, sind diese Rechner immer noch direkt erreichbar.

Wollen Sie eine längere Strecke überbrücken, ist das preiswerte Ethernet dafür nicht mehr geeignet. Sie müssen dann die IP-Pakete auf andere Hardware (z. B. FDDI oder ISDN) weiterleiten. Solche Geräte heißen Router bzw. Gateway. Ein Linuxrechner kann diese Aufgabe selbstverständlich auch erledigen, die entsprechende Option wird mit **`ip_forwarding`** bezeichnet.

Ist ein Gateway konfiguriert, wird das IP-Paket an das passende Gateway geschickt. Dieses versucht, das Paket dann wiederum nach dem gleichen Schema weiterzuleiten. Das wiederholt sich auf jedem weiteren Rechner sooft, bis das Paket entweder den Zielrechner erreicht hat oder die „Lebenszeit“ TTL (engl. *time to live*) des Paketes verbraucht ist.

Zur Veranschaulichung ein kleines Beispiel. Stellen Sie sich einmal ein Beispielnetzwerk vor, wie es in Abbildung 2.3 zu sehen ist:

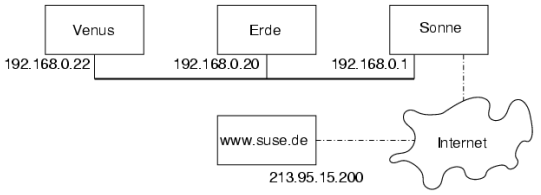



Abbildung 2.3: Unser kleines Beispielnetzwerk

Der Zielrechner `sonne` kann vom Absender `erde` direkt angesprochen werden. Ebenso können sich `venus` und `erde` direkt unterhalten. Der Zielrechner www.suse.de kann aber nicht direkt von `erde` angesprochen werden. Damit der Rechner weiß, wen er ansprechen kann, muss man auf Rechner `erde` ein Gateway angeben. Das wäre in diesem Fall der Rechner `sonne`.

Per Vereinbarung gibt es in jedem Subnetz, also jedem Teilnetz des Internets, spezielle Adressen, die in Tabelle 2.4 auf der nächsten Seite abgebildet sind:

| | |
|---|---|
| Die Netzwerkbasisadresse | Das ist die Netzmaske UND eine beliebige Adresse aus dem Netz, also das was in Tabelle 2.3 auf der vorherigen Seite unter **`Ergebnis`** abgebildet ist. Diese Adresse kann keinem Rechner zugewiesen werden. |
| Die Broadcastadresse | Sie heißt soviel wie: „Sprich alle Rechner in diesem Subnetz an“. Um sie zu erzeugen wird die Netzmaske binär invertiert und mit der Netzwerkbasisadresse ODER verknüpft. Obiges Beispiel ergibt also 192.168.0.255. Natürlich kann auch diese Adresse keinem Rechner zugewiesen werden. |

Tabelle 2.4: Fortsetzung auf der nächsten Seite...

| | |
|---|---|
| Der Localhost | Die Adresse `127.0.0.1` ist auf jedem Rechner fest dem so genannten „Loopbackdevice“ zugewiesen. Über diese Adresse kann man eine Verbindung auf den eigenen Rechner aufbauen. |

Tabelle 2.4: Spezielle Adressen

Da die IP-Adressen aber weltweit eindeutig sein müssen, können Sie natürlich nicht beliebige Adressen erfinden. Damit Sie aber trotzdem ein auf IP basierendes Netzwerk aufbauen können gibt es drei Adressbereiche, die Sie ohne weiteres verwenden können. Mit diesen können Sie allerdings nicht so ohne weiteres Verbindungen in das Internet aufbauen, da diese Adressen im Internet nicht weitergeleitet werden.

Dabei handelt es sich um diese Adressbereiche die in RFC 1597 definiert sind:

| Netzwerk, Netzmaske | Bereich |
|---|---|
| 10.0.0.0, 255.0.0.0 | 10.x.x.x |
| 172.16.0.0, 255.240.0.0 | 172.16.x.x - 172.31.x.x |
| 192.168.0.0, 255.255.0.0 | 192.168.x.x |

Tabelle 2.5: Private IP-Adressbereiche

Hinweis

Wenn Sie ein Netz aufbauen, TCP/IP verwenden möchten und keine IP-Adressen von einem Internet Provider zugewiesen bekommen haben, sollten Sie immer IP-Adressen aus einem der in Tabelle 2.5 auf der vorherigen Seite abgebildeten Bereiche wählen.

### 2.1.3 Schritt für Schritt zum eigenen Netzwerk

Wie Sie nach all der Theorie am besten vorgehen, um Ihr kleines privates Netzwerk wie in Abbildung 2.3 auf Seite 9 bei sich zu Hause einzurichten, erfahren Sie hier. Ziel ist dabei, dass Sie von allen Rechnern aus ins Internet kommen, z. B. über ISDN.

1. Richten Sie einen Internetzugang auf Sonne ein, wie in Kapitel 3.2 auf Seite 73 beschrieben. Auf allen Rechnern (Sonne, Erde und Venus) muss eine Netzwerkkarte installiert sein. Eine Verbindung unter den Rechnern mit **`ssh`** oder ähnlichem kann gut als Test verwendet werden.

2. Konfigurieren Sie zunächst den Rechner Sonne so, dass die anderen Rechner auch ins Internet kommen. Dazu ist es notwendig, das IP-Masquerading einzurichten wie in Kapitel 8.1 auf Seite 155 beschrieben. Konfigurieren Sie die Firewall gleich so, dass Sie auch mit www und anderen Diensten ins Internet kommen.

3. Erde und Venus haben jetzt schon mit einem **`ping`** Kontakt zum Internet, sofern Sonne als Default-Gateway eingetragen ist. Damit dies auch mit einem Browser funktioniert, muss noch DNS (siehe Abschnitt 2.6 auf Seite 32) und ein Proxy Server (siehe Kapitel 7 auf Seite 137) auf Sonne eingerichtet werden. Für beides reichen die Standardeinstellungen aus.

### 2.1.4 Domain Name System

#### DNS

DNS sorgt dafür, dass Sie sich nicht zwingend irgendwelche IP-Adressen merken müssen: Mit Hilfe von DNS kann eine IP-Adresse einem oder sogar mehreren Namen zugeordnet werden und umgekehrt auch eine Name einer IP-Adresse. Unter Linux erfolgt diese Umwandlung üblicherweise von einer speziellen Software namens bind. Der Rechner, der diese Umwandlung dann erledigt, nennt sich **`Nameserver`**. Dabei bilden die Namen wieder ein hierarchisches System, in dem die einzelnen Namensbestandteile durch Punkte getrennt sind. Die Namenshierarchie ist aber unabhängig von der oben beschriebenen Hierarchie der IP-Adressen.

Schauen wir uns einmal einen vollständigen Namen an:

```
   laurent.suse.de
Rechnername.Domain
```

Ein vollständiger Name – Experten sagen „fully qualified domain name“ oder kurz **FQDN** dazu – besteht aus einem Rechnernamen und einem Domainteil. Dabei wird der Domainteil aus einem frei wählbaren Anteil – im obigen Beispiel `suse` – und der so genannten **Top level domain**, **TLD** gebildet.

Aus historischen Gründen ist die Zuteilung der TLDs etwas verwirrend. So werden in den USA dreibuchstabige TLDs verwendet, woanders aber immer die aus zwei Buchstaben bestehenden ISO-Länderbezeichnungen. In der Tabelle 2.6 sind verschiedene TLDs ohne Anspruch auf Vollständigkeit aufgeführt, um Ihnen einen ersten Eindruck zu geben.

| | |
|---|---|
| `.com` | (engl. *Commercial*) - Firmen in den USA. |
| `.edu` | (engl. *Educational*) - Schulen, Universitäten und andere nichtkommerzielle Bildungseinrichtungen der USA. |
| `.gov` | (engl. *Government*) - Staatliche Einrichtungen und Regierungsstellen der USA. |
| `.org` | (engl. *Organizational*) - Nichtkommerzielle Organisationen der USA. |
| `.de` | Rechner in Deutschland. |
| `.at` | Rechner in Österreich. |

Tabelle 2.6: Verschiedene Top level domains

Wie Sie sehen, erhalten die Rechner in Deutschland üblicherweise **de**, Rechner in Österreich **at** und Rechner in der Schweiz die TLD **ch**.

In der Frühzeit des Internets (Vor 1990) gab es hierzu eine Datei `/etc/hosts`, in der die Namen aller im Internet vertretenen Rechner gespeichert waren. Dies erwies sich bei der schnell wachsenden Menge von am Internet angeschlossener Rechner als unpraktikabel. Deshalb wurde eine dezentrale Datenbank entworfen, die die Rechnernamen verteilt speichern kann. Diese Datenbank, eben jener oben erwähnte Nameserver, hält also nicht die Daten aller Rechner im Internet vorrätig, sondern kann Anfragen an ihm nachgeschaltete, andere Nameserver weiterdelegieren.

An der Spitze der Hierarchie befinden sich die „Root-Nameserver“, die die Top level domains verwalten. Die Root-Nameserver werden vom Network Information Center ( **NIC**) verwaltet. Der Root-Nameserver kennt die jeweils für eine Top level domain zuständigen Nameserver. Im Falle der deutschen Top level domain **de** ist das DE-NIC für die Domains zuständig, die mit der TLD **de** aufhören. Mehr Informationen zum DE-NIC erhalten Sie auf der Website `http://www.denic.de`, mehr Informationen zum Top level domain NIC erfahren Sie unter `http://www.internic.net`.

Damit auch Ihr Rechner einen Namen in eine IP-Adresse auflösen kann, muss ihm mindestens ein Nameserver mit einer IP-Adresse bekannt sein. Die Konfi-

guration eines Nameservers erledigen Sie komfortabel mit Hilfe von YaST bzw. YaST2 Falls Sie eine Einwahl über Modem vornehmen, kann es sein, dass das zur Einwahl verwendete Protokoll die Adresse des Nameservers während der Einwahl mitliefert.

Aber nicht nur Rechnernamen können über DNS aufgelöst werden, DNS kann noch mehr. Zum Beispiel „weiß“ der Nameserver auch, welcher Rechner für eine ganze Domain E-Mails annimmt, der so genannte **`Mail exchanger (MX)`**.

Die Konfiguration des Nameserverzugriffs unter SuSE Linux ist im Abschnitt 2.6 auf Seite 32 beschrieben.

#### whois

Eng verwandt mit DNS ist das Protokoll **`whois`**. Mit dem gleichnamigen Programm whois können Sie schnell herauskriegen, wer für eine bestimmte Domain verantwortlich ist.

## 2.2 IPv6 – Internet der nächsten Generation

### 2.2.1 Warum ein neues Internet-Protokoll?

Bedingt durch die Erfindung des WWW (engl. *World Wide Web*) ist das Internet und damit die Anzahl der Rechner, die TCP/IP „sprechen“, in den letzten zehn Jahren explosionsartig gewachsen. Seit der Erfindung des WWW durch TIM BERNERS-LEE 1990 am CERN (`http://public.web.cern.ch/`) ist die Zahl der Internet-Hosts von wenigen tausend auf mittlerweile ca. 100 Millionen gewachsen.

Wie Sie wissen, besteht eine IP-Adresse „nur“ aus 32 Bit. Viele IP-Adressen können durch organisatorische Bedingtheiten gar nicht verwendet werden, sie gehen verloren. Zur Erinnerung: Das Internet wird in Subnetze, also Teilnetze unterteilt. Diese bestehen immer aus einer Zweierpotenz minus zwei nutzbaren IP-Adressen. Ein Subnetz besteht also beispielsweise aus 2, 6, 14, 30 usw. IP-Adressen. Möchten Sie beispielsweise 128 Rechner an das Internet anbinden, so benötigen Sie ein „Class C“ Subnetz mit 256 IP-Adressen, von denen nur 254 nutzbar sind. Wie Sie oben gesehen haben, entfallen zwei der IP-Adressen aus einem Subnetz, nämlich die Broadcastadresse und die Netzwerkbasisadresse.

Die Konfiguration eines Rechners im TCP/IP-Netzwerk ist relativ kompliziert. Wie Sie oben schon gesehen haben, müssen Sie auf Ihrem Rechner folgende Dinge konfigurieren: Die eigene IP-Adresse, Subnetzmaske, Gatewayadresse (falls vorhanden) und einen Nameserver. Diese Daten müssen Sie alle „wissen“ bzw. von Ihrem Provider bekommen, sie können nicht irgendwoher abgeleitet werden. In jedem IP-Paket ist eine Prüfsumme enthalten, die bei jedem Routingvorgang überprüft und neu berechnet werden muss. Deshalb benötigen sehr schnelle Router leider sehr viel Rechenleistung, was diese Router verteuert.

Einige Dienste werden bisher mit Broadcasts realisiert (zum Beispiel das Windows Netzwerkprotokoll SMB). Rechner, die nicht an diesem Dienst interessiert

sind, sind trotzdem gezwungen, die Pakete zu verarbeiten um sie dann anschließend zu ignorieren. In sehr schnellen Netzwerken kann das durchaus ein Problem werden.

Der Nachfolger des bisherigen IP, IPv6, löst all diese Probleme. Das primäre Ziel bei der Entwicklung war, den beschränkten Adressraum stark zu erweitern und die Konfiguration von Arbeitsstationen zu vereinfachen, wenn möglich zu automatisieren. In diesem Abschnitt wird von IPv4 oder IP die Rede sein, wenn das bisher verwendete und verbreitete Internet-Protokoll gemeint ist, und von IPv6, wenn es um die neue Version 6 geht.

IPv6 ist in RFC 1752 näher erläutert. Wenn Sie sich ausführlich damit befassen wollen, können Sie in diesem RFC einen ersten Einstieg finden. IPv6 verwendet 128-Bit-Adressen, bietet also viele Billiarden IP-Adressen, genug auch bei großzügiger Verteilung der Adressen. Diese enorme Menge an IPv6-Adressen erlaubt den Luxus, das kleinste Subnetz 48 Bit „groß“ zu machen.

Dies erlaubt dann nämlich auch, die oben erläuterte MAC-Adresse als Adressbestandteil zu verwenden. Da diese weltweit nur einmal vorhanden und zugleich vom Hardwarehersteller fest vorgegeben ist, vereinfacht sich die Konfiguration der Rechner sehr. In Wirklichkeit werden sogar die ersten 64 Bit zu einem so genannten **`EUI-64`**-Token zusammengefasst. Dabei werden die letzten 48 Bit der MAC-Adresse entnommen, die restlichen 24 Bit enthalten spezielle Informationen, die etwas über den Typ des Tokens aussagen. Das ermöglicht dann auch, Geräten ohne MAC-Adresse (PPP- und ISDN-Verbindungen!) ein **`EUI-64`**-Token zuzuweisen.

Zusätzlich gibt es in IPv6 eine neue Erfindung: Einer Netzwerkschnittstelle werden üblicherweise mehrere IP-Adressen zugewiesen. Das hat den Vorteil, dass mehrere verschiedene Netze zur Verfügung stehen. Eines davon kann mit Hilfe der MAC-Adresse und einem bekannten Präfix zu einem vollautomatisch konfigurierten Netz zusammengestellt werden, und ohne weitere Konfigurationsarbeiten sind damit direkt nach dem Starten von IPv6 alle Rechner im lokalen Netz erreichbar (sogen. „Link-local-Adresse“).

Aber auch die restliche Konfiguration einer Arbeitsstation kann weitgehend automatisch erfolgen. Hierzu gibt es ein spezielles Protokoll, mit dem Arbeitsstationen von einem Router eine IP-Adresse bekommen können.

Zwingend vorgeschrieben für alle IPv6 unterstützenden Rechner ist die Unterstützung von „Multicast“. Mit Hilfe von Multicast kann eine Gruppe von Rechnern auf einmal angesprochen werden, also nicht alle auf einmal („broadcast“), oder nur einer („unicast“), sondern eben ein paar. Welche das sind, hängt von der Anwendung ab. Es gibt aber auch ein paar wohldefinierte Multicastgruppen, beispielsweise „alle Nameserver“ (engl. *all nameservers multicast group*), oder „alle Router“ (engl. *all routers multicast group*).

Da eine plötzliche Umstellung aller Rechner im Internet von IPv4 auf IPv6 nicht denkbar ist, gibt es einen Kompatibilitätsmodus. Dieser bildet die bisherigen Adressen auf IPv6-Adressen ab. Gleichzeitig gibt es Mechanismen wie „Tunneling“. Hierbei werden IPv6-Pakete in IPv4-Paketen verpackt verschickt. Natürlich sind auch Umsetzungen von IPv6 auf IPv4 und umgekehrt möglich. Um einen IPv6-Rechner von einem IPv4-Rechner aus erreichen zu können, ist es allerdings nötig, dass der IPv6-Rechner eine IPv4-Kompatibilitätsadresse hat.

### 2.2.2 Aufbau einer IPv6-Adresse

Sie können sich sicher vorstellen, dass eine IPv6-Adresse, bedingt durch die 128 Bit, wesentlich länger wird als eine IPv4-Adresse mit Ihren 32 Bit. Immerhin ist eine IPv6-Adresse damit 16 Byte lang. Verursacht durch die Größe werden die neuen IPv6-Adressen in einer anderen Schreibweise geschrieben als die bisher verwendeten IPv4-Adressen. Schauen wir uns einmal die Beispiele in Tabelle 2.7 an.

| | |
|---|---|
| Localhost | `::1` |
| IPv4 kompatible IPv6-Adresse | `::10.10.11.102`<br>(IPv6 wird unterstützt) |
| IPv4 gemappte IPv6-Adresse | `::ffff:10.10.11.102`<br>(IPv6 wird nicht unterstützt) |
| beliebige Adresse | `3ffe:400:10:100:200:c0ff:fed0:a4c3` |
| Link-local-Adresse | `fe80::10:1000:1a4` |
| Site-local-Adresse | `fec0:1:1:0:210:10ff:fe00:1a4` |
| Multicastgruppe „alle link-lokalen Router“ | `ff02:0:0:0:0:0:0:2` |

Tabelle 2.7: Darstellung verschiedener IPv6 Adressen

Wie Sie der Tabelle entnehmen, werden IPv6-Adressen mit Hilfe von Hexadezimalzahlen dargestellt. Die Hexadezimalzahlen werden immer zu je zwei Byte gruppiert zusammengefasst und durch `:` getrennt dargestellt. Es gibt daher maximal acht Gruppen und sieben Doppelpunkte in einer Adresse. Führende Null-Bytes in einer Gruppe dürfen weggelassen werden, nicht aber inmitten oder am Ende einer Gruppe. Mehr als vier Null-Bytes direkt hintereinander kann man durch das Auslassungszeichen `::` überspringen. Allerdings ist nur ein Auslassungszeichen in einer Adresse erlaubt. Dieser Vorgang des Auslassens wird in Englisch mit „collapsing“ bezeichnet. Eine Spezialdarstellung sind IPv4-Kompatibilitätsadressen: Hier wird die IPv4-Adresse einfach an den festgelegten Präfix für IPv4-Kompatibilitätsadressen angehängt.

Jeder Teil einer IPv6-Adresse hat eine definierte Bedeutung. Die ersten Bytes bilden einen Präfix und geben den Typ der Adresse an. Der Mittelteil adressiert ein Netzwerk oder ist bedeutungslos und den Schluss der Adresse bildet der Hostteil.

Tabelle 2.8 auf der nächsten Seite veranschaulicht die Bedeutung einiger häufiger Präfixe.

| Präfix (hexadezimal) | Verwendung |
|---|---|
| **00** | IPv4 und IPv4 über IPv6-Kompatibilitätsadressen. Es handelt sich um eine zu IPv4 kompatible Adresse. Ein geeigneter Router muss das IPv6-Paket noch in IPv4 verwandeln. Weitere Spezialadressen (z. B. Loopback Device) sind ebenfalls mit diesem Präfix ausgestattet. |
| erste Ziffer 2 oder 3 | (engl. *provider-based-unicast*) Provider basierte Unicast-Adressen. Wie bisher auch können Sie bei IPv6 von einem Provider Teilnetze zugewiesen bekommen. |
| **fe80** bis **febf** | (engl. *link-local*) Adressen mit diesem Präfix dürfen nicht geroutet werden und können daher nur im gleichen Subnetz erreicht werden. |
| **fec0** bis **feff** | (engl. *site-local*) Diese Adressen dürfen zwar geroutet werden, aber nur innerhalb einer Organisation. Damit entsprechen diese Adressen den bisherigen „privaten“ Netzen (beispielsweise **10.x.x.x**). |
| **ff** | (engl. *multicast*) IPv6-Adressen die mit **ff** anfangen sind Multicastadressen. |

Tabelle 2.8: verschiedene IPv6-Präfixe

Wie Sie oben schon sehen, sind speziell Unicastadressen sehr lang. Diese kann man sich praktisch nicht mehr merken. Ein funktionierender Nameserver ist für IPv6 daher noch wichtiger als bei IPv4. Der Nameserver ist so wichtig, dass es ein spezielles Autokonfigurationsprotokoll für Nameserver gibt.

### 2.2.3 IPv6-Netzmasken

Netzmasken werden in IPv6 etwas anders dargestellt. Da schon von Anfang an klassenloses Routing verwendet wird und schon das kleine Subnetz praktisch beliebig viele Rechner aufnehmen kann, macht die Unterteilung der Netze in Klassen keinen Sinn. Da die Netzmasken in der Darstellung sehr lang wären, werden diese nun ganz anders geschrieben. Die Schreibweise

```
fec0:1:1:0:210:10ff:fe00:1a4/64
```

bedeutet, dass die letzten 64 Bit den Hostteil und die vorderen 64 Bit den Netzwerkteil der Adresse bilden.

Anders gesagt bedutet die **64**, dass von links her die Netzmaske mit 1 Bits aufgefüllt wird. Es gibt in der Netzmaske also 64 1 Bits. Wie bei IPv4 wird durch eine UND-Verknüpfung der Netzmaske mit der IP-Adresse bestimmt, ob sich ein Rechner im gleichen oder in einem anderen Subnetz befindet.

### 2.2.4 Weiterführende Literatur und Links zu IPv6

Natürlich kann und will der obige Überblick keine vollständige Einführung zum sehr umfangreichen Thema IPv6 sein. Zum tieferen Einstieg in IPv6 können Sie die in Tabelle 2.9 aufgeführte Onlineliteratur und Bücher zu Rate ziehen.

| | |
|---|---|
| `http://www.bieringer.de/linux/IPv6/` | Linux-IPv6-HOWTO und viele Links. |
| `http://www.6bone.de/` | Anschluss an das IPv6 über einen Tunnel bekommen. |
| `http://www.ipv6.org/` | Alles rund um IPv6. |
| RFC 1725 | Der Einführende RFC zum Thema IPv6. |

Tabelle 2.9: verschiedene Informationen zu IPv6

## 2.3 Die Einbindung ins Netzwerk

TCP/IP ist inzwischen das Standard-Netzwerkprotokoll, über das alle modernen Betriebssysteme mit TCP/IP kommunizieren können. Dennoch unterstützt Linux auch noch andere Netzwerkprotokolle, beispielsweise das (früher) von Novell Netware verwendete IPX oder das von Macintosh-Rechnern verwendete Appletalk. In diesem Rahmen besprechen wir nur die Integration eines Linux-Rechners in ein TCP/IP-Netzwerk. Hierzu wird im folgenden vorgestellt, wie Sie den Rechner mit einer Ethernetkarte in ein LAN (engl. *local area network*) einbinden. Wenn Sie „exotische“ Arcnet, Token-Ring oder FDDI-Netzwerkkarten einbinden wollen, finden Sie weiterführende Hilfe hierzu in den Kernelquellen `/usr/src/linux/Documentation`. Ein weiterer Anlaufpunkt ist in diesem Fall aber auch ein HOWTO. Eine sehr gute und ausführliche Einführung zum Netzwerkbetrieb unter Linux finden Sie in deutscher Sprache in der Datei `/usr/share/doc/howto/de/DE-NET3-HOWTO.txt.gz`. Hier finden Sie noch einmal alles Wissenswerte, auch zu ausgefallenen Protokollen.

Die folgende Voraussetzung muss erfüllt sein:

Der Rechner muss über eine unterstützte Netzwerkkarte verfügen. Üblicherweise wird die Netzwerkkarte schon bei der Installation erkannt und der passende Treiber eingebunden. Ob Ihre Karte korrekt eingebunden wurde, können Sie unter anderem daran sehen, dass die Ausgabe des Kommandos

```
erde:~ # /sbin/ifconfig eth0
```

eine Zeile enthält, die mit **`eth0`** beginnt.

> Tipp
>
> Wenn der Kernel-Support für die Netzwerkkarte als Modul realisiert wird – so wie es beim SuSE-Kernel standardmäßig der Fall ist –, dann muss der Name des Moduls als Alias in der `/etc/modules.conf` eingetragen werden. Für die erste Ethernet-Karte z. B. in dieser Art:
>
> ```
> alias eth0 tulip
> ```
>
> Dies geschieht automatisch, wenn im linuxrc während der Erstinstallation der Treiber-Support für die Netzwerkkarte geladen wird. Nachträglich lässt sich diese Aufgabe von YaST aus erledigen.

### 2.3.1 Konfiguration mit Hilfe von YaST2

Nun können Sie die Konfiguration des Netzwerks mit YaST2 durchführen. Wählen Sie hierzu den Punkt ‘`Konfiguration der Netzwerkkarte`’, um den Dialog zur grundlegenden Konfiguration der Netzwerkkarte und des Netzwerks aufzurufen. Es erscheint der in Abbildung 2.4 auf der nächsten Seite abgebildete Dialog. Mit ‘`Hinzufügen`’ können Sie Netzwerkkarten zur Konfiguration hinzufügen, mit ‘`Entfernen`’ aus der Konfiguration entfernen und mit ‘`Bearbeiten`’ können Sie die Einstellungen zu einer Netzwerkkarte bearbeiten.

Aktivieren Sie den Punkt ‘`Hardware`’, um die Hardwaredaten einer schon eingerichteten Netzwerkkarte mit ‘`Bearbeiten`’ zu verändern. Sie gelangen in das

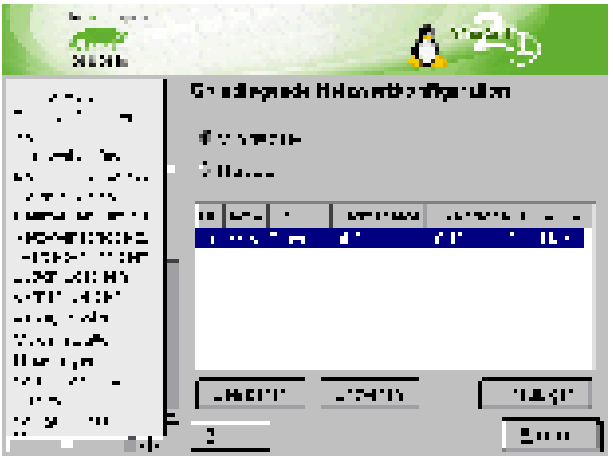


Abbildung 2.4: Dialog zur Netzwerkgrundkonfiguration

Menü zur Konfiguration der Hardwaredaten Ihrer Netzwerkkarte. Das Menü ist in Abbildung 2.5 abgebildet.

Üblicherweise wird der richtige Treiber für Ihre Netzwerkkarte schon während der Installation von YaST2 konfiguriert und die Netzwerkkarte aktiviert. Daher sind manuelle Einstellungen der Hardwareparameter nur nötig, wenn Sie mehr als eine Netzwerkkarte einsetzen oder die Netzwerkhardware nicht automatisch erkannt wird. In diesem Fall müssen Sie den Punkt ‘Neu’ anwählen, damit ein neues Treibermodul ausgewählt werden kann.

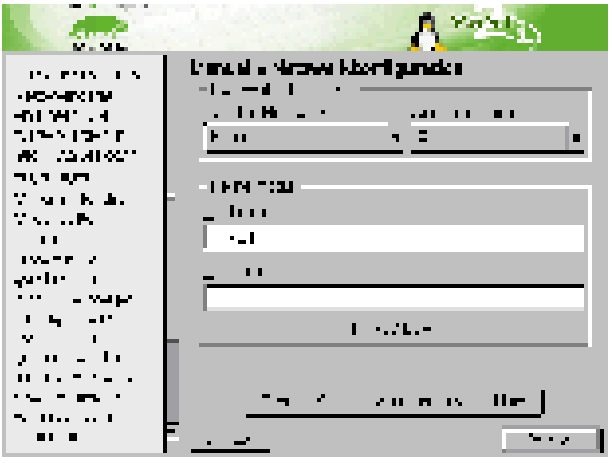


Abbildung 2.5: Konfiguration der Hardwareparameter

In diesem Dialog können Sie den Typ der Netzwerkkarte und im Falle von ISA-Karten auch den zu verwendenden Interrupt und die IO-Adresse einstellen. Manchen Netzwerktreibern können Sie auch spezielle Parameter wie die

zu verwendende Schnittstelle mitgeben, ob Sie beispielsweise den **RJ-45**- oder **BNC**-Anschluss auf der Karte verwenden wollen. Beachten Sie hierzu die Dokumentation des Treibermoduls.

Nach der Eingabe der Hardwareparameter konfigurieren Sie die weiteren Daten der Netzwerkschnittstelle. Wählen Sie im Dialog ‘Grundlegende Netzwerkkonfiguration’ den Punkt ‘Schnittstelle’ aus, um die soeben einrichtete Netzwerkkarte zu aktivieren und dieser Netzwerkkarte eine IP-Adresse zuzuweisen. Wählen Sie dann die Kartennummer aus und klicken Sie auf ‘Bearbeiten’. Es erscheint ein neuer Dialog, in dem Sie die IP-Adresse und die weiteren Daten des IP-Netzwerks auswählen können. Beachten Sie hierzu auch die Abbildung 2.6. Falls Sie selber ein eigenes Netzwerk aufbauen, können Sie sich bei der Vergabe der Adressen am Abschnitt 2.1 auf Seite 3 bzw. der Tabelle 2.5 auf Seite 10 orientieren. Ansonsten tragen Sie bitte die von Ihrem Netzwerkadministrator zugewiesenen Adressen in die vorgesehenen Felder ein.


Abbildung 2.6: Konfiguration der Netzwerkadressen

Vergessen Sie nicht, einen Nameserver unter ‘Rechnername und Nameserver’ einzustellen, damit die Namensauflösung wie in Abschnitt 2.6 auf Seite 32 beschrieben funktionieren kann.

Über den Punkt ‘Routing’ können Sie das Routing einstellen. Wählen Sie den Punkt ‘Konfiguration für Experten’, um fortgeschrittene Einstellungen vorzunehmen.

### 2.3.2 Konfiguration mit Hilfe von YaST

Sie können die Konfiguration der Netzwerksoftware auch mit YaST durchführen.

1. Loggen Sie sich als Benutzer ‘root’ ein.

2. Starten Sie YaST und wechseln Sie in das Menü ‘`Administration des Systems`’, ‘`Netzwerk konfigurieren`’, ‘`Netzwerk-Grundkonfiguration`’.

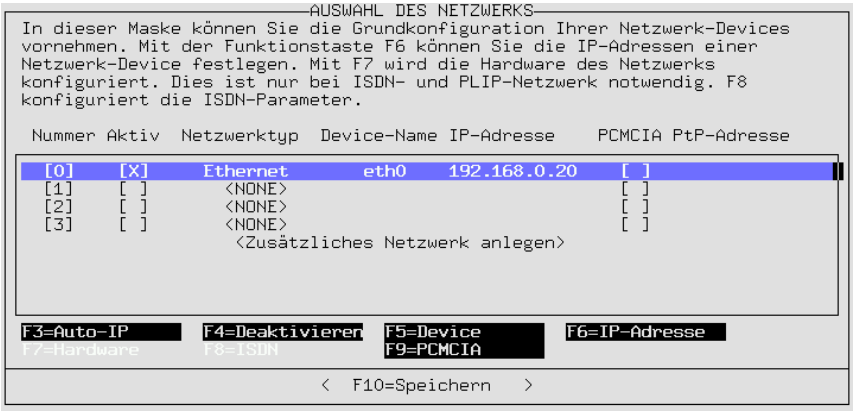



Abbildung 2.7: Netzwerkkonfiguration mit YaST

3. Wählen Sie eine freie ‘`Nummer`’, z. B. `0`.

4. Wählen Sie durch Drücken von F5 als Device ‘`Ethernet`’ aus, und verlassen Sie die Eingabemaske durch Betätigen des Buttons ‘`Weiter`’.

5. Drücken Sie F6 (‘`IP-Adresse`’), und geben Sie die IP-Adresse des Rechners ein, also beispielsweise `192.168.0.20`. Als nächstes muss die Netzwerkmaske angegeben werden. Für ein Class-C-Netz (bis zu 254 Rechner in einem Subnetz), ist diese typischerweise `255.255.255.0`. Ist kein Gatewayrechner im Netzwerk, sollte hier nichts angegeben werden (vgl. Abbildung 2.7).

6. Verlassen Sie die Eingabemaske durch Betätigen des Buttons ‘`Weiter`’.

7. Aktivieren Sie das Netzwerk-Device mit F4.

8. Durch Drücken von F10 können Sie die Konfiguration speichern, mit Esc können Sie die Maske verlassen, ohne dass die Änderungen gesichert werden.

9. Im Menü ‘`Rechnernamen ändern`’ können Sie dem Rechner einen Namen geben oder einen bestehenden Rechnernamen ändern. In die Eingabemaske des Menüs wird auch der Name der Domain eingetragen, der der Rechner angehören soll.

10. Unter dem Punkt ‘`Netzwerkdienste konfigurieren`’ können Sie festlegen, ob der inetd, das Programm portmap und der NFS-Server gestartet werden sollen und welcher Rechner- und Domainname beim Posten von Artikeln im USENET in die From-Zeile eintragen wird.

- Der inetd wird benötigt, um bei Bedarf bestimmte Netzwerkdienste (z. B. telnet, finger, ftp usw.) zu starten. Der inetd sollte beim Hochfahren des Systems immer gestartet werden, da andernfalls eine Vielzahl von Diensten auf dem System nicht zur Verfügung stehen. Beherzigen Sie bei gefährdeten Systemen die Richtlinien zur Sicherheit in Abschnitt 8.3 auf Seite 168.
- Wenn der Rechner als NFS-Server eingesetzt oder NIS verwendet werden soll, muss der Portmapper portmap beim Hochfahren des Systems gestartet werden. Haben Sie sich dafür entschieden, den Portmapper zu starten, werden Sie anschließend gefragt, ob auch der NFS-Server gestartet werden soll.

11. Im Menü '`Konfiguration Nameserver`' kann der Zugriff auf einen oder mehrere Nameserver konfiguriert werden. Es können bis zu drei IP-Adressen durch Leerzeichen getrennt angegeben werden.
12. Über den Menüpunkt '`Sendmail konfigurieren`' kann eine grundlegende Konfiguration des Pakets sendmail vorgenommen werden. Eine ausführlichere Beschreibung der Konfiguration von Sendmail finden Sie im Abschnitt 3.7 auf Seite 104.

Darüber hinaus können Sie eine ganze Reihe weiterer Einstellungen direkt in der zentralen Konfigurationsdatei `/etc/rc.config` vornehmen. Auch hier unterstützt Sie YaST.

Damit ist die Netzwerkkonfiguration abgeschlossen. YaST ruft abschließend SuSEconfig auf und lässt die gemachten Angaben in die entsprechenden Dateien eintragen (siehe Abschnitt 2.4 auf der nächsten Seite). Damit die Einstellungen wirksam werden, müssen die betroffenen Programme neu konfiguriert und die benötigten Daemonen neu gestartet werden. Dies erreichen Sie, indem Sie den Befehl

```
erde:~ # rcnetwork restart
```

eingeben.

### 2.3.3 Konfiguration von IPv6 mit YaST und YaST2

Falls Sie die Verwendung von IPv6 konfigurieren möchten, müssen Sie in der Regel keine Konfiguration auf den Arbeitsstationen durchführen. Allerdings muss die IPv6-Unterstützung geladen werden. Dies können Sie am einfachsten mit dem Kommando

```
erde:~ # modprobe ipv6
```

erledigen. Aufgrund der Autokonfigurationsphilosophie von IPv6 wird dann der Netzwerkkarte eine Adresse im **`link-local`** Netz zugewiesen. Diese wird dann auch nach der Eingabe des Kommandos ifconfig mit aufgeführt.

Normalerweise wird auf einer Arbeitsstation keine Routingtabelle gepflegt. Die Router im Netz können über das **`Router advertisment protocol`** von der Arbeitsstation darüber befragt werden, welches Präfix und welche Gateways zu verwenden sind. Um einen IPv6-Router aufzusetzen, können Sie das Programm

radvd aus Paket `radvd`, Serie `n` (Netzwerk) verwenden. Dieses Programm teilt den Arbeitsstationen das zu verwendende Präfix für IPv6-Adressen und den/die Router mit.

Um einer Arbeitsstation eine IPv6-Adresse bequem zuweisen zu können, ist es also ratsam, einen Router mit dem Programm radvd zu installieren und zu konfigurieren. Die Arbeitsstationen bekommen die IPv6-Adresse dann automatisch zugewiesen.

## 2.4 Manuelle Netzwerkkonfiguration

Die manuelle Konfiguration der Netzwerksoftware sollte stets die zweite Wahl sein. Wir empfehlen, YaST zu benutzen; allerdings kann YaST nicht alle Bereiche der Netzwerkkonfiguration abdecken, so dass in manchen Fällen manuelle Nacharbeit nötig sein wird.

### 2.4.1 Konfigurationsdateien

Dieser Abschnitt gibt eine Übersicht über die Netzwerkkonfigurationsdateien und erklärt ihre Funktion sowie das verwendete Format.

**`/etc/rc.config`**
: In dieser zentralen Konfigurationsdatei wird der größte Teil der Netzwerkkonfiguration vorgenommen. Bei Veränderung mittels YaST oder durch den Aufruf von SuSEconfig, nachdem die Datei manuell verändert wurde, werden aus diesen Einträgen die meisten der folgenden Dateien automatisch generiert. Auch die Bootskripten werden über die Einstellungen in dieser Datei konfiguriert.

> **Tipp**
> Wenn Sie diese Datei von Hand verändern, müssen Sie nachfolgend immer SuSEconfig aufrufen, damit die geänderte Konfiguration automatisch in die richtigen Dateien eingetragen wird.

**`/etc/hosts`**
: In dieser Datei (siehe Datei 2.4.1 auf der nächsten Seite) werden Rechnernamen IP-Adressen zugeordnet. Wird kein Nameserver verwendet, müssen hier alle Rechner aufgeführt werden, zu denen eine IP-Verbindung aufgebaut werden soll. Je Rechner wird eine Zeile bestehend aus IP-Adresse, dem voll qualifizierten Hostnamen und dem Rechnernamen (z. B. `erde`) in die Datei eingetragen. Die IP-Adresse muss am Anfang der Zeile stehen, die Einträge werden durch Leerzeichen bzw. Tabulatoren getrennt. Kommentare werden durch `'#'` eingeleitet.

**`/etc/networks`**
: Hier werden Netzwerknamen in Netzwerkadressen umgesetzt. Das Format

```
#
# 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.  Just add the names, addresses
#               and any aliases to this file...
#
127.0.0.1 localhost
192.168.0.1 sonne.kosmos.all sonne
192.168.0.20 erde.kosmos.all erde
# End of hosts
```

Datei 2.4.1: `/etc/hosts`

ähnelt dem der `hosts`-Datei, jedoch stehen hier die Netzwerknamen vor den Adressen (siehe Datei 2.4.2).

```
#
# networks   This file describes a number of netname-to-address
#            mappings for the TCP/IP subsystem.  It is mostly
#            used at boot time, when no name servers are running.
#
loopback     127.0.0.0
localnet     192.168.0.0
# End of networks.
```

Datei 2.4.2: `/etc/networks`

**`/etc/host.conf`**

Das Auflösen von Namen – d. h. das Übersetzen von Rechner- bzw. Netzwerknamen über die *resolver*-Bibliothek – wird durch diese Datei gesteuert. Diese Datei wird nur für Programme verwendet, die gegen die libc4 oder die libc5 gelinkt sind; für aktuelle glibc-Programme vgl. die Einstellungen in `/etc/nsswitch.conf`! Ein Parameter muss in einer eigenen Zeile stehen, Kommentare werden durch `'#'` eingeleitet. Die möglichen Parameter zeigt Tabelle 2.10 auf der nächsten Seite.

| | |
|---|---|
| order *hosts*, *bind* | Legt fest, in welcher Reihenfolge die Dienste zum Auflösen eines Namens angesprochen werden sollen. Mögliche Argumente sind (durch Leerzeichen oder Kommata voneinander getrennt):<br>*hosts*: Durchsuchen der Datei `/etc/hosts`<br>*bind*: Ansprechen eines Nameservers<br>*nis*: Über NIS |
| multi *on*/*off* | Bestimmt, ob ein in `/etc/hosts` eingetragener Rechner mehrere IP-Adressen haben darf. |

Tabelle 2.10: Fortsetzung auf der nächsten Seite...

| | |
|---|---|
| nospoof *on*<br>alert *on*/*off* | Diese Parameter beeinflussen das *spoofing* des Nameservers, haben aber weiter keinen Einfluss auf die Netzwerkkonfiguration. |
| trim <domainname> | Der angegebene Domainname wird vor dem Auflösen des Rechnernamens von diesem abgeschnitten (insofern der Rechnername diesen Domainnamen enthält). Diese Option ist dann von Nutzen, wenn in der Datei `/etc/hosts` nur Namen aus der lokalen Domain stehen, diese aber auch mit angehängtem Domainnamen erkannt werden sollen. |

Tabelle 2.10: Parameter für `/etc/host.conf`

Ein Beispiel für `/etc/host.conf` zeigt Datei 2.4.3.

```
#
# /etc/host.conf
#
# We have named running
order hosts bind
# Allow multiple addrs
multi on
# End of host.conf
```

Datei 2.4.3: `/etc/host.conf`

**`/etc/nsswitch.conf`**

Mit der GNU C Library 2.0 hat der „Name Service Switch“ (NSS) Einzug gehalten (vgl. Manual-Page von **`nsswitch.conf`** (**`man 5 nsswitch.conf`**), sowie ausführlicher *The GNU C Library Reference Manual*, Kap. "System Databases and Name Service Switch"; vgl. Paket `libcinfo`, Serie `doc`).

In der Datei `/etc/nsswitch.conf` wird festgelegt, in welcher Reihenfolge bestimmte Informationen abgefragt werden. Ein Beispiel für `nsswitch.conf` zeigt Datei 2.4.4 auf Seite 27. Kommentare werden durch `'#'` eingeleitet. Dort bedeutet z. B. der Eintrag bei der „Datenbank“ `hosts`, dass nach `/etc/hosts` (`files`) eine Anfrage über DNS (vgl. Abschnitt 2.6 auf Seite 32) losgeschickt wird.

Die über NSS verfügbaren „Datenbanken“ sind in Tabelle 2.11 auf der nächsten Seite genannt. Zusätzlich sind in Zukunft `automount`, `bootparams`, `netmasks` und `publickey` zu erwarten.

| | |
|---|---|
| `aliases` | Mail-Aliase, von `sendmail`(8) verwendet; vgl. Manual-Page von **`aliases`** (**`man 5 aliases`**). |

Tabelle 2.11: Fortsetzung auf der nächsten Seite...

| | |
|---|---|
| `ethers` | Ethernet-Adressen. |
| `group` | Für Benutzergruppen, von `getgrent`(3) verwendet; vgl. Manual-Page von **`group`** (**`man 5 group`**). |
| `hosts` | Für Hostnamen und IP-Adressen, von `gethostbyname`(3) und ähnlichen Funktionen verwendet. |
| `netgroup` | Im Netzwerk gültige Liste von Hosts und Benutzern, um Zugriffsrechte zu steuern; vgl. Manual-Page von **`netgroup`** (**`man 5 netgroup`**). |
| `networks` | Netzwerknamen und -adressen, von `getnetent`(3) verwendet. |
| `passwd` | Benutzerpasswörter, von `getpwent`(3) verwendet; vgl. Manual-Page von **`passwd`** (**`man 5 passwd`**). |
| `protocols` | Netzwerk-Protokolle, von `getprotoent`(3) verwendet; vgl. Manual-Page von **`protocols`** (**`man 5 protocols`**). |
| `rpc` | „Remote Procedure Call“-Namen und -Adressen, von `getrpcbyname`(3) und ähnlichen Funktionen verwendet. |
| `services` | Netzwerkdienste, von `getservent`(3) verwendet. |
| `shadow` | „Shadow“-Passwörter der Benutzer, von `getspnam`(3) verwendet; vgl. Manual-Page von **`shadow`** (**`man 5 shadow`**). |

Tabelle 2.11: Über `/etc/nsswitch.conf` verfügbare „Datenbanken“

```
#
# /etc/nsswitch.conf
#
passwd:     compat
group:      compat

hosts:      files dns
networks:   files dns

services:   db files
protocols:  db files

netgroup:   files
```

Datei 2.4.4: `/etc/nsswitch.conf`

Die Konfigurationsmöglichkeiten der NSS-„Datenbanken“ stehen in Tabelle 2.12.

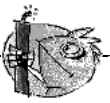
| | |
|---|---|
| `files` | direkt auf Dateien zugreifen, z. B. auf `/etc/aliases`. |
| `db` | über eine Datenbank zugreifen. |
| `nis` | NIS, vgl. Abschnitt 2.7 auf Seite 44. |
| `nisplus` | |
| `dns` | Nur bei `hosts` und `networks` als Erweiterung verwendbar. |
| `compat` | Nur bei `passwd`, `shadow` und `group` als Erweiterung verwendbar. |
| *zusätzlich* | ist es möglich, unterschiedliche Reaktionen bei bestimmten Lookup-Ergebnissen auszulösen; Details sind der Manual-Page von **`nsswitch.conf`** (**`man 5 nsswitch.conf`**) zu entnehmen. |

Tabelle 2.12: Konfigurationsmöglichkeiten der NSS-„Datenbanken“

**`/etc/nscd.conf`**

Über diese Datei wird der nscd (engl. *Name Service Cache Daemon*) konfiguriert (vgl. Manual-Page von **`nscd`** (**`man 8 nscd`**) und Manual-Page von **`nscd.conf`** (**`man 5 nscd.conf`**)). Betroffen sind die Informationen von `passwd`, `groups` und `hosts`. Der Daemon muss neu gestartet werden, wenn z. B. die Namensauflösung (DNS) durch Änderung der `/etc/resolv.conf` umgestellt wird. Dazu dient dieser Befehl:

```
erde: # rcnscd restart
```



**Achtung**

Wenn beispielsweise das Caching für `passwd` aktiviert ist, dauert es in der Regel 15 Sekunden, bis ein neu angelegter lokaler Benutzer dem System bekannt ist. Durch das Neustarten des nscd kann diese Wartezeit verkürzt werden.

**`/etc/resolv.conf`**

Wie bereits die Datei `/etc/host.conf`, so spielt auch diese Datei in Bezug auf Auflösung von Rechnernamen durch die *resolver*-Bibliothek eine Rolle.

In dieser Datei wird angegeben, welcher Domain der Rechner angehört (Schlüsselwort **`search`**) und wie die Adresse des Nameservers ist (Schlüsselwort **`nameserver`**), der angesprochen werden soll. Es können mehrere Domainnamen angegeben werden. Beim Auflösen eines nicht voll qualifizierten Namens wird versucht, durch Anhängen der einzelnen Einträge in **`search`** einen gültigen, voll qualifizierten Namen zu erzeugen. Mehrere Nameserver können durch mehrere Zeilen, die mit **`nameserver`** beginnen, bekannt gemacht werden. Kommentare werden wieder mit `'#'` eingeleitet.

Ein Beispiel für `/etc/resolv.conf` zeigt Datei 2.4.5.

```
# /etc/resolv.conf
#
# Our domain
search kosmos.all
#
# We use sonne (192.168.0.1) as nameserver
nameserver 192.168.0.1
# End of resolv.conf
```

Datei 2.4.5: `/etc/resolv.conf`

YaST (siehe Abschnitt 2.3.2 auf Seite 20) trägt hier den angegebenen Nameserver ein!

Einige Dienste wie pppd (wvdial), ipppd (isdn), dhcp (dhcpcd und dhclient), pcmcia und hotplug modifizieren die Datei `/etc/resolv.conf` über das Skript `modify_resolvconf`.

Wenn die Datei `/etc/resolv.conf` durch dieses Skript vorübergehend modifiziert wurde, enthält sie einen definierten Kommentar, der Auskunft darüber gibt, welcher Dienst sie modifiziert hat, wo die ursprüngliche Datei gesichert ist und wie man die automatischen Modifikationen abstellen kann.

Wenn `/etc/resolv.conf` mehrmals modifiziert wird, wird diese Verschachtelung von Modifikationen auch dann wieder sauber abgebaut, wenn sie in einer anderen Reihenfolge zurückgenommen werden; dies kann bei isdn, pcmcia und hotplug durchaus vorkommen.

Wenn ein Dienst nicht sauber beendet wurde, kann mit Hilfe des Skripts `modify_resolvconf` der Ursprungszustand wiederhergestellt werden. Beim

Booten wird geprüft, ob eine modifizierte `resolv.conf` stehen geblieben ist (z. B. wegen Systemabsturz). Dann wird die ursprüngliche (unmodifizierte) `resolv.conf` wiederhergestellt.

YaST findet mittels `modify_resolvconf check` heraus, ob `resolv.conf` modifiziert wurde, und dann den Benutzer warnen, dass seine Änderungen nach der Restauration wieder verloren sein werden. Ansonsten verwendet YaST `modify_resolvconf` nicht, das heißt eine Änderung der Datei resolv.conf mittels YaST und eine manuelle Änderung sind äquivalent. Beides entspricht einer gezielten und dauerhaften Änderung, während eine Änderung durch einen der genannten Dienste nur vorübergehend ist.

**`/etc/HOSTNAME`**
Hier steht der Name des Rechners, also nur der Hostname ohne den Domainnamen. Diese Datei wird von verschiedenen Skripten während des Starts des Rechners gelesen. Sie darf nur eine Zeile enthalten, in der der Rechnername steht! Auch diese Datei wird automatisch aus den Einstellungen in `/etc/rc.config` generiert.

### 2.4.2 Startup-Skripten

Neben den beschriebenen Konfigurationsdateien gibt es noch verschiedene Skripten, die während des Hochfahrens des Rechners die Netzwerkprogramme starten. Diese werden gestartet, sobald das System in einen der *Multiuser-Runlevel* übergeht (vgl. Tabelle 2.13 auf der nächsten Seite).

| | |
|---|---|
| `/etc/init.d/network` | Dieses Skript übernimmt die Konfiguration der Netzwerk Hard- und Software während der Startphase des Systems. Dabei werden auch die durch YaST (siehe Abschnitt 2.3.2 auf Seite 20) in `/etc/rc.config` eingetragenen Angaben zu IP- und Netzwerk-Adresse, Netzmaske und Gateway ausgewertet. |
| `/etc/init.d/route` | Dient dem Setzen der statischen Routen im Netzwerk. Eine detaillierte Beschreibung finden Sie in Abschnitt 2.5 auf der nächsten Seite. |
| `/etc/init.d/inetd` | Startet den inetd, sofern es in `/etc/rc.config` festgelegt ist. Dies ist beispielsweise dann nötig, wenn Sie sich vom Netzwerk aus auf diese Maschine einloggen möchten. |
| `/etc/init.d/portmap` | Startet den Portmapper, der benötigt wird, um RPC-Server verwenden zu können, wie z. B. einen NFS-Server. |
| `/etc/init.d/nfsserver` | Startet den NFS-Server. |

Tabelle 2.13: Fortsetzung auf der nächsten Seite...

| | |
|---|---|
| `/etc/init.d/sendmail` | Kontrolliert den sendmail-Prozess in Abhängigkeit von den Einstellungen in `/etc/rc.config`. |
| `/etc/init.d/ypserv` | Startet den NIS-Server in Abhängigkeit von den Einstellungen in `/etc/rc.config`. |
| `/etc/init.d/ypbind` | Startet den NIS-Client in Abhängigkeit von den Einstellungen in `/etc/rc.config`. |

Tabelle 2.13: Einige Startup-Skripten der Netzwerkprogramme

### 2.4.3 PCMCIA

Eine Sonderstellung nehmen PCMCIA-Netzwerkkarten ein. Im Gegensatz zu festeingebauten Netzwerkkarten, die eine gleich bleibende Gerätebezeichnung erhalten, beispielsweise **`eth0`**, wird PCMCIA-Karten dynamisch bei Bedarf eine freie Gerätebezeichnung zugewiesen. Um Konflikte mit eventuell fest eingebauten Karten zu vermeiden wird PCMCIA beim Booten auch erst nach dem Netzwerk gestartet.

Für PCMCIA liegen die Konfigurations- und Startskripte im Verzeichnis `/etc/pcmcia`. Diese Skripte werden ausgeführt, sobald cardmgr, der so genannte „PCMCIA Device Manager“, eine angeschlossene PCMCIA-Karte entdeckt. Deshalb ist es nicht notwendig, dass PCMCIA vor dem Netzwerk gestartet wird.

Ausführliche Informationen zu PCMCIA finden Sie im Referenz-Handbuch.

## 2.5 Routing unter SuSE Linux

### Vorbemerkung

Das Einstellen der Routing-Tabelle erfolgt unter SuSE Linux nicht über Variablen in der zentralen Konfigurationsdatei `/etc/rc.config`, sondern über das Skript `/etc/init.d/route` und die Konfigurationsdatei `/etc/route.conf`.

Nach der Initialisierung des Netzwerks durch die Boot-Skripten unter `/etc/init.d/network`, `/etc/init.d/inetd`, `/etc/init.d/i4l_hardware` und eventuell zusätzlicher Boot-Skripten, wird die Datei `/etc/route.conf` mit der Routing-Tabelle von `/etc/init.d/route` durchsucht und diese Tabelle im System gesetzt.

In der Datei `/etc/route.conf` können alle statischen Routen eingetragen werden, die für die verschiedenen Aufgaben eines Systems benötigt werden könnten: Route zu einem Rechner, Route zu einem Rechner über ein Gateway und Route zu einem Netzwerk.

Eine andere Möglichkeit ist die Benutzung des dynamischen Routings durch /usr/sbin/routed, dessen Konfiguration jedoch aufwendiger ist. Hier sei auf die Manpage von **`routed`** hingewiesen.

## Vorgehensweise und Benutzung

Die Regeln für die Konfigurationsdatei `/etc/route.conf` lehnen sich an die Ausgabe des Befehls **`/sbin/route`** an. Wird **`/sbin/route`** ohne weitere Argumente aufgerufen, erscheint die Routing-Tabelle, die der Kernel gerade benutzt. Bis auf die Spalten für die Einträge `Flags`, `Metric`, `Ref` und `Use` sind die Einträge in `/etc/route.conf` analog.

Dazu kurz die Regeln von `/etc/route.conf`:

- Zeilen mit `'#'` am Anfang und Leerzeilen werden ignoriert. Ein Eintrag besteht aus einer Zeile mit mindestens zwei und maximal vier Spalten.
- In der ersten Spalte steht das Ziel einer Route. Dabei kann dort die IP-Adresse eines Netzes oder Rechners oder bei *erreichbaren* Nameservern auch der voll qualifizierte Name eines Netzes oder eines Rechners stehen.
- Das Stichwort `default` ist dem Eintrag des Default-Gateways vorbehalten. Bitte verwenden Sie `0.0.0.0` *nicht* als Ziel für Routing-Einträge.
- Die zweite Spalte enthält entweder einen Platzhalter (`0.0.0.0`) oder die IP-Adresse bzw. den vollen Namen eines Rechners. Dieser Rechner kann das Default-Gateway sein oder ein Gateway, hinter dem ein Rechner oder Netzwerk erreichbar ist.
- Die dritte Spalte enthält die Netzmaske für Netzwerke oder Rechner hinter einem Gateway. Für Rechner hinter einem Gateway lautet die Maske z. B. `255.255.255.255`.
- Die letzte Spalte ist nur für die am lokalen Rechner angeschlossenen Netzwerke (Loopback, Ethernet, ISDN, PPP, Dummy-Device, ...) wichtig. Hier muss der Name des Devices eingetragen werden.

Ein einfaches Beispiel einer `/etc/route.conf` gibt die Datei 2.5.1. Werden neue Einträge in `/etc/route.conf` vorgenommen, wird durch die Eingabe

```
erde:~ # rcroute restart
```

die Routing-Tabelle mit den neuen Einträgen gesetzt.

```
# Destination       Dummy/Gateway       Netmask             Device
#
# 192.168.0.1       0.0.0.0             255.255.255.255     ippp0
# default           192.168.0.1
#
# Net devices
#
127.0.0.0           0.0.0.0             255.255.255.0       lo
204.127.235.0       0.0.0.0             255.255.255.0       eth0
#
# Gateway
#
default             204.127.235.41
#
# Host behind Gateway
#
207.68.156.51       207.68.145.45       255.255.255.255
#
# Net behind a Gateway
#
192.168.0.0         207.68.156.51       255.255.0.0
```

Datei 2.5.1: Einfaches Beispiel einer `/etc/route.conf`

## 2.6 DNS – Domain Name Service

DNS (engl. *Domain Name Service*) wird benötigt, um die Domain- und Rechnernamen in IP-Adressen aufzulösen. Bevor Sie einen eigenen Nameserver einrichten, sollten Sie die allgemeinen Informationen zu DNS im Abschnitt 2.1.4 auf Seite 11 lesen.

### 2.6.1 Nameserver BIND starten

Der Nameserver BIND8, das gilt auch für die neue Version BIND9, ist auf SuSE Linux bereits soweit vorkonfiguriert, dass man ihn problemlos sofort nach der Installation starten kann.

Hat man bereits eine funktionsfähige Internetverbindung und trägt in der `/etc/resolv.conf` als Nameserver 127.0.0.1 für localhost ein, hat man in der Regel schon eine funktionierende Namensauflösung, ohne dass man den DNS des Providers kennt. BIND führt so die Namensauflösung über die Root-Nameserver durch, was aber merklich langsamer ist. Normalerweise sollte man den DNS des Providers mit seiner IP-Adresse in der Konfigurationsdatei `/etc/named.conf` unter **`forwarders`** eintragen, um eine effektive und sichere Namensauflösung zu erhalten. Funktioniert das soweit, läuft der Nameserver als reiner „Caching-only“-Nameserver. Erst wenn man ihm eigene Zonen bereitstellt, wird er ein richtiger DNS werden. Ein einfaches Beispiel dafür, findet man im Doku-Verzeichnis: `/usr/share/doc/packages/bind8/sample-config`.

Man sollte allerdings keine offizielle Domain aufsetzen, solange man diese nicht von der zuständigen Institution – für ’.de’ ist das die DENIC eG – zugewiesen bekommen hat. Auch wenn man eine eigene Domain hat, diese aber vom Provider verwaltet wird, sollte man diese besser nicht verwenden, da BIND sonst keine

Anfragen für diese Domain mehr forwarden würde und so z. B. der Webserver beim Provider für die eigene Domain nicht mehr erreichbar wäre.

Um den Nameserver zu starten gibt man auf der Kommandozeile (als root)

```
rcnamed start
```

ein. Erscheint rechts in grün „done“, ist der `named`, so heißt der Nameserver-Prozess, erfolgreich gestartet. Auf dem lokalen System kann man die Funktionsfähigkeit des Nameservers sofort testen, indem man das Programm `nslookup` verwendet. Als Default Server muss localhost mit der Adresse 127.0.0.1 angezeigt werden. Sollte das nicht der Fall sein, steht wahrscheinlich in der `/etc/resolv.conf` ein falscher Nameserver oder diese Datei existiert gar nicht. Für einen ersten Test gibt man auf dem Prompt von `nslookup` „127.0.0.1“ ein, das sollte immer funktionieren; erhält man stattdessen eine Fehlermeldung „No response from server“ oder ähnlich, dann sollte man mit folgendem Kommando überprüfen, ob der `named` überhaupt läuft

```
rcnamed status
```

Falls der Nameserver nicht startet oder ein fehlerhaftes Verhalten zeigt, findet man die Ursache in den meisten Fällen in „/var/log/messages“ protokolliert.

Hat man eine Wählverbindung, muss man beachten, dass BIND8 beim Starten die Root-Nameserver überprüfen will. Gelingt ihm das nicht, weil keine Internetverbindung zustande kommt, kann das dazu führen, dass überhaupt keine DNS-Anfragen außer für lokal definierte Zonen aufgelöst werden können. BIND9 verhält sich da anders, benötigt aber ein Mehrfaches an Ressourcen im Vergleich zu BIND8.

Um den Nameserver des Providers, oder einen eigenen, den man schon im eigenen Netz laufen hat, als „forwarder“ zu verwenden, trägt man diesen oder auch mehrere, im Abschnitt **`options`** unter **`forwarders`** ein; vgl. Beispiel 2.6.1.

```
        options {
                  directory "/var/named";
                  forwarders { 10.11.12.13; 10.11.12.14; };
                  listen-on { 127.0.0.1; 192.168.0.99; };
                  allow-query { 127/8; 192.168.0/24; };
                  notify no;
                };
```

Datei 2.6.1: Forwarding-Optionen in `named.conf`

Die im Beispiel verwendeten IP-Adressen sind willkürlich gewählt und müssen entsprechend den eigenen Gegebenheiten eingetragen werden.

Nach den **`options`** folgen dann die Einträge für die Zonen, die Einträge für „localhost“, „0.0.127.in-addr.arpa“, sowie „.“ vom „type hint“ sollten mindestens immer vorhanden sein. Die zugehörigen Dateien müssen nicht verändert werden, da sie so funktionieren wie sie sind. Beachten muss man auch, dass nach jedem Eintrag ein „;“ steht und die geschweiften Klammern korrekt gesetzt sind. Hat man nun Änderungen an der Konfigurationsdatei `/etc/named.conf` oder an den Zonen-Dateien vorgenommen, muss man BIND dazu bringen diese neu einzulesen. Das gelingt mit dem Kommando **`rcnamed reload`**. Alter-

nativ kann man den Nameserver auch komplett neu starten, durch den Befehl **`rcnamed restart`**. Fehlt nur noch das Kommando, um den Nameserver wieder zu beenden: **`rcnamed stop`**. Soll der `named` bereits beim Booten gestartet werden, muss man in `/etc/rc.config` lediglich den Eintrag **`START_NAMED=no`** auf **`START_NAMED=yes`** abändern.

### 2.6.2 Die Konfigurationsdatei /etc/named.conf

Alle Einstellungen zum Nameserver BIND8 bzw. BIND9 sind in der Datei `/etc/named.conf` vorzunehmen. Die Zonendaten selbst, die Rechnernamen, IP-Adressen usw. für die zu verwaltenden Domains, sind in separaten Dateien im Verzeichnis `/var/named` abzulegen, dazu aber unten mehr.

Die `/etc/named.conf` unterteilt sich grob in zwei Bereiche, zum einen der Abschnitt **`options`** für allgemeine Einstellungen und zum anderen die **`zone`**-Einträge für die einzelnen Domains. Außerdem kann man noch einen Bereich **`logging`**, sowie Einträge vom Typ **`acl`** definieren. Kommentarzeilen beginnen mit einem `'#'`-Zeichen, alternativ ist `'//'` auch erlaubt.

Eine minimalistische `/etc/named.conf` stellt Beispiel 2.6.2 dar.

```
options {
        directory "/var/named";
        forwarders { 10.0.0.1; };
        notify no;
};

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

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

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

Datei 2.6.2: Minimalistische Datei `/etc/named.conf`

Dieses Beispiel funktioniert für Bind8 und Bind9 gleichermaßen, da keine speziellen Optionen verwendet werden, die nur von einer Version verstanden werden. Bind-9.1.1 akzeptiert alle Bind8-Konfigurationen und vermerkt allenfalls beim Start, wenn eine Option nicht implementiert ist. Spezielle Bind9-Optionen werden vom Bind8 aber nicht unterstützt.

**Die wichtigsten Konfigurationsoptionen im Abschnitt options**

**directory "/var/named";** gibt das Verzeichnis an, in dem der BIND die Dateien mit den Zonendaten findet,

**forwarders { 10.0.0.1; };** verwendet man, um den oder die Nameserver (meist des Providers) anzugeben, an den oder die DNS-Anfragen weitergereicht werden, die nicht direkt beantwortet werden können.

**forward first;** bewirkt, dass die DNS-Anfragen zu erst geforwarded werden, bevor versucht wird diese über die Root-Nameserver aufzulösen. Anstelle von **`forward first`** kann man auch **`forward only`** schreiben, dann werden alle Anfragen weitergeleitet und die Root-Nameserver werden gar nicht mehr angesprochen. Das kann für Firewall-Konfigurationen sinnvoll sein.

**listen-on port 53 { 127.0.0.1; 192.168.0.1; };** sagt dem BIND, auf welchen Netzwerkinterfaces und welchem Port er auf Anfragen der Clients horcht. Die Angabe **`port 53`** kann man sich dabei sparen, da 53 ohnehin der Standardport ist. Lässt man diesen Eintrag komplett weg, werden standardmäßig alle Interfaces verwendet

**query-source address * port 53;** Dieser Eintrag kann notwendig sein, wenn eine Firewall die externen DNS-Abfragen blockiert. So wird BIND dazu gebracht, Anfragen nach außen von Port 53 aus und nicht von den hohen Port's > 1024 zu stellen.

**allow-query { 127.0.0.1; 192.168.1/24; };** bestimmt die Netze aus denen Clients DNS-Anfragen stellen dürfen. Das **`/24`** ist dabei eine Kurzschreibweise für die Netzmaske, in diesem Fall 255.255.255.0.

**allow-transfer { ! *; };** regelt welche Rechner Zonentransfers anfordern dürfen, dieses Beispiel unterbindet sie, aufgrund des **`! *`** komplett. Ohne diesen Eintrag können Zonentransfers ohne Einschränkungen von überall angefordert werden.

**statistics-interval 0;** Ohne diesen Eintrag produziert Bind8 stündlich mehrere Zeilen Statisikmeldungen in `/var/log/messages`. Die Angabe von 0 bewirkt, dass diese komplett unterdrückt werden, ansonsten kann man hier die Zeit in Minuten angeben.

**cleaning-interval 720;** Diese Option legt fest, in welchem Zeitabstand Bind8 seinen Cache aufräumt. Die Aktivität führt jedes Mal zu einem Eintrag in /var/log/messages. Die Zeitangabe erfolgt in Minuten. Voreingestellt sind 60 Minuten.

**interface-interval 0;** Bind8 durchsucht regelmäßig die Netzwerkschnittstellen nach neuen oder nicht mehr vorhandenen Interfaces. Setzt man diesen Wert auf **`0`**, so wird darauf verzichtet und Bind8 lauscht nur auf den beim Start gefundenen Interfaces. Alternativ kann man das Intervall in Minuten angeben. Voreingestellt sind 60 Minuten.

**notify no;** Das **no** bewirkt, dass keine anderen Nameserver benachrichtigt werden, wenn an den Zonendaten Änderungen vorgenommen werden oder der Nameserver neu gestartet wird.

#### Der Konfigurationsabschnitt „Logging"

Was und wie wohin mitprotokolliert wird, kann man beim Bind8 recht vielseitig konfigurieren. Normalerweise sollte man mit den Voreinstellungen zufrieden sein können. Beispiel 2.6.3 auf der nächsten Seite zeigt die einfachste Form so eines Eintrages und unterdrückt das „Logging" komplett

#### Aufbau der Zonen-Einträge

Nach **`zone`** wird der Name der zu verwaltenden Domain angegeben, hier willkürlich **`meine-domain.de`** gefolgt von einem **`in`** und einem in geschweiften

```
        logging {

                category default { null; };

        };
```

Datei 2.6.3: Logging wird unterdrückt

```
        zone "meine-domain.de" in {
                type master;
                file "meine-domain.zone";
                notify no;
        };
```

Datei 2.6.4: Zone-Eintrag für meine-domain.de

Klammern gesetzten Block zugehöriger Optionen; vgl. Beispiel 2.6.4. Will man eine „Slave-Zone“ definieren, ändert sich nur der **`type`** auf **`slave`** und es muss ein Nameserver angegeben werden, der diese Zone als **`master`** verwaltet (kann aber auch ein „slave“ sein); vgl. Beispiel 2.6.5.

```
        zone "andere-domain.de" in {
                type slave;
                file "slave/andere-domain.zone";
                masters { 10.0.0.1; };
        };
```

Datei 2.6.5: Zone-Eintrag für andere-domain.de

Die Optionen:

**type master;** Das **`master`** legt fest, dass diese Zone auf diesem Nameserver verwaltet wird. Das setzt eine sauber erstellte Zonendatei voraus.

**type slave;** Diese Zone wird von einem anderen Nameserver transferiert. Muss zusammen mit **`masters`** verwendet werden.

**type hint;** Die Zone **`.`** vom Typ **`hint`** wird für die Angabe der Root-Nameserver verwendet. Diese Zonendefinition kann man unverändert lassen.

**file „meine-domain.zone“ oder file „slave/andere-domain.zone“;** Dieser Eintrag gibt die Datei an, in der die Zonendaten für die Domain eingetragen sind. Bei einem **`slave`** braucht die Datei nicht zu existieren, da ihr Inhalt von einem anderen Nameserver geholt wird. Um Master- und Slave-Dateien auseinander zu halten, gibt man für die Slave-Dateien das Verzeichnis `slave` an.

**masters { 10.0.0.1; };** Diesen Eintrag braucht man nur für Slave-Zonen und er gibt an, von welchem Nameserver die Zonendatei transferiert werden soll.

**allow-update { ! *; };** diese Option regelt den Schreibzugriff von extern auf die Zonendaten. Damit wäre es Clients möglich, sich selbst im DNS einzutragen, was aus Sicherheitsgründen nicht wünschenswert ist. Ohne diesen Eintag, sind Zonen-Updates generell untersagt, dieses Beispiel würde daran auch nichts ändern, da `! *` ebenfalls alles verbietet.

### Aufbau der Zonendateien

Man benötigt zwei Arten von Zonen-Dateien, die einen dienen dazu, einem Rechnernamen die IP-Adresse zu zuordnen und die anderen gehen den umgekehrten Weg und liefern zu einer gegebenen IP-Adresse den Rechnernamen.

Eine wichtige Bedeutung hat der '.' in den Zonendateien. Werden Rechnernamen, ohne abschließenden '.' angegeben, wird immer die Zone ergänzt. Man muss also komplette Rechnernamen, die bereits mit vollständiger Domain angegeben wurden, mit einem '.' abschließen, damit die Domain nicht noch einmal dran gehängt wird. Ein fehlender Punkt oder einer an der falschen Stelle, dürfte die häufigste Fehlerursache bei der Konfiguration von Nameservern sein.

Den ersten Fall betrachten wir an der Zonen-Datei `welt.zone`, die für die Domain `welt.all` zuständig ist;vgl. Datei 2.6.6.

```
1.  $TTL 2D
2.  welt.all.   IN SOA      gateway  root.welt.all. (
3.              2001040901  ; serial
4.              1D          ; refresh
5.              2H          ; retry
6.              1W          ; expiry
7.              2D )        ; minimum
8.
9.              IN NS       gateway
10.             IN MX       10 sonne
11.
12. gateway     IN A        192.168.0.1
13.             IN A        192.168.1.1
14. sonne       IN A        192.168.0.2
15. mond        IN A        192.168.0.3
16. erde        IN A        192.168.1.2
17. mars        IN A        192.168.1.3
```

Datei 2.6.6: Datei `/var/named/welt.zone`

**Zeile 1:** **`$TTL`** definiert die Standard-TTL, die für alle Einträge in dieser Datei gilt, hier 2 Tage (2D = 2 days). TTL bedeutet hier „time to live“, zu deutsch Gültigkeitsdauer.

**Zeile 2:** Hier beginnt der **`SOA control record`**:

- An erster Stelle steht hier der Name der zu verwaltenden Domain **`welt.all`**, diese ist mit einem '.' abgeschlossen, da ansonsten die Zone noch einmal angehängt würde. Alternativ kann man hier ein **`'@'`** schreiben, dann wird

die Zone dem zugehörigen Eintrag in der `/etc/named.conf` entnommen.

- Nach dem **`IN SOA`** steht der Name des Nameservers, der als Master für diese Zone zuständig ist. In diesem Fall wird der Name **`gateway`** zu **`gateway.welt.all`** ergänzt, da er nicht mit einem **`'.'`** abgeschlossen ist.
- Danach folgt eine E-Mail-Adresse, der für diesen Nameserver zuständigen Person. Da das **`'@'`**-Zeichen bereits eine besondere Bedeutung hat, ist hier stattdessen einfach ein **`'.'`** einzutragen, für **`root@welt.all`** schreibt man hier folglich **`root.welt.all.`**. Den **`'.'`** am Ende darf man hier nicht vergessen, da sonst die Zone noch angehängt würde.
- Am Ende folgt eine **`'('`**, um die folgenden Zeilen, bis zur **`')'`** mit in den SOA-Record einzuschließen.

**Zeile 3:** Die **`serial number`** ist eine willkürliche Zahl, die bei jeder Änderung an dieser Datei erhöht werden sollte. Sie wird benötigt, um sekundäre Nameserver (Slave-Server) über Änderungen zu informieren. Eingebürgert hat sich dafür eine zehnstellige Zahl aus Datum und fortlaufender Nummer in der Form JJJJMMTTNN.

**Zeile 4:** Die **`refresh rate`** gibt das Zeitintervall an, in dem Sekundär-Nameserver die **`serial number`** der Zone überprüfen. In diesem Fall 1 Tag (1D = 1 day).

**Zeile 5:** Die **`retry rate`** gibt den Zeitabstand an, in dem ein sekundärer Nameserver, im Fehlerfall versucht den primären Server erneut zu kontaktieren. Hier 2 Stunden (2H = 2 hours).

**Zeile 6:** Die **`expiration time`** gibt den Zeitraum an, nachdem ein sekundärer Nameserver die gecacheten Daten verwirft, wenn er keinen Kontakt zum primären Server mehr bekommen hat. Hier ist das eine Woche (1W = 1 week).

**Zeile 7:** Die **`minimum time to live`** sagt aus, wie lange die Ergebnisse von DNS-Anfragen von anderen Servern gecached werden dürfen, bevor sie ihre Gültigkeit verlieren und neu angefragt werden müssen.

**Zeile 9:** Das **`IN NS`** gibt den Nameserver an, der für diese Domain zuständig ist. Auch hier gilt, dass **`gateway`** wieder zu **`gateway.welt.all`** ergänzt wird, weil es nicht mit einem **`'.'`** abgeschlossen ist. Es kann mehrere Zeilen dieser Art geben, eine für den primären und jeweils eine für jeden sekundären Nameserver. Ist für diese Zone **`notify`** in der `/etc/named.conf` nicht auf **`no`** gesetzt, werden alle hier aufgeführten Nameserver über Änderungen der Zonendaten informiert.

**Zeile 10:** Der MX-Record gibt den Mailserver an, der für die Domain **`welt.all`** die Mails annimmt und weiterverarbeitet oder weiterleitet. In diesem Beispiel ist das der Rechner **`sonne.welt.all`**. Die Zahl vor dem Rechnernamen ist der Präferenz-Wert, gibt es mehrere MX-Einträge, wird zuerst der Mailserver mit dem kleinsten Wert genommen und falls die Auslieferung an diesen scheitert, wird der mit dem nächst höheren Wert versucht.

**Zeile 12-17:** Das sind jetzt die eigentlichen Adress-Records, in denen den Rechnernamen eine oder mehrere IP-Adressen zugeordnet werden. Die Namen stehen hier ohne abschließenden **.**, da sie ohne angehängte Domain eingetragen sind und alle um **welt.all** ergänzt werden dürfen. Dem Rechner **gateway** sind zwei IP-Adressen zugeordnet, da er über zwei Netzwerkkarten verfügt.

Für die Rückwärts-Auflösung (reverse lookup) von IP-Adressen in Rechnernamen wird die Pseudo-Domain **in-addr.arpa** zu Hilfe genommen. Diese wird dazu an den in umgedrehter Reihenfolge geschriebenen Netzanteil angehängt. Aus **192.168.1** wird dann **1.168.192.in-addr.arpa**; vgl. 2.6.7.

```
1. $TTL 2D
2. 1.168.192.in-addr.arpa. IN SOA gateway.welt.all. root.welt.all. (
3.                          2001040901      ; serial
4.                          1D              ; refresh
5.                          2H              ; retry
6.                          1W              ; expiry
7.                          2D )            ; minimum
8.
9.                          IN NS           gateway.welt.all.
10.
11. 1                       IN PTR          gateway.welt.all.
12. 2                       IN PTR          erde.welt.all.
13. 3                       IN PTR          mars.welt.all.
```

Datei 2.6.7: Umgekehrte Adress-Auflösung

**Zeile 1:** $TTL definiert die Standard-TTL, die hier für alle Einträge gilt.

**Zeile 2:** Der 'revers lookup' soll mit dieser Datei für das Netz 192.168.1.0 ermöglicht werden. Da die Zone hier '1.168.192.in-addr.arpa' heißt, will man dies natürlich nicht an die Rechnernamen anhängen, deshalb sind diese alle komplett mit Domain und abschließendem '.' eingetragen. Der Rest entspricht dem, was im vorangegangenen Beispiel für "welt.all", bereits beschrieben wurde.

**Zeile 3-7:** Siehe vorangegangenes Beispiel für "welt.all".

**Zeile 9:** Diese Zeile gibt auch hier wieder den Nameserver an, der für diese Zone zuständig ist, diesmal wird aber der Name komplett mit Domain und abschließendem '.' hier eingetragen.

**Zeile 11-13:** Das sind die Pointer-Records, die zu einer IP-Adresse auf den zugehörigen Rechnernamen zeigen. Hier steht am Anfang der Zeile nur die letzte Stelle der IP-Adresse, ohne abschließenden '.'. Wird jetzt die Zone daran angehängt und man denkt sich das '.in-addr.arpa' weg, hat man die komplette IP-Adresse in verdrehter Reihenfolge.

Die Zonendateien sind in dieser Form für Bind8 und Bind9 gleichermaßen verwendbar. Auch Zonentransfers zwischen den verschiedenen Versionen sollten normalerweise kein Problem darstellen.

### 2.6.3 DNS Beispielkonfiguration

In der hier vorgeführten Beispiel-Konfiguration eines Nameservers, gehen wir von folgendem Fall aus: Ihre Domain heißt welt.all, Sie haben einen Gateway-Rechner, der die Verbindung zum Internet herstellt und zwei firmeninterne Netze mit einander verbindet. Dieser Rechner wird unser Nameserver und erhält den Namen „gateway“. Ihr Netzwerk sieht wie folgt aus:

Netz 1 enthält die Rechner:

- gateway IP 192.168.1.1
- erde IP 192.168.1.2
- mars IP 192.168.1.3

Netz 2 enthält die Rechner:

- gateway IP 192.168.0.1
- sonne IP 192.168.0.2
- mond IP 192.168.0.3

Die Rechner erde und mars sind nur über gateway mit sonne und mond verbunden, ebenso können alle Rechner nur über gateway das Internet erreichen.

Für die Konfiguration des Nameservers sind folgende Dateien nötig:

**named.conf** die zentrale Konfigurationsdatei,

**welt.zone** enthält die host-Tabelle,

**192.168.1.zone** für das Subnetz mit den Rechnern erde und mars,

**192.168.0.zone** für das Subnetz mit den Rechnern sonne und mond,

**localhost.zone** enthält die IP-Adresse des localhost,

**127.0.0.zone** loopback,

**root.hint** enthält die Root-Server des Internets.

Die Dateien `localhost.zone`, `127.0.0.zone` und `root.hint` werden automatisch bei der Installation on BIND 8 angelegt.

Die Datei `/etc/named.conf` muss angepasst werden wie in Datei 2.6.8 auf der nächsten Seite.

**`acl`** ist die Zugriffskontroll-Liste, die festlegt, von welchen IP-Adressen aus auf den DNS-Server zugegriffen werden darf.

Der **`directory`**-Eintrag gibt an, wo sich die anderen Konfigurationsdateien befinden. Standard ist `/var/named/`.

**`listen-on Port`** definiert den Port, von dem der Name Server Anfragen erhält.

Die **`zone`**-Einträge geben an, in welchen Konfigurationsdateien die IP-Adressen und Rechnernamen einander zugeordnet werden. Diese Dateien müssen im nächsten Schritt im Verzeichnis `/var/named/` angelegt werden.

```
acl internal { 127.0.0.1; 192.168.1/24; 192.168.0/24; };

options {
     directory "/var/named";
     allow-query { internal; };
#    forwarders { 10.0.0.1; };
#    listen-on port 53 { 127.0.0.1; 192.168.0.1; 192.168.1.1; };
#    query-source address * port 53;
     cleaning-interval 120;
     statistics-interval 0;
     notify no;
};
   zone "welt.all" in {
     type master;
     file "welt.zone";
   };

   zone "0.168.192.in-addr.arpa" in {
     type master;
     file "192.168.0.zone";
   };

   zone "1.168.192.in-addr.arpa" in {
     type master;
     file "192.168.1.zone";
   };

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

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

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

Datei 2.6.8: Datei `named.conf`

```
$TTL 2D

welt.all. IN SOA       gateway    root.welt.all. (
                       2001040501     ; serial
                       1D             ; refresh
                       2H             ; retry
                       1W             ; expiry
                       2D )           ; minimum

          IN NS        gateway
          IN MX        10 sonne

gateway   IN A         192.168.0.1
          IN A         192.168.1.1
sonne     IN A         192.168.0.2
mond      IN A         192.168.0.3
erde      IN A         192.168.1.2
mars      IN A         192.168.1.3
```

Datei 2.6.9: Datei `/var/named/welt.zone`

In der Datei `welt.zone` (siehe Datei 2.6.9) wird die komplette host-Tabelle eingetragen. In unserem Beispiel sieht das wie folgt aus:

Die Option '`$TTL`' gibt die „Time to Live“ also die Gültigkeitsdauer an. '`SOA`' steht für „Start of Authority“ und leitet den vordefinierten Datensatz ein: Rechnername, E-Mail-Adresse, wobei das „@“-Zeichen durch einen „.“ ersetzt wird, Seriennummer (Datum und zweistellige Versionsnummer) und Gültigkeitszeiten. '`NS`' markiert den Name Server. '`A`' signalisiert, dass nun die IP-Adressen der Rechner in der Domain folgen. Der Name-Server „gateway“ hat zwei IP-Adressen, da er zwei Subnetzen angehört. In den beiden weiteren zone-Dateien erfolgt die umgekehrte Adress-Auflösung für die beiden Subnetze; vgl. Beispiel-Dateien 2.6.10 und 2.6.11 auf der nächsten Seite.

```
$TTL 2D
0.168.192.in-addr.arpa. IN SOA gateway.welt.all. root.welt.all. (
                                2001040501     ; serial
                                1D             ; refresh
                                2H             ; retry
                                1W             ; expiry
                                2D )           ; minimum

          IN NS                 gateway.welt.all.

1         IN PTR                gateway.welt.all.
2         IN PTR                sonne.welt.all.
3         IN PTR                mond.welt.all.
```

Datei 2.6.10: Die Datei 192.168.0.zone

Bevor Sie nun Ihren Name-Server testen können, muss in der Datei `/etc/rc.`

```
$TTL 2D
1.168.192.in-addr.arpa. IN SOA gateway.welt.all. root.welt.all. (
                                2001040501     ; serial
                                1D             ; refresh
                                2H             ; retry
                                1W             ; expiry
                                2D )           ; minimum

          IN NS                 gateway.welt.all.

1         IN PTR                gateway.welt.all.
2         IN PTR                erde.welt.all.
3         IN PTR                mars.welt.all.
```

Datei 2.6.11: Die Datei 192.168.1.zone

`config` der Eintrag **`START_NAMED=yes`** gesetzt werden. Anschließend kann man den anderen Rechnern die IP-Adresse des Name-Servers mitteilen, zum Beispiel über YaST1 (Administration des Systems -> Netzwerk konfigurieren -> Konfiguration Nameserver). Wenn Sie nun in einer Konsole **`nslookup erde`** eingeben, sollte Ihnen der Nameserver mit IP-Adresse sowie die IP-Adresse des Rechners erde ausgegeben werden. Sollte der Nameserver nicht funktionieren, finden Sie die Ursache in der Datei `/var/log/messages`.

### 2.6.4 Weitere Informationen

- Dokumentation zum Paket `bind8`: `file:/usr/share/doc/packages/bind8/html/index.html`.
- Eine Beispielkonfiguration findet man unter: `/usr/share/doc/packages/bind8/sample-config`
- Manual-Page von **`named`** (**`man 8 named`**), in der die einschlägigen RFCs genannt werden, sowie besonders die Manual-Page von **`named.conf`** (**`man 5 named.conf`**).

## 2.7 NIS – Network Information Service

### 2.7.1 Was ist NIS?

Sobald mehrere Unix-Systeme in einem Netzwerk auf gemeinsame Ressourcen zugreifen wollen, muss sichergestellt sein, dass Benutzer- und Gruppenkennungen auf allen Rechnern miteinander harmonieren. Das Netzwerk soll für den Anwender transparent sein. Egal welcher Rechner, der Anwender findet immer die gleiche Umgebung vor. Möglich wird dies durch die Dienste NIS und NFS. NFS dient der Verteilung von Dateisystemen im Netz und wird in Abschnitt 2.8 auf Seite 46 beschrieben.

NIS (engl. *Network Information Service*) kann als Datenbankdienst verstanden werden, der Zugriff auf Informationen aus den Dateien `/etc/passwd`, `/etc/shadow` oder `/etc/group` netzwerkweit ermöglicht. NIS kann auch für weitergehende Aufgaben eingesetzt werden (z. B. für `/etc/hosts` oder `/etc/services`). Darauf soll hier jedoch nicht im Detail eingegangen werden. Für NIS wird vielfach synonym der Begriff `'YP'` verwendet. Dieser leitet sich ab von den *yellow pages*, also den *gelben Seiten* im Netz.

### 2.7.2 Einrichten eines NIS-Clients

Im Paket `ypbind`, Serie `n`, befinden sich alle notwendigen Programme zum Einrichten eines NIS-Clients. Zur Einrichtung des NIS-Clients sind folgende Schritte zu erledigen:

- Setzen der NIS-Domain beim Starten des Systems.

  Dazu muss in `/etc/rc.config` die Variable **`YP_DOMAINNAME`** gesetzt werden. Beim Übergang in einen Runlevel, in dem das Netzwerk verwendet wird, wertet `/etc/init.d/network` diesen Wert aus und setzt den Namen entsprechend. Der NIS-Domainname ist nicht zu verwechseln mit dem DNS-Domainnamen. Diese können gleich lauten, haben jedoch grundsätzlich nichts miteinander zu tun!

- Festlegen des NIS-Servers.

  Der Name des NIS-Servers wird in der `/etc/rc.config` durch die Variable **`YP_SERVER`** gesetzt. SuSEconfig schreibt diese Namen im richtigen Format in die Datei `/etc/yp.conf` (vgl. Datei 2.7.1). Haben Sie die Variable mit YaST gesetzt, dann geschieht dies automatisch. In dieser Datei muss es eine Zeile geben, die mit dem Schlüsselwort **`ypserver`** beginnt und in der der Name des NIS-Servers steht.

```
# /etc/yp.conf
#
# Syntax:
#
# ypserver   <servername>      Define which host to contact
#                              for YP service.
#
ypserver     sonne.kosmos.all
# End of /etc/yp.conf
```

Datei 2.7.1: `/etc/yp.conf`

- Der Name des NIS-Servers (z. B. `sonne.kosmos.all`) muss über `/etc/hosts` auflösbar sein.

- Es muss sichergestellt sein, dass der RPC-Portmapper gestartet wird.

  NIS wird über RPC (engl. *Remote Procedure Calls*) realisiert, deshalb ist es Bedingung, dass der RPC-Portmapper läuft. Gestartet wird dieser Server

vom Skript `/etc/init.d/portmap`. Auch dies wird automatisch erledigt, wenn das Starten des Portmappers in `/etc/rc.config` veranlasst wurde.

- Ergänzen der Einträge in `/etc/passwd` und `/etc/group`.

  Damit nach dem Durchsuchen der lokalen Dateien eine Anfrage beim NIS-Server gemacht wird, müssen die entsprechenden Dateien durch eine Zeile, die mit einem Pluszeichen (`'+'`) beginnt, ergänzt werden. NIS erlaubt es, hier eine Menge weiterer Optionen zu aktivieren, z. B. Netgroups oder lokales Überschreiben von NIS-Einträgen.

- Starten von ypbind.

  Der letzte Schritt des Aufsetzens des NIS-Clients besteht aus dem Start des Programmes ypbind, das den eigentlichen Start des NIS-Clients bedeutet. Auch dieses Programm wird automatisch gestartet, wenn Sie die Konfiguration des Netzwerks mit YaST vorgenommen haben.

- Aktivieren der Änderungen.

  Entweder muss nun das System neu gestartet werden oder die benötigten Dienste werden durch

  ```
  erde:~ # rcnetwork restart
  erde:~ # rcypbind restart
  ```

  neu gestartet.

### 2.7.3 NIS-Master- und -Slave-Server

Zu installieren ist das Paket `ypserv`, Serie n. Das genaue Vorgehen ist in `/usr/share/doc/packages/ypbind/HOWTO.SuSE` beschrieben.

## 2.8 NFS – verteilte Dateisysteme

Wie bereits in Abschnitt 2.7 auf Seite 44 erwähnt, dient NFS neben NIS dazu, ein Netzwerk für Anwender transparent zu machen. Durch NFS lassen sich Dateisysteme im Netz verteilen. Unabhängig davon, an welchem Rechner im Netz ein Anwender arbeitet, findet er so stets die gleiche Umgebung vor.

Wie NIS ist auch NFS ein asymmetrischer Dienst. Es gibt NFS-Server und NFS-Clients. Allerdings kann ein Rechner beides sein, d. h. gleichzeitig Dateisysteme dem Netz zur Verfügung stellen („exportieren“) und Dateisysteme anderer Rechner mounten („importieren“). Im Regelfall jedoch benutzt man dafür Server mit großer Festplattenkapazität, deren Dateisysteme von Clients gemountet werden.

### 2.8.1 Importieren von Dateisystemen

Dateisysteme von einem NFS-Server zu importieren, ist sehr einfach. Einzige Voraussetzung ist, dass der RPC-Portmapper gestartet wurde. Das Starten dieses Servers wurde bereits im Zusammenhang mit NIS besprochen (siehe Ab-

schnitt 2.7.2 auf Seite 45). Ist diese Voraussetzung erfüllt, können fremde Dateisysteme, sofern sie von den entsprechenden Maschinen exportiert werden, analog zu lokalen Platten mit dem Befehl **mount** in das Dateisystem eingebunden werden. Die Syntax ist wie folgt:

**mount -t nfs** <**Rechner**>:<**Remote-Pfad**> <**Lokaler-Pfad**>

Sollen also z. B. die Benutzerverzeichnisse vom Rechner sonne importiert werden, so kann dies mit folgendem Befehl erreicht werden:

```
erde:~ # mount -t nfs sonne:/home /home
```

### 2.8.2 Exportieren von Dateisystemen

Ein Rechner, der Dateisysteme exportiert, wird als NFS-Server bezeichnet. Auf einem NFS-Server müssen die folgenden Netzwerkserver gestartet werden:

- RPC-Portmapper (portmap)
- RPC-Mount-Daemon (rpc.mountd)
- RPC-NFS-Daemon (rpc.nfsd)

Diese werden beim Hochfahren des Systems von den Skripten /etc/init.d/portmap und /etc/init.d/nfsserver gestartet. Das Starten des RPC-Portmappers wurde bereits in Abschnitt 2.7.2 auf Seite 45 beschrieben.

Neben dem Start dieser Daemonen muss noch festgelegt werden, welche Dateisysteme an welche Rechner exportiert werden sollen. Dies geschieht in der Datei `/etc/exports`.

Je Verzeichnis, das exportiert werden soll, wird eine Zeile für die Information benötigt, welche Rechner wie darauf zugreifen dürfen. Alle Unterverzeichnisse eines exportierten Verzeichnisses werden ebenfalls automatisch exportiert. Die berechtigten Rechner werden üblicherweise mit ihren Namen (inklusive Domainname) angegeben, es ist aber auch möglich, mit den Jokerzeichen `'*'` und `'?'` zu arbeiten, die die aus der bash bekannte Funktion haben. Wird kein Rechnername angegeben, hat jeder Rechner die Erlaubnis, auf dieses Verzeichnis (mit den angegebenen Rechten) zuzugreifen.

Die Rechte, mit denen das Verzeichnis exportiert wird, werden in einer von Klammern umgebenen Liste nach dem Rechnernamen angegeben. Die wichtigsten Optionen für die Zugriffsrechte sind in der folgenden Tabelle beschrieben.

| | |
|---|---|
| ro | Dateisystem wird nur mit Leserechten exportiert (Vorgabe). |
| rw | Dateisystem wird mit Schreib- und Leserechten exportiert. |
| root_squash | Diese Option bewirkt, dass der Benutzer `'root'` des angegebenen Rechners keine für `'root'` typischen Sonderrechte auf diesem Dateisystem hat. Erreicht wird dies, indem Zugriffe mit der User-ID `0` auf die User-ID `65534 (-2)` umgesetzt werden. Diese User-ID sollte dem Benutzer `'nobody'` zugewiesen werden (Vorgabe). |
| no_root_squash | Rootzugriffe nicht umsetzen; Rootrechte bleiben also erhalten. |
| link_relative | Umsetzen von absoluten, symbolischen Links (solche, die mit `'/'` beginnen) in eine entsprechende Folge von `'../'`. Diese Option ist nur dann sinnvoll, wenn das gesamte Dateisystem eines Rechners gemountet wird (Vorgabe). |
| link_absolute | Symbolische Links bleiben unverändert. |
| map_identity | Auf dem Client werden die gleichen User-IDs wie auf dem Server verwendet (Vorgabe). |
| map_daemon | Client und Server haben keine übereinstimmenden User-IDs. Durch diese Option wird der nfsd angewiesen, eine Umsetztabelle für die User-IDs zu erstellen. Voraussetzung dafür ist jedoch die Aktivierung des Daemons ugidd. |

Tabelle 2.14: Zugriffsrechte für exportierte Verzeichnisse

Die `exports`-Datei kann beispielsweise aussehen wie Datei 2.8.1.

```
#
# /etc/exports
#
/home            sonne(rw)   venus(rw)
/usr/X11         sonne(ro)   venus(ro)
/usr/lib/texmf   sonne(ro)   venus(rw)
/                erde(ro,root_squash)
/home/ftp        (ro)
# End of exports
```

Datei 2.8.1: `/etc/exports`

Die Datei `/etc/exports` wird von mountd und nfsd gelesen. Wird also eine Änderung daran vorgenommen, so müssen mountd und nfsd neu gestartet werden, damit diese Änderung berücksichtigt wird! Erreicht wird dies am einfachsten mit dem Befehl:

```
erde:~ # rcnfsserver restart
```

# 2.9 DHCP

## 2.9.1 Das DHCP-Protokoll

Das so genannte „Dynamic Host Configuration Protocol“ dient dazu, Einstellungen in einem Netzwerk zentral von einem Server aus zu vergeben, statt diese dezentral an einzelnen Arbeitsplatzrechnern zu konfigurieren. Ein mit DHCP konfigurierter Client verfügt selbst nicht über statische Adressen, sondern konfiguriert sich voll und ganz selbstständig nach den Vorgaben des DHCP-Servers.

Dabei ist es möglich, jeden Client anhand der Hardware-Adresse seiner Netzwerkkarte zu identifizieren und ständig mit denselben Einstellungen zu versorgen, sowie Adressen aus einem dafür bestimmten Pool „dynamisch“ an jeden „interessierten“ Rechner zu vergeben. In diesem Fall wird sich der DHCP-Server bemühen, jedem Client bei jeder Anforderung (auch über längere Zeiträume hinweg) dieselbe Adresse zuzuweisen – dies funktioniert natürlich nicht, wenn es mehr Rechner im Netz als Adressen gibt.

Ein Systemadministrator kann somit gleich in zweierlei Hinsicht von DHCP profitieren. Einerseits ist es möglich, selbst umfangreiche Änderungen der Netzwerk-Adressen oder der Konfiguration komfortabel in der Konfigurationsdatei des DHCP-Servers zentral vorzunehmen, ohne dass eine Vielzahl von Clients einzeln konfiguriert werden müssen. Andererseits können vor allem neue Rechner sehr einfach ins Netzwerk integriert werden, indem sie aus dem Adress-Pool eine IP-Nummer zugewiesen bekommen. Auch für Laptops, die regelmäßig in verschiedenen Netzen betrieben werden, ist die Möglichkeit, von einem DHCP-Server jeweils passende Netzwerkeinstellungen zu beziehen, sicherlich interessant.

Neben IP-Adresse und Netzmaske werden der Rechner- und Domain-Name, der zu verwendende Gateway und Nameserver-Adressen dem Client mitgeteilt. Im Übrigen können auch etliche andere Parameter zentral konfiguriert werden, z. B. ein Timeserver, von dem die jeweils aktuelle Uhrzeit abrufbar ist oder ein Printserver. Im Folgenden möchten wir Ihnen nun einen kurzen Einblick in die Welt von DHCP geben. Wir möchten Ihnen anhand des DHCP-Servers dhcpd zeigen, wie einfach auch in Ihrem Netzwerk die gesamte Netzwerkkonfiguration zentral per DHCP erledigt werden kann.

## 2.9.2 DHCP-Softwarepakete

Bei SuSE Linux sind drei für DHCP relevante Pakete in der Serie `n` enthalten.

Einerseits gibt es den vom Internet Software Consortium herausgegebenen DHCP-Server `dhcpd`, der im Netzwerk die entsprechenden Einstellungen vergibt und verwaltet. Doch während bei SuSE Linux normalerweise nur `dhcpd` als Server in Frage kommt, stehen als DHCP-Clients zwei Alternativen zur Auswahl. Einerseits ist hier der ebenfalls von ISC herausgegebene dhclient zu nennen, andererseits der so genannte „DHCP Client Daemon“ im Paket `dhcpcd`.

Der bei SuSE Linux standardmäßig installierte `dhcpcd` ist sehr einfach zu handhaben und wird beim Starten des Rechners automatisch gestartet, um nach einem DHCP-Server zu suchen. Er kommt ohne eine Konfigurationsdatei aus und sollte im Normalfall ohne weitere Konfiguration funktionieren.

Für komplexere Situationen kann man auf den ISC dhclient zurückgreifen, der sich über die Konfigurationsdatei `/etc/dhclient.conf` steuern lässt. Egal ob eine zusätzliche Domain in die Suchliste aufgenommen oder gar das Verhalten eines Microsoft DHCP-Clients emuliert werden soll – dem technisch versierten Anwender stehen unzählige Möglichkeiten zur Verfügung, das Verhalten des dhclient bis ins Detail seinen Bedürfnissen entsprechend anzupassen.

### 2.9.3 Der DHCP-Server dhcpd

Der *Dynamic Host Configuraton Protocol Daemon* ist das Herz eines DHCP-Systems. Er „vermietet“ Adressen und wacht über deren Nutzung, wie in der Konfigurationsdatei `/etc/dhcpd.conf` festgelegt. Über die dort definierten Parameter und Werte stehen dem Systemadministrator eine Vielzahl von Möglichkeiten zur Verfügung, das Verhalten des DHCP nach seinen Wünschen zu beeinflussen.

Ein Beispiel für eine einfache `/etc/dhcpd.conf`-Datei:

```
default-lease-time 600;         # 10 minutes
max-lease-time 7200;            # 2  hours

option domain-name "kosmos.all";
option domain-name-servers 192.168.1.1 192.168.1.2;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option subnet-mask 255.255.255.0;

subnet 192.168.1.0 netmask 255.255.255.0
 {
  range 192.168.1.10 192.168.1.20;
  range 192.168.1.100 192.168.1.200;
 }
```

Datei 2.9.1: Die Konfigurationsdatei `/etc/dhcpd.conf`

Diese einfache Konfigurationsdatei reicht bereits aus, damit DHCP in Ihrem Netzwerk IP-Adressen zuweisen kann. Bitte achten Sie insbesondere auf die Strichpunkte am Ende jeder Zeile, ohne die dhcpd nicht starten wird!

Wie Sie sehen, lässt sich obige Beispieldatei in drei Blöcke unterteilen:

Im ersten Abschnitt wird definiert, wie viele Sekunden eine IP-Adresse standardmäßig an einen anfragenden Rechner „vermietet“ wird, bevor sich dieser um eine Verlängerung bemühen sollte (`default-lease-time`). Auch wird hier angegeben, wie lange ein Rechner maximal eine vom DHCP-Server vergebene IP-Nummer behalten darf, ohne für diese eine Verlängerung zu beantragen (max-lease-time).

Im zweiten Block werden nun einige grundsätzliche Netzwerk-Parameter global festgesetzt:

- Mit `option domain-name` wird die Default-Domain Ihres Netzwerks definiert.
- Bei `option domain-name-servers` können bis zu drei DNS-Server angegeben werden, die zur Auflösung von IP-Adressen in Hostnamen (und umgekehrt) verwendet werden sollen. Idealerweise sollte auf Ihrem System bzw. innerhalb Ihres Netzwerks ein Nameserver bereits in Betrieb sein, der auch für dynamische Adressen jeweils einen Hostnamen und umgekehrt bereit hält. Mehr über die Einrichtung eines eigenen Nameservers erfahren Sie in Abschnitt 2.6 auf Seite 32.
- `option broadcast-address` legt fest, welche Broadcast-Adresse der anfragende Rechner verwenden soll.
- `option routers` definiert, wohin Datenpakete geschickt werden können, die (aufgrund der Adresse von Quell- und Zielhost sowie Subnetz-Maske) nicht im lokalen Netz zugestellt werden können. Gerade bei kleineren Netzen ist dieser Router auch meist der Übergang zum Internet.
- `option subnet-mask` gibt die an den Client zu übergebende Netzmaske an.

Unterhalb dieser allgemeinen Einstellungen wird nun noch ein Netzwerk samt Subnet Mask definiert. Abschließend muss noch ein Bereich gewählt werden, aus dem der DHCP-Daemon Adressen an anfragende Clients vergeben darf. Im Beispiel stehen alle Adressen zwischen `192.168.1.10` und `192.168.1.20` bzw. `192.168.1.100` und `192.168.1.200` zur Verfügung.

Falls Ihr Server über mehr als eine Netzwerkkarte verfügt, sollten Sie unter `DHCPD_INTERFACE` in der Datei `/etc/rc.config.d/dhcpd.rc.config` eintragen, welche Interfaces überhaupt von dhcpd bedient werden sollen.

Nach diesen wenigen Zeilen sollten Sie bereits in der Lage sein, den DHCP-Daemon mit dem Kommando **`rcdhcpd start`** zu aktivieren, der sogleich zur Verfügung steht. Auch könnten Sie mit **`rcdhcpd syntax-check`** eine kurze, formale Überprüfung der Konfigurationsdatei vornehmen lassen. Sollte wider Erwarten ein Problem mit der Konfiguration auftreten und der Server mit einem Fehler abbrechen und nicht mit einem „done“ starten, finden Sie in der zentralen Systemprotokolldatei `/var/log/messages` meist ebenso Informationen dazu wie auf Konsole 10 (Strg + Alt + F10).

Möchten Sie den dhcpd bei jedem Systemstart automatisch aktivieren, bietet es sich an, in der Datei `/etc/rc.config` die Variable `START_DHCPD` auf `yes` zu setzen. Diese Änderung kann selbstverständlich auch im rc.config-Editor von YaST2 vorgenommen werden.

### 2.9.4 Rechner mit fester IP-Adresse

Nachdem wir es nun geschafft haben, den Server für die Vergabe von dynamischen Adressen zu konfigurieren, sollten wir uns die Vergabe *statischer* Adressen einmal genauer ansehen. Wie eingangs bereits erwähnt, kann mit DHCP auch an

ein- und denselben Rechner bei jeder Anfrage eine ganz bestimmte, definierte Adresse vergeben werden.

Selbstverständlich haben solche expliziten Adresszuweisungen Vorrang vor solchen aus dem Pool der dynamischen Adressen. Im Gegensatz zu diesen verfallen die festen Adressinformationen in keinem Fall, wie es bei den dynamischen der Fall ist, wenn nicht mehr genügend freie Adressen zur Verfügung stehen und deshalb eine Neuverteilung erforderlich ist.

Zur Identifizierung eines mit einer *statischen* Adresse definierten Systems, bedient sich der DHCPD der so genannten Hardwareadresse. Dies ist eine weltweit i. d. R. einmalige, fest definierte Nummer aus sechs Oktettpaaren, über die jedes Netzwerkgerät verfügt, z. B. `00:00:45:12:EE:F4`.

Wird nun die Konfigurationsdatei aus Datei 2.9.1 auf Seite 50 um einen entsprechenden Eintrag wie in Datei 2.9.4 ergänzt, wird DHCPD unter allen Umständen dieselben Daten an den entsprechenden Rechner ausliefern.

```
host erde
 hardware ethernet 00:00:45:12:EE:F4;
 fixed-address 192.168.1.21;
```

Datei 2.9.2: Ergänzungen zur Konfigurationsdatei

Der Aufbau dieser Zeilen ist nahezu selbsterklärend:

Zuerst wird der DNS-Name des zu definierenden Rechners eingetragen (host *hostname*) und in der folgenden Zeile die MAC-Adresse definiert. Diese Adresse kann bei Linux-Rechnern bei aktiviertem Netzwerk übrigens sehr einfach mit dem Befehl **`ifconfig`** herausbekommen werden (bei `HWaddr`), aber auch Windows-Rechnern lassen sich mit dem Befehl **`winipcfg`** (Windows 95/98/ME) bzw. **`ipconfig /all`** (Windows NT/2000) entsprechende Informationen entlocken. In unserem Beispiel wird also dem Rechner, dessen Netzwerkkarte die MAC-Adresse *00:00:45:12:EE:F4* hat, die IP-Adresse `192.168.1.21` sowie der Rechnername `erde` zugewiesen.

Als *hardware*-Typ wird heutzutage in aller Regel `ethernet` zum Einsatz kommen, wobei durchaus auch das vor allem bei IBM-Systemen häufig zu findende `token-ring` unterstützt wird.

### 2.9.5 Weitere Informationen

Wenn Sie an zusätzlichen Informationen interessiert sind, bietet sich z. B. die Seite des *Internet Software Consortium* an, auf der detaillierte Informationen zu DHCP verfügbar sind: `http://www.isc.org/products/DHCP/`.

Auch die neue Version 3 des Protokolls, die sich im Moment im Beta-Test befindet, wird dort dokumentiert. Im Übrigen stehen Ihnen selbstverständlich auch die Manpages zur Verfügung, dies sind insbesondere **`man dhcpd`**, **`man dhcpd.conf`**, **`man dhcpd.leases`** und **`man dhcp-options`**. Auf dem Markt sind bisweilen einige Bücher erschienen, die sich umfassend mit den Möglichkeiten des *Dynamic Host Name Configuration Protocol* auseinander setzen.

Übrigens, `dhcpd` kann sogar anfragenden Rechnern eine in der Konfigurationsdatei mit dem *filename*-Parameter definierte Datei anbieten, die einen bootbaren Betriebssystemkern enthält. Damit lassen sich Clients aufbauen, die über keine Festplatte verfügen und sowohl ihr Betriebssystem wie auch ihre Daten ausschließlich über das Netzwerk laden (*diskless clients*). Dies kann sowohl aus Kosten- als auch aus Sicherheitsgründen interessant sein. In Zusammenarbeit mit dem alice-Paket sind dabei erstaunliche Sachen möglich, aber das ist eine andere Geschichte…

## 2.10 Samba

Mit dem Programmpaket Samba kann ein beliebiger Unix-Rechner zu einem leistungsfähigen File- und Printserver für DOS-, Windows- und OS/2 Rechner ausgebaut werden. Das Samba-Projekt wird vom SAMBA TEAM betreut und wurde ursprünglich von dem Australier ANDREW TRIDGELL entwickelt.

Samba ist inzwischen ein sehr komplexes Produkt. An dieser Stelle kann keine vollständige Darstellung aller Möglichkeiten, sondern nur ein kleiner Einblick in die Funktionalität erfolgen. Im Verzeichnis `/usr/share/doc/packages/samba` finden sich zahlreiche Dokumente, anhand derer man auch komplexe Netzkonfigurationen aufbauen kann. Die Referenz zur Konfigurationsdatei von Samba ist in der Manual-Page von **`smb.conf`** (**`man smb.conf`**).

Samba benutzt das SMB-Protokoll (Server Message Block) der Firma Microsoft, das auf den NetBIOS Diensten aufgesetzt ist. Auf Drängen der Firma IBM gab die Firma Microsoft das Protokoll frei, sodass auch andere Software-Hersteller Anbindungen an ein Microsoft-Domain-Netz finden konnten. Samba setzt das SMB- auf das TCP/IP-Protokoll auf, d. h. auf allen Clients muss auch das Protokoll TCP/IP installiert sein.

### NetBIOS

NetBIOS ist eine Softwareschnittstelle (API), die zur Rechnerkommunikation entworfen wurde. Dabei wird ein Namensdienst (engl. *name service*) bereitgestellt, der zur gegenseitigen Identifikation der Rechner dient. Für die Namensvergabe gibt es keine zentrale Instanz, die Rechte vergeben oder überprüfen könnte. Jeder Rechner am Netz kann beliebig Namen für sich reservieren, sofern diese noch nicht vergeben sind. Die NetBIOS-Schnittstelle kann auf unterschiedlichen Netzarchitekturen implementiert werden. Eine Implementation erfolgt relativ „dicht“ an der Netzwerkhardware und nennt sich NetBEUI. NetBEUI wird häufig als NetBIOS bezeichnet. Netzwerkprotokolle, mit denen NetBIOS implementiert wurde, sind IPX (NetBIOS über TCP/IP) von Novell und TCP/IP.

Die NetBIOS-Namen, die auch bei der Implementation von NetBIOS mittels TCP/IP vergeben werden, haben zunächst einmal nichts mit den in der Datei `/etc/hosts` oder per DNS vergebenen Namen zu tun, NetBIOS ist ein vollständig eigener Namensraum. Es empfiehlt sich jedoch zwecks vereinfachter Administration, zumindest für die Server NetBIOS-Namen zu vergeben, die ihrem DNS-Hostnamen entsprechen. Samba macht dies als Voreinstellung.

#### Clients

Alle gängigen Betriebssysteme wie DOS, Windows und OS/2 unterstützen das SMB-Protokoll. Auf den Rechnern muss das TCP/IP Protokoll installiert sein. Für die verschiedenen UNIX Versionen kann man ebenfalls Samba einsetzen.

SMB-Server stellen den Clients Plattenplatz in Form von so genannten „Shares“ zur Verfügung. Dabei umfasst ein Share ein Verzeichnis mit allen Unterverzeichnissen auf dem Server. Es wird unter einem eigenen Namen exportiert und kann von Clients unter diesem Namen angesprochen werden. Dabei kann der Sharename frei vergeben werden. Er muss nicht dem Namen des exportierten Verzeichnisses entsprechen. Ebenso wird einem exportierten Drucker ein Name zugeordnet, unter dem Clients darauf zugreifen können.

### 2.10.1 Installation und Konfiguration des Servers

Zunächst sollte das Paket `samba` aus der Serie n installiert sein. Durch das Setzen der Variable <START_SMB> auf den Wert `yes` in der Datei `/etc/rc.config`, werden die SMB-Dienste während des Systemstarts gestartet. Manuell startet man die Dienste mit **`rcsmb start`**; mit **`rcsmb stop`** kann man die Dienste beenden.

Die zentrale Konfigurationsdatei von Samba ist die `/etc/samba/smb.conf`. Hier kann der gesamte Dienst konfiguriert werden. Grundsätzlich ist die Konfigurationsdatei `/etc/samba/smb.conf` in zwei Sektionen aufgeteilt. In der so genannten **`[globals]`**-Section werden zentrale und übergreifende Einstellungen vorgenommen. Die zweite Sektion ist die **`[share]`**-Section. Hier werden die Verzeichnisse benutzerabhängig freigegeben und die Datei- und Verzeichnisrechte gesetzt. Soll ein bestimmter Wert aus der **`[share]`**-Section für alle Shares gelten, kann dieser in die **`[globals]`**-Section übernommen werden und gilt somit systemweit für alle Shares. Dieses erspart dem gestressten Administrator etwas Arbeit.

Um das Ganze etwas zu veranschaulichen, ist im zweiteiligen Beispiel 2.10.1 auf der nächsten Seite und 2.10.2 auf Seite 56 die Datei abgebildet; sie wird im folgenden erläutert.

```
; /etc/samba/smb.conf
; Copyright (c) 1999 SuSE GmbH Nuernberg, Germany.
;
[global]
   workgroup = arbeitsgruppe
   guest account = nobody
   keep alive = 30
   os level = 2
   kernel oplocks = false
   security = user
; Uncomment the following, if you want to use an existing
; NT-Server to authenticate users, but don't forget that
; you also have to create them locally!!!
;  security = server
;  password server = 192.168.1.10
;  encrypt passwords = yes
   printing = bsd
   printcap name = /etc/printcap
   load printers = yes
   socket options = TCP_NODELAY

   map to guest = Bad User

; Uncomment this, if you want to integrate your server
; into an existing net e.g. with NT-WS to prevent nettraffic
;  local master = no

; Please uncomment the following entry and replace the
; ip number and netmask with the correct numbers for
; your ethernet interface.
;   interfaces = 192.168.1.1/255.255.255.0

; If you want Samba to act as a wins server, please set
; 'wins support = yes'
   wins support = no

; If you want Samba to use an existing wins server,
; please uncomment the following line and replace
; the dummy with the wins server's ip number.
;   wins server = 192.168.1.1

; Do you wan't samba to act as a logon-server for
; your windows 95/98 clients, so uncomment the
; following:
;   logon script =%U.bat
;   domain logons = yes
;   domain master = yes
; [netlogon]
;   path = /netlogon
```

Datei 2.10.1: Datei /etc/samba/smb.conf, Teil 1

```
[homes]
   comment = Heimatverzeichnis
   browseable = no
   read only = no
   create mode = 0750

[cdrom]
   comment = Linux CD-ROM
   path = /cdrom
   read only = yes
   locking = no
   guest ok = yes

[printers]
   comment = All Printers
   browseable = no
   printable = yes
   public = no
   read only = yes
   create mode = 0700
   directory = /tmp
```

Datei 2.10.2: Datei /etc/samba/smb.conf, Teil 2

#### Erläuterung

- **`workgroup = arbeitsgruppe`** Wie jeder Windows-Rechner wird der Samba-Server einer Arbeitsgruppe zugeordnet, unter der er in der „Netzwerkumgebung“ erscheint. `Arbeitsgruppe` ist die Voreinstellung der deutschen Version von Windows for Workgroups.

- **`guest account = nobody`** Samba benötigt einen in der `/etc/password` aufgeführten Benutzer, der keine oder minimale Rechte im Dateisystem hat. Wenn öffentlich zugängliche Shares (Parameter **`public = yes`**)definiert werden, werden alle Operationen unter dieser Benutzerkennung durchgeführt. Auch wenn kein öffentliches Share definiert ist, muss der **`guest account`** definiert sein, da sonst der Samba-Rechner nicht in der Netzwerkumgebung erscheint.

- **`keep alive = 30`** Windows-Rechner stürzen gelegentlich ab. Wenn sie dabei offene Verbindungen zurücklassen, kann es sein, dass der Server dies erst sehr viel später bemerkt. Damit Samba keine unnötigen Ressourcen auf dem Server verschwendet, kann es mit dem Parameter **`keep alive = 30`** angewiesen werden, alle 30 Sekunden nachzuschauen, ob der Windows-Client noch lebt.

- **`os level = 2`** Der Parameter **`os level = 2`** legt fest, dass Samba für WfW und Windows 95 Browser-Dienste anbietet. Befindet sich ein Windows-NT-Rechner im Netz, wird Samba diesen Dienst nicht anbieten, sondern den NT-Rechner selbst in Anspruch nehmen.

- **`Kernel oplocks = false`** Der Linux–Kernel unterstützt leider noch kein so genanntes opportunistic locking, daher muss die Option **`Kernel oplocks`** auf den Wert `false` gesetzt werden.
- **`security = user`** Hierzu siehe den Abschnitt 2.10.1.

Der Eintrag **`[cdrom]`** ist der nach außen hin sichtbare Verzeichnisname. Unter diesem Namen wird das Verzeichnis in die Netzwerkumgebung exportiert. Es ist von allen Benutzern im Netz erreichbar, da **`guest ok = yes`** gesetzt ist.

Eine besondere Stellung nimmt hier das so genannte **`[home]`**-Share ein. Hat der Benutzer auf dem Linux-File-Server einen gültigen Account und ein eigenes Home-Verzeichnis, so kann er sich bei gültiger Nutzerkennung und Passwort mit diesem verbinden.

- **`path = /cdrom`** Mit `path` wird das Verzeichnis `/cdrom` exportiert.
- **`comment = Linux CD-ROM`** Jedes Share kann bei SMB-Servern mit einem Kommentar versehen werden, der das Share näher kennzeichnet.
- **`browsable = yes`** Diese Einstellung ermöglicht, dass das Share `cdrom` in der Netzwerkumgebung sichtbar wird.
- **`read only = yes`** Samba verbietet in der Voreinstellung den Schreibzugriff auf exportierte Shares. Soll also ein Verzeichnis als schreibbar freigegeben werden, muss man den Wert **`read only = no`** setzten.
- **`create mode = 750`** Windows-Rechner kennen das Konzept der Unix-Zugriffsrechte nicht. Daher können sie bei der Erstellung von Dateien auch nicht angeben, mit welchen Zugriffsrechten dies zu geschehen hat.
  Der Parameter **`create mode`** legt fest, mit welchen Zugriffsrechten Dateien angelegt werden. Dieses gilt nur für schreibbare Shares.
- **`public = yes`** Der Gastzugang zu diesem Share wird erlaubt. Ein Passwort wird nicht abgefragt! Der Benutzer erscheint als User nobody.

#### Security Level

Das SMB-Protokoll kommt aus der DOS/Windows-Welt und berücksichtigt die Sicherheitsproblematik direkt. Jeder Zugang zu einem Share kann mit einem Passwort geschützt werden. SMB kennt drei verschiedene Möglichkeiten, dies zu bewerkstelligen:

- Share Level Security: Bei der Share Level Security wird einem Share ein Passwort fest zugeordnet. Jeder, der dieses Passwort kennt, hat Zugriff auf das Share.
- User Level Security: Diese Variante führt das Konzept des Benutzers in SMB ein. Jeder Benutzer muss sich bei einem Server mit einem Passwort anmelden. Nach der Anmeldung kann der Server dann, abhängig vom Benutzernamen, Zugang zu den einzelnen, exportierten Shares gewähren.

- Server Level Security: Samba behauptet gegenüber den Clients, im User Level Mode zu arbeiten. Allerdings übergibt es alle Passwortanfragen an einen anderen User Level Mode Server, der die Authentifizierung übernimmt. Diese Einstellung erwartet einen weiteren Parameter (**`password server =`**).

Die Unterscheidung zwischen Share, User und Server Level Security gilt für den gesamten Server. Es ist nicht möglich, einzelne Shares per Share Level Security und andere per User Level Security zu exportieren.

Weitere Infos zu diesem Thema finden Sie in der Datei `/usr/share/doc/packages/samba/textdocs/security_level.txt`.

**Tipp**

Für die einfache Administration des Samba-Servers gibt es noch das Programm swat. Es stellt ein einfaches Webinterface zur Verfügung, mit dem Sie bequem den Samba-Server konfigurieren können. Rufen Sie in einem Webbrowser `http://localhost:901` auf und loggen Sie sich als Benutzer root ein. Bitte beachten Sie, dass swat auch in den Dateien `/etc/inetd.conf` und `/etc/services` aktiviert ist. Weitere Informationen zu swat finden Sie in der Manual-Page von **`swat`** (**`man swat`**).

### 2.10.2 Samba als Anmelde-Server

In Netzwerken, in denen sich überwiegend Windows-Clients befinden, ist es oft wünschenswert, dass sich die Benutzer nur mit gültigem Account und Passwort anmelden dürfen. Dies kann mit Hilfe eines Samba-Servers realisiert werden. In einem reinen Windows-Netzwerk übernimmt ein Windows-NT-Server diese Aufgabe, dieser ist als so genannter Primary Domain Controller (PDC) konfiguriert. Es müssen Einträge in die **`[globals]`**-Section der `smb.conf` vorgenommen werden wie in Beispiel 2.10.3.

```
[global]
 workgroup = TUI-NET
 domain logons = yes
 domain master = yes
```

Datei 2.10.3: Global-Section in smb.conf

Werden verschlüsselte Passwörter zur Verifizierung genutzt, muss der Samba Server damit umgehen können. Der Eintrag **`encrypt passwords = yes`** in der **`[globals]`**-Section ermöglicht dies. Außerdem müssen die Benutzeraccounts bzw. die Passwörter in eine Windows konforme Verschlüsselungsform gebracht werden. Das geschieht mit dem Befehl **`smbpasswd -a name`**. Da nach dem Windows NT Domänenkonzept auch die Rechner selbst einen Domänen-Account benötigen, wird dieser mit den folgenden Befehlen angelegt:

Bei dem Befehl **`useradd`** wurde ein Dollarzeichen, maskiert durch den Backslash hinzugefügt. Der Befehl **`smbpasswd`** fügt diesen bei der Verwendung des Parameters -m selbst hinzu.

```
useradd -m rechnername

$
smbpasswd -a -m rechnername
```

Datei 2.10.4: Anlegen eines Maschinenaccounts

### 2.10.3 Installation der Clients

Zunächst sei erwähnt, dass die Clients den Samba-Server nur über TCP/IP erreichen können. NetBEUI oder NetBIOS über IPX sind mit Samba momentan nicht verwendbar. Da TCP/IP überall, sogar bei Novell und Microsoft, auf dem Vormarsch ist, ist es fraglich, ob sich dies jemals ändern wird.

#### Windows 9x/ME

Windows 9x/ME bringt die Unterstützung für TCP/IP bereits mit. Wie bei Windows for Workgroups wird sie jedoch in der Standardinstallation nicht mitinstalliert. Um TCP/IP nachzuinstallieren, wählt man im Netzwerk-Applet der Systemsteuerung '`Hinzufügen...`' unter '`Protokolle`' TCP/IP von Microsoft. Bitte achten Sie auf die korrekte Angabe Ihrer Netzwerkadresse und der Netzwerkmaske! Nach einem Neustart des Windows-Rechners können Sie den richtig konfigurierten Samba-Server in der Netzwerkumgebung wiederfinden (Doppelklick auf das entsprechende Icon auf dem Desktop).

Tipp

Um einen Drucker auf dem Samba-Server zu nutzen, sollte man den allgemeinen oder den Apple PostScript-Druckertreiber von der jeweiligen Windows-Version installieren; am besten verbindet man dann mit der Linux Drucker-Queue, die die automatische apsfilter-Erkennung beinhaltet.

### 2.10.4 Optimierung

Die Standardkonfiguration in `/etc/samba/smb.conf` ist natürlich nicht auf das jeweilige Netz und den entsprechenden Einsatz abgestimmt, so dass auch noch nachgearbeitet werden kann. Da dieses „Fine-Tuning“ von sehr vielen Faktoren abhängig ist, gibt es keine Universallösung. Beachten Sie deshalb bitte auch die Hilfe und die vielen Tipps zur Optimierung in den Dateien `/usr/share/doc/packages/samba/textdocs/Speed.txt` und `/usr/share/doc/packages/samba/textdocs/Speed2.txt`.

## 2.11 Netatalk

Mit dem Paket `netatalk` können Sie einen leistungsfähigen File- und Druckserver für Mac OS-Clients realisieren. Es ist möglich, von einem Macintosh aus auf Daten des Linux-Rechners zuzugreifen oder auf einem angeschlossenen Drucker zu drucken.

Netatalk ist eine Suite von Unix-Programmen, die auf dem im Kernel implementierten DDP (Datagram Delivery Protocol) aufsetzen und die AppleTalk-Protokoll-Familie (ADSP, ATP, ASP, RTMP, NBP, ZIP, AEP und PAP) implementieren.

AppleTalk ist im Prinzip ein Äquivalent zum wesentlich weiter verbreiteten TCP (Transmission Control Protocol). Viele auf TCP/IP aufsetzende Dienste, z. B. zur Auflösung von Hostnamen und Zeitsynchronisation, finden ihre Entsprechung unter AppleTalk. Beispielsweise wird anstelle von **`nslookup`** (DNS, Domain Name Service) der Befehl **`nbplkup`** (NBP, Name Binding Protocol) verwendet, und anstelle von **`ping`** (ICMP ECHO_REQUEST, Internet Control Message Protocol) der Befehl **`aecho`** (AEP, AppleTalk Echo Protocol).

Folgende drei Daemonen werden normalerweise auf dem Server gestartet:

- Der atalkd („AppleTalk-Netzwerk-Manager“), der quasi den Programmen ifconfig und routed entspricht;
- afpd (AppleTalk Filing Protocol daemon), der für Macintosh-Clients ein Interface zu Unix-Dateisystemen zur Verfügung stellt;
- papd (Printer Access Protocol daemon), der Drucker im (AppleTalk-) Netz bereitstellt.

Sie können ohne weiteres – und in heterogenen Netzwerkumgebungen ist dies sehr nützlich – Verzeichnisse auf dem Server nicht nur über Netatalk, sondern gleichzeitig über Samba (für Windows-Clients, siehe voriges Kapitel) und über NFS (siehe 2.8 auf Seite 46), exportieren. Datensicherung und die Verwaltung der Nutzerrechte können zentral auf dem Linux-Server erfolgen.

Beachten Sie bitte:

- Wegen einer Einschränkung der Macintosh-Clients dürfen die Passwörter der Benutzer auf dem Server maximal 8 Zeichen lang sein.
- Auf Unix-Dateien mit Namen länger als 31 Zeichen können Macintosh-Clients nicht zugreifen.
- Dateinamen dürfen keine Doppelpunkte (`':'`) enthalten, weil diese unter Mac OS als Separator in Pfadnamen dienen.

Zu installieren ist das Paket `netatalk`, Serie `n`.

### 2.11.1 Konfiguration des Fileservers

In der Standardkonfiguration ist Netatalk als Fileserver für die auf dem Linux-System eingetragenen Benutzer schon voll funktionsfähig. Um die weitergehen-

den Features zu nutzen, müssen Sie einige Einstellungen in den Konfigurationsdateien vornehmen. Diese befinden sich im Verzeichnis `/etc/atalk`.

Alle Konfigurationsdateien sind reine Textdateien. Text, der hinter einer Raute `'#'` steht, wird ignoriert („Kommentare“), leere Zeilen ebenso.

#### Netz konfigurieren – `atalkd.conf`

In `/etc/atalk/atalkd.conf` legt man fest, über welche Interfaces die Dienste angeboten werden. Meist ist dies `eth0`, und es genügt, wenn hier als einziger Wert

```
eth0
```

eingetragen ist (dies ist in der Beispieldatei der Fall). Hier tragen Sie weitere Interfaces ein, wenn Sie z. B. mehrere Netzwerkkarten gleichzeitig verwenden. Wird der Server gestartet, sucht er im Netzwerk nach bereits vorhandenen Zonen und Servern und verändert die entsprechende Zeile, indem er die konfigurierten AppleTalk-Netzwerk-Adressen einträgt. Sie finden dann eine Zeile wie

```
eth0 -phase 2 -net 0-65534 -addr 65280.57
```

am Ende der Datei. Sollten Sie komplexere Konfigurationen vornehmen wollen, finden Sie in der Konfigurationsdatei Beispiele. Dokumentation über weitere Optionen können Sie außerdem der Manual-Page zum afpd entnehmen.

#### Fileserver definieren – `afpd.conf`

In der Datei `afpd.conf` wird festgelegt, wie Ihr Fileserver auf Mac-OS-Rechnern in der ‘`Auswahl`’ erscheint. Wie die anderen Konfigurationsdateien enthält auch diese ausführliche Kommentare, die die vielfältigen Optionen erklären.

Ändern Sie hier nichts, wird einfach der Default-Server gestartet und in der ‘`Auswahl`’ mit dem Hostnamen angezeigt. Sie müssen also hier nicht unbedingt etwas eintragen, allerdings ist es auch möglich, Fileserver mit verschiedenen Namen und Optionen zu definieren, um z. B. einen speziellen „Guest Server“ anzubieten, auf dem man als „Gast“ Dateien ablegen kann:

```
"Guest server" -uamlist uams_guest.so
```

Oder Sie können einen Server definieren, der keinen Gastzugang erlaubt, sondern nur für Benutzer zugänglich ist, die auf dem Linux-System existieren:

```
"Font server" -uamlist uams_clrtxt.so,uams_dhx.so
```

Dieses Verhalten wird gesteuert durch die Option `uamlist` gefolgt von einer durch Kommata getrennten Liste der zu verwendenden Authentifizierungsmodule. Default ist, dass alle Verfahren aktiv sind.

Ein AppleShare-Server stellt seine Dienste standardmäßig nicht nur über AppleTalk, sondern auch („encapsulated“) über TCP/IP zur Verfügung. Der Default-Port ist 548. Für zusätzliche AppleShare-Server (auf dem gleichen Rechner)

müssen Sie, wenn diese ebenfalls auch über TCP laufen sollen, dedizierte Ports zuweisen. Die Bereitstellung des Dienstes über TCP/IP ermöglicht den Zugriff auf den Server auch über nicht AppleTalk-Netze wie zum Beispiel das Internet.

Die Syntax wäre dann z. B.:

```
"Font server" -uamlist uams_clrtxt.so,uams_dhx.so -port 12000
```

Der AppleShare-Server erscheint hier im Netz mit dem Namen „Font Server“, erlaubt keinen Zugriff für Gäste und ist auf den Port 12 000 eingestellt. Damit ist er auch über TCP/IP-Router hinweg erreichbar.

Welche (auf dem Server liegenden) Verzeichnisse der jeweilige AppleShare-Server dann als Netz-„Volumes“ bereitstellt, wird in der Datei `AppleVolumes.default` definiert (die weiter unten näher erläutert wird). Mit der `-defaultvol` Option können Sie für einen einzelnen AppleShare-Server auch eine andere Datei festlegen, in der abweichende Vorgaben gemacht werden, z. B. (in einer Zeile):

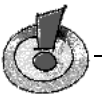
```
"Guest server" -uamlist uams_guest.so -defaultvol
/etc/atalk/AppleVolumes.guest
```

Weitere Optionen sind in der Datei `afpd.conf` selbst erklärt.

#### Verzeichnisse und Zugriffsrechte – `AppleVolumes.default`

Hier legen Sie Verzeichnisse fest, die exportiert werden sollen. Die Zugriffsrechte werden dabei durch die unter Unix üblichen Benutzer- und Gruppen-Rechte festgelegt.

Dies wird in der Datei `AppleVolumes.default` eingerichtet.



**Hinweis**

Hier hat sich die Syntax teilweise geändert. Bitte berücksichtigen Sie dies, wenn Sie von einer älteren Version updaten; z. B. heißt es statt `access=` jetzt `allow:` (ein charakteristisches Symptom wäre, wenn Sie auf den Mac-Clients unter AppleTalk statt der Laufwerksbezeichnung deren Optionen angezeigt bekommen.) Da bei einem Update die neuen Dateien mit der Endung `.rpmnew` angelegt werden, kann es sein, dass Ihre alten Einstellungen unter Umständen wegen der geänderten Syntax nicht mehr funktionieren.

Wir empfehlen Ihnen, ein Backup von Ihren Konfigurationsdateien zu machen, aus diesen Ihre alten Einstellungen in die neuen Dateien zu übernehmen und diese dann umzubenennen. So profitieren Sie auch von den aktuellen ausführlichen Kommentaren, die zur Erklärung der diversen Optionen in den Konfigurationsdateien enthalten sind.

Neben `AppleVolumes.default` können zusätzliche Dateien angelegt werden, z. B. `AppleVolumes.guest`, die von bestimmten Servern benutzt werden (indem in der Datei `afpd.conf` die `-defaultvol`-Option benutzt wird – siehe voriger Abschnitt).

Die Syntax ist denkbar einfach:

```
/usr/local/psfonts "PostScript Fonts"
```

bedeutet, dass das in dem Rootverzeichnis liegende Linux-Verzeichnis `/usr/local/psfonts` als AppleShare-Volume mit dem Namen „PostScript Fonts“ freigegeben wird.

Optionen werden, durch Leerzeichen getrennt, an die Zeile angehängt. Eine sehr nützliche Option ist die Zugriffsbeschränkung:

```
/usr/local/psfonts "PostScript Fonts" allow:User1,@gruppe0
```

was den Zugriff auf das Volume „PostScript Fonts“ auf den Benutzer „User1“ und alle Mitglieder der Gruppe „gruppe0“ beschränkt. Diese müssen natürlich dem Server bekannt sein. Entsprechend können Sie mit `deny:User2` auch explizit Nutzer ausschließen.

Bitte berücksichtigen Sie, dass diese Einschränkungen für den Zugriff über AppleTalk gelten und nichts mit den Rechten zu tun haben, die der User hat, wenn er sich auf dem Server selber einloggen kann.

Netatalk legt zur Abbildung der Mac-OS-typischen Ressource-Fork von Dateien im Linux-Dateisystem `.AppleDouble`-Verzeichnisse an. Mit der Option `noadouble` können Sie bestimmen, dass diese Verzeichnisse erst dann angelegt werden, wenn sie tatsächlich benötigt werden. Syntax:

```
/usr/local/guests "Guests" options:noadouble
```

Weitere Optionen und Möglichkeiten entnehmen Sie bitte den Erklärungen in der Datei selbst.

Übrigens: In dieser Konfigurationsdatei finden Sie ebenfalls eine kleine unschuldige Tilde (`'~'`). Diese Tilde steht für das Homeverzeichnis eines jeden Benutzers auf dem Server. Dadurch kann jedem Benutzer automatisch sein Homeverzeichnis bereitgestellt werden, ohne dass jedes einzelne hier explizit angegeben werden müsste. Die installierte Beispieldatei enthält bereits eine Tilde, weshalb Netatalk standardmäßig die Homeverzeichnisse bereitstellt, wenn Sie an dieser Datei nichts ändern.

Der afpd sucht außerdem im Homeverzeichnis eines angemeldeten Benutzers nach einer Datei `Applevolumes` oder `.Applevolumes`. Einträge in dieser Datei ergänzen die Einträge in den Serverdateien `AppleVolumes.system` und `AppleVolumes.default`, um weitere individuelle type/creator-Zuordnungen zu ermöglichen und auf Dateisysteme zuzugreifen. Diese Einträge sind Ergänzungen und ermöglichen keine Zugriffe, die nicht von Serverseite für diesen Benutzer erlaubt sind.

Die Datei `netatalk.pamd` dient der Authentifizierung über PAM (Pluggable Authentication Modules), was in unserem Rahmen hier ohne Bedeutung ist.

#### Dateizuordnungen – `AppleVolumes.system`

In der Datei `AppleVolumes.System` legen Sie fest, welche (Mac-OS-typischen) Type- und Creator-Zuordnungen zu bestimmten Dateiendungen erfolgen soll. Eine ganze Reihe von Standardwerten sind schon vorgegeben. Wenn

eine Datei mit einem generischen weißen Icon angezeigt wird, ist in diesem Fall noch kein Eintrag vorhanden. Sollten Sie Probleme haben, eine Textdatei eines anderen Systems unter Mac OS korrekt öffnen zu können, bzw.das umgekehrte Problem, kontrollieren Sie dort die Einträge.

### 2.11.2 Konfiguration des Druckservers

Über die Datei `papd.conf` konfigurierbar wird ein Laserwriter-Dienst zur Verfügung gestellt. Der Drucker muss lokal schon mit dem **`lpd`** funktionieren. Wenn Sie mit dem Kommando **`lpr datei.txt`** lokal drucken können, ist der erste Schritt erfolgreich getan.

Sie müssen in `papd.conf` nichts eingeben, wenn unter Linux ein lokaler Drucker eingerichtet ist, da ohne weitere Angaben Druckaufträge einfach an den Druck-Daemon **`lpd`** weitergegeben werden. Der Drucker meldet sich im AppleTalk-Netz als Laserwriter. Sie können aber auch bestimmte Drucker wie folgt eintragen:

```
Drucker_Empfang:pr=lp:pd=/etc/atalk/kyocera.ppd
```

Dies lässt den Drucker mit dem Namen **`Drucker_Empfang`** in der Auswahl erscheinen. Die entsprechende Druckerbeschreibungsdatei gibt es gewöhnlich beim Hersteller. Ansonsten nehmen Sie einfach die Datei `Laserwriter` aus dem Ordner ‘`Systemerweiterungen`’; allerdings können Sie dann meist nicht alle Features benutzen.

### 2.11.3 Starten des Servers

Der Server wird per „Init-Skript“ beim Systemstart gestartet oder per Hand mit: **`rcatalk start`**

Um den Server beim Systemstart zu aktivieren, muss in `/etc/rc.config` die Variable **`START_ATALK`** auf `yes` gesetzt werden. Das Init-Skript befindet sich in `/etc/init.d/atalk`. Den Start erledigt das Startskript im Hintergrund; es dauert ca. eine Minute, bis die AppleTalk-Interfaces konfiguriert und erreichbar sind. Sie können mit einer Statusabfrage sehen, ob es soweit ist (erkennbar daran, dass dreimal `OK` ausgegeben wird):

```
erde: # rcatalk status
Checking for service atalk:OKOKOK
erde: #
```

Gehen Sie nun an einen Mac, der unter Mac OS läuft. Kontrollieren Sie, dass AppleTalk aktiviert ist, wählen Sie ‘`Filesharing`’, doppelklicken Sie ‘`Appleshare`’; in dem Fenster sollten Sie nun den Namen Ihres Servers sehen. Doppelklicken Sie ihn und melden sie sich an. Wählen Sie das Laufwerk und – voilà – hier ist Ihr Netzlaufwerk unter Mac OS.

Mit Servern, die nur über TCP und nicht über DDP laufen, können Sie sich verbinden, indem Sie in der ‘`Auswahl`’ auf ‘`Server IP-Adresse`’ klicken und die entsprechende IP-Adresse, gegebenenfalls gefolgt von einem Doppelpunkt und der Portnummer, eingeben.

### Weiterführende Informationen

Um alle Möglichkeiten, die das Paket `netatalk` bietet, voll auszuschöpfen, empfiehlt es sich, in den entsprechenden Manual-Pages zu stöbern. Diese finden Sie mit dem Befehl: **`rpm -qd netatalk`** Noch ein Hinweis: Die Datei `/etc/atalk/netatalk.conf` wird in unserer Version von netatalk nicht verwendet, Sie können sie einfach ignorieren. Hilfreiche URLs:

- `http://netatalk.sourceforge.net/`
- `http://www.umich.edu/~rsug/netatalk/`
- `http://www.anders.com/projects/netatalk/`
- `http://cgi.zettabyte.net/fom-serve/netatalk/cache/1.html`

Und wie sieht es eigentlich „andersherum“ aus? Kann ich unter Linux ein AppleShare-Laufwerk erreichen? Die ehrlichste Antwort ist: Besser nicht, da das entsprechende Paket, sich in einem Prä-Alpha-Stadium befindet. Tapfere Experimentatoren finden es unter: `http://www.panix.com/~dfoster/afpfs/`

## 2.12 Netware-Emulation mit MARSNWE

Der Netware-Emulator MARSNWE kann einen Novell-Netware 2.2 bzw. 3.11 Server für Datei- und Druckdienste relativ leicht ersetzen und kann dabei auch als IPX-Router verwendet werden. Die Funktionalität neuerer Netware Versionen, wie z. B. NDS (engl. *Netware Directoy Services*) kann er allerdings nicht bieten. Arbeitsstationen, die mit DOS oder Windows laufen und bereits für den Zugriff auf einen Netware 2.2/3.11/3.12-Server konfiguriert sind, können mit minimalen Änderungen den Linux-Server mit dem Netware-Emulator MARSNWE als Server nutzen. Die Administration erledigt man am besten unter Linux, da die Novell-Programme zur Systemadministration nur bedingt verwendbar sind und dabei auch Lizenzen zu beachten sind.

### 2.12.1 Netware Emulator MARSNWE starten

Der MARSNWE auf SuSE Linux kann sofort nach der Installation gestartet werden, da er bereits soweit vorkonfiguriert ist, dass man ihn sofort testen kann. Die erforderliche IPX-Unterstützung seitens des Kernels ist als ladbares Kernelmodul vorhanden und wird vom Startskript bei Bedarf automatisch geladen. Das Aufsetzen des IPX-Interfaces wird vom MARSNWE automatisch durchgeführt. Netznummer und zu verwendendes Protokoll werden dabei der ausführlich kommentierten Konfigurationsdatei `/etc/nwserv.conf` entnommen. Gestartet wird der MARSNWE mit dem Kommando **`rcnwe start`**. Meldet er dabei am rechten Bildschirmrand in grün **`done`**, wurde er erfolgreich gestartet und man kann mit **`ifconfig`** kontrollieren, ob IPX auf dem Netzwerkinterface korrekt aufgesetzt wurde.

Ob der Netware-Emulator läuft, überprüft man mit **`rcnwe status`**, und beendet wird er mit **`rcnwe stop`**. Damit er bereits beim Booten gestartet wird, muss lediglich die Variable **`START_MARSNWE`** in der Datei `/etc/rc.config` auf **`yes`** gesetzt werden.

### 2.12.2 Die Konfigurationsdatei /etc/nwserv.conf

Die Konfigurationsoptionen sind zu „Sections“ zusammengefasst, die einfach durchnummeriert werden. Jede Konfigurationszeile beginnt dabei immer mit der Nummer der jeweiligen Section. Interessant sind lediglich die Sections 1 bis 22, wobei aber nicht alle Nummern verwendet werden. Im Normalfall kommt man mit folgenden Sections für die Konfiguration aus:

**1** Netware Volumes

**2** Servername

**4** IPX-Netzwerk

**13** Benutzernamen

**21** Drucker

Nach Änderungen an der Konfiguration, muss MARSNWE mit dem Befehl **`rcnwe restart`** neu gestartet werden.

Die Konfigurations-Optionen im Detail:

**Volumes (Section 1):**

```
1    SYS     /usr/local/nwe/SYS/     kt      711 600
```

Hier werden die zu exportierenden Volumes definiert. Jede Zeile beginnt mit der Nummer der Section (hier 1), danach folgt der Volume-Name und dann der Pfad des Verzeichnisses auf dem Server. Zusätzlich können noch diverse Optionen angegeben werden, die durch einzelne Buchstaben dargestellt sind, sowie jeweils eine UMASK für das Erzeugen von Verzeichnissen und eine für Dateien. Wird keine UMASK angegeben, wird der Standardwert aus Section 9 verwendet. Das Volume für SYS ist bereits eingetragen. Um Probleme mit Groß- und Kleinschreibung bei Dateinamen zu vermeiden, empfiehlt sich die Verwendung der Option **`k`**, denn dann werden alle Dateinamen in Kleinschreibung konvertiert.

**Servername (Section 2):**

```
2    MARS
```

Diese Angabe ist Optional, standardmäßig wird der Hostname verwendet.

**Interne Netznummer (Section 3):**

```
3    auto
```

Die interne Netznummer wird aus der MAC-Adresse der Netzwerkkarte generiert, wenn hier **auto** angegeben ist. Normalerweise behält man diese Einstellung bei.

**IPX-Konfiguration (Section 4):**

```
4    0x0     *       AUTO            1
4    0x22    eth0    ethernet_ii     1
```

Hier gibt man die Netware-Netznummer an und auf welche Netzwerk-Schnittstelle es mit welchem Protokoll gebunden werden soll. Das erste Beispiel setzt alles automatisch auf, während das zweite die Netznummer **0x22** auf die Netzwerkkarte **eth0** mit dem Frametyp **Ethernet-II** bindet. Hat man mehrere Netzwerkkarten und trägt diese alle mit unterschiedlichen Netznummern ein, so wird IPX dazwischen geroutet.

**Create Mode (Section 9):**

```
9    0751    0640
```

Gibt die Standardrechte an, mit denen Verzeichnisse und Dateien angelegt werden.

**GID und UID mit minimalen Rechten (Section 10, 11):**

```
10   65534
11   65534
```

Gruppen- und Benutzer-ID für nicht angemeldete Benutzer. Hier **nogroup** und **nobody**.

**Supervisor Login (Section 12):**

```
12   SUPERVISOR      root
```

Der Supervisor wird auf den Benutzer root abgebildet.

**Benutzer Logins (Section 13):**

```
13   LINUX           linux
```

Die Zuordnung der Netware-Benutzer zu den Linux-Usern wird hier festgelegt. Optional kann ein festes Passwort mit eingetragen werden.

**Automatische Benutzer-Abbildung (Section 15):**

```
15   0       top-secret
```

Gibt man hier statt der `0` eine `1` an, werden die Linux Logins automatisch als Netware Logins zur Verfügung gestellt, das Passwort ist in diesem Fall „top-secret“.

**Drucker-Queues (Section 21):**

```
21   LP      -       lpr -
```

Der erste Parameter **`LP`** ist der Name des Netware-Druckers, als zweites kann man den Namen des Spool-Verzeichnisses angeben und als drittes das Druckkommando.

**Print-Server (Section 22):**

```
22   PS_NWE  LP_PS   1
```

Drucker die über pserver aus dem Paket `ncpfs` angesprochen werden, können hier definiert werden.

### 2.12.3 Zugriff auf Netware-Server und deren Administration

Das Paket `ncpfs` aus der Serie `n` ist eine Sammlung kleiner Programme, mit denen man einen Netware 2.2/3.11 Server von Linux aus administrieren, Netware-Volumes mounten oder Drucker verwalten kann. Will man auf neuere Netware-Server ab Version 4 zugreifen, muss auf diesen die Bindery-Emulation und IPX aktiviert sein.

Folgende Programme stehen dafür zur Verfügung, deren Funktion man den Manpages dazu entnehmen kann:

| | | | |
|---|---|---|---|
| nwmsg | ncopy | ncpmount | ncpumount |
| nprint | nsend | nwauth | nwbocreate |
| nwbols | nwboprops | nwborm | nwbpadd |
| nwbpcreate | nwbprm | nwbpset | nwbpvalues |
| nwdir | nwdpvalues | nwfsctrl | nwfsinfo |
| nwfstime | nwgrant | nwpasswd | nwpurge |
| nwrevoke | nwrights | nwsfind | nwtrustee |
| nwtrustee2 | nwuserlist | nwvolinfo | pqlist |
| pqrm | pqstat | pserver | slist |

Wichtig ist z. B. **`ncpmount`**, mit dem man Volumes von einem Netware-Server unter Linux mounten kann und **`ncpumount`** um es wieder zu unmounten.

Außerdem enthält das Paket `ncpfs` Tools zur Konfiguration des IPX-Protokolls und IPX-Routing:

```
ipx_cmd
ipx_configure
ipx_interface
ipx_internal_net
ipx_route
```

Mit **ipx_configure** oder **ipx_interface** kann man die IPX-Konfiguration der Netzwerkkarte vornehmen. Hat man MARSNWE laufen macht dieser das aber bereits automatisch.

### 2.12.4 IPX-Router mit ipxrip

Ein weiteres Paket, um Linux in einen IPX-Router zu verwandeln ist Paket `ipxrip` aus der Serie `n`. In der Regel wird man es aber nicht benötigen, da man mit MARSNWE oder den Tools aus Paket `ncpfs` ebenfalls einen IPX-Router konfigurieren kann.

# 3 Der Anschluss an die weite Welt – PPP, ISDN, Modem, Fax...

Neben der Netzwerkanbindung im lokalen Netz ist der Anschluss an ein größeres und verteiltes Netz, an ein WAN (engl. *Wide Area Networks*) oder die Nutzung von Mailboxen von Interesse.

In der Unix-Welt haben sich zwei Standards zum Anschluss an große Netze durchgesetzt: UUCP und TCP/IP über Modemverbindungen bzw. über ISDN. Während UUCP (**U**nix to **U**nix **C**o**P**y) hauptsächlich dem Transport von News und E-Mail dient, stellt eine TCP/IP-Verbindung eine *echte* Netzwerkanbindung dar. Eine solche echte Netzanbindung erlaubt es, alle aus dem LAN bekannten Dienste global zur Verfügung zu stellen – dem LAN liegt ja auch TCP/IP zu Grunde.

Wird TCP/IP über eine Modem- oder ISDN-Verbindung gefahren, kommt heutzutage zumeist PPP (**P**oint to **P**oint **P**rotocol) zum Einsatz.[1] Bei ISDN wird zumeist `syncPPP` gewählt, manchmal aber auch `rawip` Wie solch eine WAN-Anbindung erfolgen kann, ist Thema der folgenden Abschnitte. PPP wird kurz vorgestellt (Abschnitt 3.1 auf der nächsten Seite) und dann die ISDN-Konfiguration beschrieben. Danach wird der Anschluss eines analogen Modems (Abschnitt 3.5 auf Seite 96) und die Konfiguration einer PPP-Verbindung für Modems erklärt (Abschnitt 3.6 auf Seite 97). E-Mail-Anschluss und News-System-Einrichtung werden in ihren Grundzügen präsentiert.

[1] SLIP (**S**erial **L**ine **I**nternet **P**rotocol) gerät mehr und mehr aus der Mode.

## 3.1 PPP

PPP (engl. *Point to Point Protocol*) bietet die Möglichkeit, TCP/IP über eine serielle Leitung zu betreiben. PPP-Client und -Server können sich beim Verbindungsaufbau über diverse Protokollparameter verständigen. Der Server kann dem Client seine IP-Adresse mitteilen und ihm eine IP-Adresse zuordnen.

PPP ist – im Gegensatz zu SLIP – ein definierter Standard und wird von den meisten Internet-Providern inzwischen als einzige Einwahlmöglichkeit angeboten. Die zentrale Rolle bei PPP spielt der PPP-Daemon pppd, über den die PPP-Geräte angesprochen werden; der PPP-Daemon kann sowohl als Client als auch als Server eingesetzt werden. Zum eigentlichen Verbindungsaufbau wird das Programm wvdial oder das Programm chat benötigt.

### 3.1.1 Voraussetzungen für PPP

Die Voraussetzungen für PPP unter SuSE Linux sind:

- Der Kernel muss TCP/IP und PPP unterstützen.
- Die Netzwerkpakete müssen installiert sein. Unbedingt erforderlich sind die Pakete `netcfg` und `net-tools` aus der Serie a.
- Das grundlegende Paket ppp, Serie n, das den pppd und das Programm chat enthält, ist ebenfalls notwendig.
- Wenn ein analoges Modem zum Einsatz kommen soll, ist das Paket `wvdial`, Serie n für den Auf- und Abbau der Verbindungen erforderlich.
- Wenn ISDN konfiguriert werden soll, vgl. zudem Abschnitt 3.2.3 auf Seite 76.
- Login und Passwort beim PPP-Server müssen bekannt sein.

### 3.1.2 Weitere Informationen zu PPP

PPP bietet eine Fülle von Möglichkeiten, die Verbindung zu konfigurieren. Werden mehr als die im folgenden vorgestellten Optionen benötigt, kann in den entsprechenden Manpages nachgelesen werden, z. B. in der Manual-Page von **`pppd`** (**`man 8 pppd`**). Weiterhin gibt es ausführliche Darstellungen in den Dateien `NET3-4-HOWTO.gz` (früher: `NET-3-HOWTO.gz`) und `PPP-HOWTO.gz` im Verzeichnis `/usr/share/doc/howto/en` sowie in den Dokumentations-Dateien im Verzeichnis `/usr/share/doc/packages/ppp` oder `/usr/share/doc/packages/wvdial`.

Detaillierte Informationen zu den von PPP benutzten Protokollen finden Sie in den zugehörigen RFCs:

- RFC1144: Jacobson, V.: Compressing TCP/IP headers for low-speed serial links, 1990 February.
- RFC1321: Rivest, R.: The MD5 Message-Digest Algorithm, 1992 April.

- RFC1332: McGregor, G.: PPP Internet Protocol Control Protocol (IPCP), 1992 May.
- RFC1334: Lloyd, B., Simpson, W.A.: PPP authentication protocols, 1992 October.
- RFC1548: Simpson, W.A.: The Point-to-Point Protocol (PPP), 1993 December.
- RFC1549: Simpson, W.A.: PPP in HDLC Framing, 1993 December.

## 3.2 Internet-Zugang mit ISDN

SuSE Linux eignet sich hervorragend, Netzwerkverbindungen zu anderen Rechnern (z. B. zu Internet-Providern) über ISDN aufzubauen und zu verwalten. Ein großer Teil der ISDN-Konfiguration kann von YaST bzw. YaST2 aus durchgeführt werden. Diese Beschreibung konzentriert sich in erster Linie auf eine Standard-Anbindung an einen anderen Rechner via ISDN.

**Achtung**

Die hier beschriebenen Verfahren sind unter Umständen nicht zugelassen. Bei aktiven ISDN-Karten besitzt die Karte mitsamt der Firmware eine Zulassung, die auch für den Betrieb unter Linux gilt. Bei passiven Karten gilt generell die Zulassung der Karte nur dann, wenn Sie mit der Software des Herstellers betrieben wird. Eine Ausnahme sind die Karten ELSA Microlink PCI (früher Quickstep) und Eicon Diva 2.01 – diese sind unter Linux ebenfalls zugelassen.
Wer auf eine Zulassung angewiesen ist, muss eine aktive Karte einsetzen oder die passive Karte an einer TK-Anlage anschließen.

Im Gegensatz zu Modemverbindungen muss kein spezielles Kommando gestartet werden, um eine Verbindung zu initiieren. Ist das Netzwerk gestartet, kann jederzeit eine Verbindung zum Partner durch normale Aktivitäten wie telnet, http (WWW), ftp etc. hergestellt werden. Erst dann wird die Wählverbindung aufgebaut. Dieser Vorgang dauert etwa drei Sekunden. Außer `'root'` können auch andere Benutzer eine Verbindung starten. Man kann einstellen, wie viele Sekunden die Verbindung inaktiv sein soll, bevor automatisch aufgelegt wird.

Während der gesamten ISDN-Konfiguration ist es ratsam, die Systemmeldungen in der Datei `/var/log/messages` zu verfolgen. Laden Sie dazu die Datei in den „Viewer“ less:

```
erde: # less +F /var/log/messages
```

Die Option **+F** veranlasst, dass der Bildschirm dann immer die jeweils dazugekommenen Zeilen dieser Datei „online“ anzeigt; mit (Strg) + (c) verlassen Sie diesen Modus wieder.

Mit dem Programm xisdnload haben Sie auch die Möglichkeit, den ISDN-Verkehr grafisch zu überwachen.

### 3.2.1 ISDN einrichten

Im folgenden Abschnitt wird eine Schritt-für-Schritt-Anleitung für den Einstieg ins Internet angeboten.

1. Falls Sie unter dem X Window System (z. B. mit KDE) gearbeitet haben, beenden Sie bitte die X-Sitzung und wechseln Sie auf eine Text-Konsole mit der Tastenkombination (Strg) + (Alt) + (F2). Andernfalls kann es zu Schwierigkeiten beim Runlevel-Wechsel kommen.
2. Melden Sie sich als `'root'` an. Geben Sie im Terminalfenster **`init 1`** ein, um in den Runlevel 1 zu gelangen.
3. Starten Sie als `'root'` an der Konsole YaST.
4. Gehen Sie in YaST auf '`Administration des Systems`', '`Hardware in System integrieren`' und dann auf '`ISDN-Hardware konfigurieren`'.
5. Füllen Sie die Maske aus. Falls Sie nicht genau wissen, was Sie eingeben sollen, so finden Sie die Dokumentation unter `/usr/share/doc/packages/i4l` und in den nachfolgenden Abschnitten.
6. Wählen Sie dann in dem Fenster '`Starten`'.
7. War dies erfolgreich (positive Rückmeldung erscheint am Bildschirm), dann wählen Sie '`ISDN-Parameter`'; falls Sie eine Fehlermeldung erhalten, überprüfen Sie bitte Ihre Eingaben.
8. Füllen Sie auch diese Maske aus.
9. Wählen Sie in dem Fenster '`Starten`'.
10. War dies erfolgreich (positive Rückmeldung erscheint am Bildschirm), dann wählen Sie '`Speichern`'.
11. Gehen Sie auf '`Konfiguration Nameserver`', beantworten Sie die Frage mit '`Ja`'.
12. Geben Sie die IP-Adresse des Nameservers (DNS) Ihres Providers an. Wenn Sie diese nicht wissen, so erfragen Sie diese bei Ihrem Provider, meist findet man diese auch auf der Website des Providers.
13. Gehen Sie jetzt auf '`Netzwerk konfigurieren`' und '`Netzwerk Grundkonfiguration`'.
14. Legen Sie ein neues Device an (mit (F5)) und zwar `ISDN SyncPPP`.
15. Drücken Sie (F6) ('`F6=IP-Adresse`'), um in das Menü '`Eingabe der Netzwerk-Adressen`' zu kommen.
16. Belassen Sie die Einstellungen – die IP-Adresse Ihres lokalen Rechners ist `192.168.0.99`, die Adresse des Point-To-Point-Partners ist `192.168.0.1` – und ändern Sie nur das „Default-Gateway“ auf dieselbe Adresse wie

„Adresse des Point-To-Point-Partners“ (`192.168.0.1`). Bei der dynamischen Adressen-Zuweisung erscheinen zunächst unter „Adresse des Point-To-Point-Partners“ und „Default-Gateway“ Adressen, die nur als Platzhalter für die später zugewiesenen Adressen dienen. Beenden Sie mit ‘`Weiter`’.

17. In der Maske ‘`Auswahl des Netzwerks`’ müssen Sie die Karte noch mit (F4) aktiv setzen.
18. Speichern Sie mit (F10) und beenden Sie YaST.
19. Starten Sie das Netzwerk mit **`init 3`**, wenn Sie X mit **`startx`** starten. Falls Sie sich grafisch anmelden, starten Sie mit **`init 5`**.
20. Testen Sie den Verbindungsaufbau mit **`ping`** . Starten Sie auch **`xisdnload`** , um Ihren Verbindungsstatus beobachten zu können:

    ```
    erde:~ # xisdnload &
    erde:~ # ping -c 3 www.suse.de
    ```

    Nach drei „Pings“ beendet sich der Befehl automatisch (vgl. Manual-Page von **`ping`** (**`man ping`**)).

Nun starten Sie unter X z. B. Netscape oder einen anderen Webbrowser, und schon können Sie auf das Internet zugreifen. Bei manchen Providern müssen Sie noch den jeweiligen *Proxyserver* angeben:

- In Netscape wählen Sie ‘`Edit`’, ‘`Preferences`’, ‘`Advanced`’, ‘`Proxies`’, ‘`Manual Proxy Configuration`’ und dann ‘`View`’.

Jetzt sollte Ihr ISDN-Zugang funktionieren und die Verbindung in das Internet automatisch aufgebaut werden, sobald Sie z. B. in Netscape eine Internetadresse angeben, und abgebaut werden, sobald 60 Sekunden lang kein Datenpaket übertragen wird. Kontrollieren Sie den Verkehr mit **`xisdnload`** !

### 3.2.2 Überblick

SuSE Linux enthält das Paket `isdn4linux`, ein Programmpaket mit Hardware-Treibern, Netzwerkinterface und Modem-Emulation (nur digitales Modem). Außerdem ist z. B. Software für einen Anrufbeantworter verfügbar.

Der Hardware-Treiber zur ISDN-Karte wird von dem Startskript `/etc/init.d/i4l_hardware` geladen. Die Konfiguration der ISDN-Seite übernimmt das Tool isdnctrl (vgl. Manual-Page von **`isdnctrl`** (**`man isdnctrl`**)). Die Konfiguration der zur Verfügung gestellten Netzwerkinterfaces geschieht wie bei einem Ethernet-Interface durch die Befehle **`ifconfig`** (Manual-Page von **`ifconfig`** (**`man ifconfig`**)) und **`route`** (Manual-Page von **`route`** (**`man route`**)). Bei SuSE Linux werden diese Aufgaben von dem Skript `/etc/init.d/i4l` übernommen.

Grundlage sind jeweils die in `rc.config` eingetragenen Parameter. Die Namensgebung für die dort verwendeten Variablen orientiert sich soweit wie möglich an den Optionen zu isdnctrl. Durch das Skript `/etc/init.d/route` wird das Routing auf die in `/etc/route.conf` eingetragenen Werte gesetzt.

Der Verbindungsaufbau geschieht bei Bedarf mit isdnctrl bzw. `/etc/init.d/i4l` und mit den in der `rc.config` festgelegten Parametern. Die Parameter können mit

```
erde: # isdnctrl list all
```

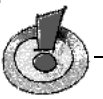
angezeigt werden. „Bei Bedarf“ bedeutet, dass eine der so entstandenen „Routen“ das entsprechende (ISDN-)Interface anspricht. Das kann durch jeden Benutzer und jede Applikation geschehen.

### 3.2.3 Hinweise zur Hardware

#### Voraussetzungen

Um unter SuSE Linux eine ISDN-Verbindung aufbauen zu können, brauchen Sie folgendes:

1. einen ISDN-Anschluss,
2. eine unterstützte ISDN-Karte,
3. ein installiertes SuSE Linux,
4. einen installierten Standard-Kernel von der SuSE Linux-CD und

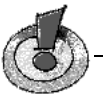


> **Hinweis**
>
> Sie brauchen *keinen* eigenen Kernel zu generieren – wenn Sie gleichwohl einen eigenen Kernel kompilieren wollen, nehmen Sie unbedingt die Quellen aus dem Paket `lx_suse`, Serie d!

5. das Paket `i4l`, Serie n.

Was Sie wissen müssen:

- ISDN-Karten-Typ,
- Einstellungen der Karte: IRQ, Portadresse etc. (je nach Typ) und
- welches ISDN-Protokoll Sie benutzen können:
  - 1TR6: (altes) nationales ISDN,
  - DSS1: Euro-ISDN,
  - Leased line: Standleitung.



> **Hinweis**
>
> Bei älteren großen TK-Anlagen wird oft *1TR6* auf dem internen S0-Bus gefahren.

#### Was ist die MSN/EAZ?

Bei Euro-ISDN ist MSN (engl. *Multiple Subscribe Number*) die Telefonnummer, allerdings ohne Vorwahl. Bei einem privaten Neuanschluss bekommen Sie meist drei unabhängige Nummern zugewiesen. Sie können sich eine beliebige davon für die ISDN-Verbindung auswählen, auch diejenige, welche Sie schon für eine Telefonverbindung benutzen, da anhand der ISDN-Dienstkennung der Typ einer Verbindung unterschieden werden kann.

Typischerweise wird die ISDN-Karte direkt an einen NTBA angeschlossen, es kann aber auch sinnvoll sein, über eine TK-Anlage einen weiteren S0-Bus bereitzustellen. Wenn Sie Euro-ISDN an einer TK-Anlage einsetzen, ist die MSN (meist) nur die Durchwahl auf der Anlage oder die `0`.

Bei 1TR6 wird anstatt der MSN eine EAZ (Endgeräte-Auswahl-Ziffer) benutzt – ansonsten ist MSN/EAZ synonym zu verwenden. Die EAZ ist eine einzelne Ziffer, die Sie auswählen können. Wählen Sie eine zwischen `1` und `9`. Verwenden Sie bei 1TR6 nicht die `0`!

### 3.2.4 Hardware mit YaST konfigurieren

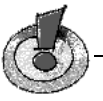
Der Treiber für die ISDN-Karte wird durch ein ladbares Kernelmodul bereitgestellt. Dafür muss das System nicht neu gebootet werden. Die üblichen ISDN-Karten werden durch den HiSax-Treiber unterstützt.

#### Schritt für Schritt

1. Loggen Sie sich als Benutzer ‘`root`’ ein.
2. Starten Sie YaST.
3. Wählen Sie das Menü ‘`Administration des Systems`’, ‘`Hardware in System integrieren`’, ‘`ISDN-Hardware konfigurieren`’ an. Diese Menüstruktur sehen Sie in Abbildung 3.1 auf der nächsten Seite.
4. Tragen Sie in der Maske bitte folgende Parameter ein:
   - **`I4L Starten`**
     Nur wenn dieses Feld aktiv ist, wird beim Booten ISDN konfiguriert. Sie können also hiermit steuern, ob überhaupt automatisch eine ISDN-Verbindung nach dem Booten aufgebaut werden kann.
   - **`ISDN-Protokoll`**
     Wählen Sie zwischen dem alten nationalen, deutschen ISDN (1TR6) oder dem heute üblichen Euro-ISDN (EDSS1) oder `Leased line`. Beachten Sie, dass bei Anschlüssen, die über eine TK-Anlage gehen, häufig 1TR6 gefahren wird.
   - **`Typ der ISDN-Karte`**
     Wählen Sie die vom HiSax-Treiber unterstützte Karte aus. Für PnP- und PCMCIA-Karten beachten Sie bitte die Datei `/usr/share/doc/packages/i4l/README.SuSE`.


Abbildung 3.1: Menüstruktur zur ISDN-Konfiguration mit YaST

- **`Interrupt`**
  **`Memory-Basisadresse`**
  **`IO-Port`**
  **`IO0-Wert`**
  **`IO1-Wert`**
  Diese Einstellmöglichkeiten werden nur für die Karten angezeigt, bei denen diese Werte einzustellen sind.



> **Hinweis**
>
> Bitte beachten Sie, dass bei PCI-Karten keine I/O-Adresse und kein Interrupt anzugeben ist. Folglich werden diese Punkte nicht angezeigt, wenn Sie eine solche Karte auswählen.

- **`Optionen zum Laden des ISDN-Moduls`**
  Lassen Sie dieses Feld bitte leer.

Weitere Information erhalten Sie durch Drücken der Hilfetaste (F1). Die Eingabemaske können Sie in Abbildung 3.2 auf der nächsten Seite sehen.

5. Betätigen Sie den Button ‘`Starten`’.

   Es wird testweise das Modul geladen. Im Fenster erkennen Sie, ob die Karte korrekt erkannt wurde.

   **Wenn OK:** Betätigen Sie den Button ‘`Speichern`’.

   Erklärung: Die Einstellungen werden dauerhaft (in Variablen in den Dateien `/etc/rc.config.d/i4l_*`) gespeichert, so dass sie nach dem nächsten Booten oder Wechsel des Runlevels wieder aktiviert werden können. Nach dem testweisen Laden des Moduls bleibt der Treiber geladen.

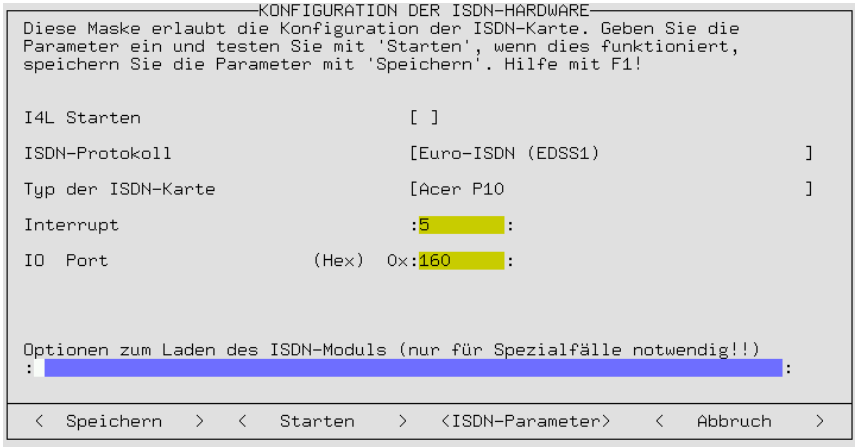



Abbildung 3.2: Eingabemaske zur ISDN-Konfiguration mit YaST

**Nicht OK:** Versuchen Sie andere Parameter, und betrachten Sie dabei die Veränderungen in der Datei `/var/log/messages` .

Übliche Probleme sind:

- Die IRQs 12 oder 15 sind bei einigen Mainboards nicht benutzbar.
- Die angegeben Adressen oder IRQs sind schon in Benutzung. Entfernen Sie alle Steckkarten, die vorerst nicht benötigt werden, z. B. Sound- und Netzwerkkarten.
- Das Modul ist schon geladen. Wechseln Sie auf eine andere Konsole und kontrollieren Sie mit **`lsmod`**. Geben Sie den folgenden Befehl zum Entladen des Moduls ein:
  `erde: # rmmod hisax`
- Sie haben eine PnP-Karte. Lesen Sie dazu in der Datei `/usr/share/doc/packages/i4l/README.SuSE` nach.
- Sie haben keine vom HiSax-, sondern von einem anderen Treiber unterstützte Karte (z. B. ICN, AVM-B1). Lesen Sie dazu bitte in der Datei `/usr/share/doc/packages/i4l/README.SuSE` nach.

6. Beenden Sie YaST.

7. Konfigurieren Sie isdnlog.

   Bevor die Module geladen werden, sollte der **`isdnlog`** konfiguriert werden. Dieser protokolliert alle Aktivitäten auf dem S0-Bus.

   Passen Sie die folgenden Dateien Ihren Gegebenheiten an:

   - `/etc/isdn/isdn.conf` :

     Zuerst wird das Land spezifiziert, in dem `isdn4linux` eingesetzt wird. Für Deutschland müssen die Werte wie in Datei 3.2.1 auf der nächsten Seite gesetzt werden. Für Österreich ist der Countrycode 43.

```
# /etc/isdn/isdn.conf

[GLOBAL]
COUNTRYPREFIX = +
COUNTRYCODE = 49
AREAPREFIX = 0
```

Datei 3.2.1: `/etc/isdn/isdn.conf`

Ebenfalls im `GLOBAL`-Abschnitt wird der `AREACODE` (die Vorwahl) ohne führende Null angegeben. Wenn Ihre Vorwahl z. B. `0911` ist, wird `AREACODE = 911` eingetragen.

Dies ist (in Deutschland) der einzige Teil, der angepasst werden muss.

Mit `CHARGEMAX = 20.00` können Sie angeben, wie viel Geld (in DM) maximal pro Tag vertelefoniert werden darf. Dies schützt vor unerwünschten Connects. Aber verlassen Sie sich nicht auf dieses automatische Feature!

- `/etc/isdn/callerid.conf`:
  Hier können Sie alle bekannten Telefonnummern eintragen. In der Datei `/var/log/messages` und durch **`isdnrep`** werden dann die Namen anstatt der Telefonnummer angezeigt.
  Vgl. das Beispiel in Datei 3.2.2. Ihre eigene Nummer ist `4711` und die Ihres Providers ist `4712`.

```
# /etc/isdn/callerid.conf

[MSN]
NUMBER = 4711
SI = 1
ALIAS = ich
ZONE = 1

[MSN]
NUMBER = 4712
SI = 1
ALIAS = Provider
ZONE = 1
```

Datei 3.2.2: `/etc/isdn/callerid.conf`

- `/etc/isdn/isdnlog.isdnctrl0.options`:
  Hier können Sie Optionen für **`isdnlog`** eingeben. Dies ist normalerweise nicht nötig.

8. Geben Sie die Befehle

   ```
   erde: # init 1
   erde: # init 2
   ```

   ein, um u. a. die Netzwerkdienste neu zu starten, oder aktivieren Sie ISDN erneut mit YaST (oder booten Sie neu).

### 3.2.5 Internet-Anbindung einrichten

#### ISDN-Konfiguration für Ihren Internetprovider

**Die Protokollwahl**

Für den ISDN-Zugang gibt es drei wichtige ISDN-Protokolle:

- syncPPP,
- rawip-HDLC und
- Terminal-Login mit X.75 .

Meist verwenden die Internetprovider das Protokoll syncPPP. Sie sollten Ihr Linux also damit konfigurieren.

**Voraussetzungen**

- Die ISDN-Hardwarekonfiguration hat funktioniert.
- Der ISDN-Treiber ist geladen.
- Sie wissen die von Ihnen zu verwendende MSN oder EAZ.
- Sie wissen das von Ihrem Provider verwendete Protokoll (syncPPP, rawip).
- Sie wissen die Zugangstelefonnummer.
- Sie wissen den Benutzernamen und das Passwort.
- Sie wissen den Domain Name Service (DNS) („Nameserver“) des Providers.

Erst wenn alle oben genannten Punkte erfüllt sind, können Sie den Internetzugang erfolgreich einrichten. Das folgende Beispiel beschreibt den syncPPP-Zugang, rawip ist aber im Wesentlichen genauso – nur einfacher.

**Schritt für Schritt**

1. Starten Sie YaST und wechseln Sie in das Menü ‘`Administration des Systems`’, ‘`Netzwerk konfigurieren`’, ‘`Netzwerk-Grundkonfiguration`’. Die nun erscheinende Eingabemaske sehen Sie in Abbildung 3.3 auf der nächsten Seite.
2. Wählen Sie eine freie Nummer, z. B. `1`.
3. Wählen Sie durch Drücken von F5 als Device ‘`ISDN SyncPPP`’ aus.
4. Drücken Sie F6 (‘`IP-Adresse`’) und geben Sie die Standard-IP-Adressen ein:
   - IP-Adresse Ihres Rechners (ISDN-Karte): `192.168.0.99`
   - Kreuzen Sie ‘`Dynamische IP-Adresse`’ *nur* an, wenn Sie eine Adresse von Ihrem Provider (ISP) bei jeder Verbindung „dynamisch“ zugewiesen bekommen.
   - IP-Adresse des Default-Gateways: `192.168.0.1`


Abbildung 3.3: Netzwerkkonfiguration mit YaST

- IP-Adresse des Point-To-Point-Partners: `192.168.0.1`

5. Verlassen Sie die Eingabemaske durch Betätigen des Buttons ‘`Weiter`’.
6. Aktivieren Sie das Netzwerk-Device mit (F4), falls nicht schon geschehen.
7. Mit (F8) (‘`ISDN`’) können Sie jetzt weitere ISDN-spezifische Parameter angeben. Dies können Sie in Abbildung 3.4 sehen.

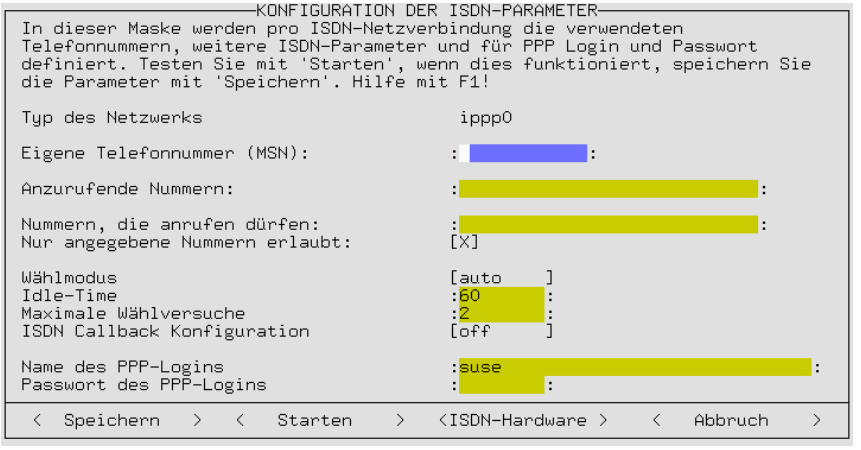



Abbildung 3.4: Konfiguration der ISDN-Parameter mit YaST

Geben Sie dabei bitte die folgenden Werte an:

- **`Eigene Telefonnummer (MSN)`**
  Ihre eigene MSN, z. B. 123456 .

- **Anzurufende Nummer:** 012345678

  Erklärung: Die Nummer, die angerufen werden soll. 012345678 ist die Nummer Ihres Internet-Providers.

> Hinweis
>
> Bei TK-Anlagen müssen Sie eventuell eine zusätzliche 0 vorwählen. – Bitte beachten Sie weiterhin, dass durch Leerzeichen getrennte Telefonnummern wie zwei *unterschiedliche* Telefonnummern behandelt werden.

- **Nummern, die anrufen dürfen:**

  Nur für Dialin-Server nötig.

- **Nur angegebene Nummern erlaubt:**

  Setzen Sie dieses Flag, damit niemand unerlaubt eine Verbindung zu Ihrem System aufbauen kann.

- **Wählmodus:**

  Mit auto werden Verbindungen automatisch aufgebaut, wenn versucht wird, auf Adressen zuzugreifen, die nur über die ISDN-Schnittstelle zu erreichen sind. Mit der Einstellung manual ist es notwendig, bei Bedarf die Verbindung *per Hand* herzustellen. Dafür benötigen Sie folgende Shell-Kommandos:
  Verbindung herstellen: **isdnctrl dial DEVICE-NAME**
  Verbindung beenden: **isdnctrl hangup DEVICE-NAME**.
  Beispiele mit Pfadangabe des Befehls **isdnctrl** sähen dann z. B. so aus: **/usr/sbin/isdnctrl dial ippp0** und entsprechend **/usr/sbin/isdnctrl hangup ippp0**. Bei off ist es gar nicht möglich, Verbindungen über diese ISDN-Schnittstelle aufzubauen.

- **Idle-Time:**

  Zeit, nach der automatisch aufgelegt wird, wenn keine Internet-Pakete über die ISDN-Leitung übertragen werden.

- **Maximale Wählversuche:**

  Anzahl der Wiederholversuche festlegen.

- **ISDN Callback Konfiguration:**

  off, out oder in.

- **Name des PPP-Logins:**

  Geben Sie hier den Benutzernamen für Ihren Provider an.

- **Passwort des PPP-Logins:**

  Geben Sie hier das Passwort für Ihren Provider an. Das Passwort wird bei der Eingabe nur durch Sterne angedeutet. Es wird in der Datei /etc/ppp/pap-secrets gespeichert.

Mit (F1) erhalten Sie weitere Hilfe.

8. Betätigen Sie den Button ‘Starten’.

   Erklärung: Es wird testweise das Netzwerk konfiguriert. Im Fenster erkennen Sie, ob es funktioniert hat. Hier sollte es keine Probleme geben.

   **Wenn OK:** Betätigen Sie den Button ‘Speichern’.

   Erklärung: Die Einstellungen werden dauerhaft (in Variablen in den Dateien `/etc/rc.config.d/i4l_*`) gespeichert, so dass sie nach dem nächsten Booten oder Wechsel des Runlevels wieder aktiviert werden können. Nach dem testweisen Starten bleiben die Einstellungen aber erhalten.

   **Wenn nicht OK:** Vermutlich sind dann die ISDN-Module nicht geladen. Beachten Sie außerdem die Meldungen in `/var/log/messages` .

9. Gehen Sie in YaST in das Menü ‘Konfiguration Nameserver’ und beantworten Sie die Frage mit **Ja**. Geben Sie hier die IP-Nummer des Nameservers (DNS) Ihres Providers an. Wenn Sie die IP-Nummer nicht wissen, gehen Sie bitte gemäß den folgenden Hinweisen auf Seite 86 vor.

10. Vermutlich hat der Verbindungsaufbau funktioniert. Dann wechseln Sie wieder in YaST, betätigen den Button ‘Speichern’ und beenden YaST.

### Kanalbündelung (MPPP) über syncPPP

Bei ISDN gibt es die Möglichkeit, zwei oder mehr so genannte physische B-Kanäle zu einem logischen Kanal zusammenzufügen. Dadurch hat man dann eine doppelte (oder mehrfache) Transferrate.

**Konfiguration** Beispiel: Auf dem Device ippp0 ist eine laufende syncPPP-Verbindung konfiguriert. Diese soll jetzt erweitert werden, so dass bei Bedarf ein zweiter oder dritter etc. Kanal dazugeschaltet werden kann.

#### Vorgehensweise

1. Prüfen Sie, welche Devices für syncPPP schon verwendet werden. Wählen Sie z. B. ippp1 als Slave (er darf jedoch nicht schon benutzt werden).

2. In `/etc/rc.config` suchen Sie nach der Konfiguration für das Device ippp0 (Beispiel: **`NETDEV_1="ippp0"`**, hier ist die Extension **`_1`** wichtig).

3. In `/etc/rc.config.d/i4l.rc.config` ändern Sie den Wert für **`I4L_SLAVE_1`** auf **`I4L_SLAVE_1="ippp1"`**

4. In `/etc/ppp/options.ippp0` erweitern Sie:

   ```
   # The device(s)
   # for more than one device try:
   # /dev/ippp0 /dev/ippp1 ...
   /dev/ippp0
   ```

   zu:

```
# The device(s)
# for more than one device try:
# /dev/ippp0 /dev/ippp1 ...
/dev/ippp0 /dev/ippp1

+mp
```

5. Starten Sie den ipppd neu, z. B. durch ein Reboot oder durch:

```
init 1
rci4l restart
```

Erklärung: Es wird ein Pseudo-Device angelegt (ippp1), welches nur im i4l-Subsystem bzw. ipppd bekannt ist. Aus der Sicht des Netzwerks ist es nicht vorhanden. Die notwendigen **`isdnctrl`**-Befehle werden durch das Script i4l (siehe unten) automatisch angelegt, wenn die Variable **`I4L_SLAVE`** wie oben gesetzt wird.

**Benutzung** Zunächst muss das Slave-Device in den `dialmode auto` gebracht werden (siehe auch unten): Das geschieht seit SuSE Linux 6.4 automatisch.

```
isdnctrl dialmode ippp1 auto
```

- Mit den folgenden Programmen kann der zweite Kanal manuell hoch- und runtergefahren werden:

  Starten: **`isdnctrl addlink ippp0`**

  Stoppen: **`isdnctrl removelink ippp0`**

- Es gibt die Tools **`xibod`** und **`ibod`**, die bei Last automatisch den zweiten Kanal dazuschalten und auch wieder wegnehmen. **`xibod`** ist auf der Distribution in der Serie `n` enthalten.

  Siehe: `http://www.compound.se/ibod.html`

**Dial-In-Server**

Ein Dialin-Server kann genauso für MPPP erweitert werden.

**Bündelung von mehr als 2 Kanälen**

Einfach eine Liste mit device Namen angeben:

```
I4L_SLAVE_1="ippp1 ippp2 ippp3"
```

und in der `option.ippp0` :

```
# The device(s)
# for more than one device try:
# /dev/ippp0 /dev/ippp1 ...
/dev/ippp0 /dev/ippp1 /dev/ippp2 /dev/ippp3
```

Beachten Sie bitte, dass erst die **`ibod/xibod`**-Versionen ab SuSE Linux 7.2 mehr als 2 Kanäle und mehrfache Bündel unterstützen.

Vgl. auch die Hinweise in:

`file:/usr/share/doc/sdb/de/html/i4lmppp.html`

### Dynamische IP-Nummer bei syncPPP

Im Fall von dynamischen IP-Adressen dienen die vergebenen Dummy-Adressen aus dem privaten Bereich nur als Platzhalter bis zum Verbindungsaufbau.

**IP-Nummer des Nameservers des Providers**

Wenn Sie die IP-Nummer nicht wissen, müssen Sie diese bei Ihrem Provider erfragen. Oder Sie bitten jemanden mit Internet-Zugang um Eingabe des folgenden Befehls in einem Terminalfenster (als Beispiel sei hier T-Online angeführt):

```
whois t-online.de
```

Sie erhalten dann eine Ausgabe, die unter anderem folgende Zeilen wie in Ausgabe 3.2.1 enthält; **whois** ist im Paket `whois` enthalten.

```
domain:      t-online.de
descr:       T-Online International AG
descr:       Waldstrasse 3
descr:       D-64331 Weiterstadt
descr:       DE
admin-c:     DK162-RIPE
tech-c:      HD1710-RIPE
zone-c:      HD1710-RIPE
nserver:     dns00.btx.dtag.de
nserver:     pns.dtag.de
nserver:     techfac.techfak.uni-bielefeld.de
status:      connect
changed:     lastchange@denic.de 20000411
source:      DENIC
```

Ausgabe 3.2.1: Ausgabe von **whois t-online.de**

In der Zeile **nserver:** sehen Sie den Nameserver des Providers. Dann brauchen Sie nur noch die IP-Adresse des Namens. Dazu gibt man folgenden Befehl ein:

```
host dns00.btx.dtag.de
```

Dann erfolgt z. B. eine Antwort wie in Ausgabe 3.2.2.

```
dns00.btx.dtag.de has address 194.25.2.132
```

Ausgabe 3.2.2: Ausgabe von **host**

Dies (`194.25.2.132`) wäre dann die IP-Adresse des Nameservers von T-Online (es gibt noch `194.25.0.125` und `129.70.132.100`).

### Mögliche Probleme

Falls der Verbindungsaufbau nicht klappt:

- Prüfen Sie `/var/log/messages` auf „verdächtige“ Ausgaben.
- Versuchen Sie auch den rawip-Zugang.
- Ist die MSN/EAZ richtig eingestellt?
- Müssen Sie eventuell eine `0` vorwählen?

#### Weitere Informationen

Weitere Informationen, wie Sie eine ISDN-Verbindung und Ihr ISDN-Subsystem konfigurieren, finden Sie in folgenden Quellen:

- Datei `/usr/share/doc/packages/i4l/README.SuSE`
- Support-Datenbank: `http://sdb.suse.de/sdb/de/html/`(online) oder über die SuSE-Hilfe (Aufruf mit **`hilfe`** oder aus dem Menü), wenn Sie die aus der Serie `doc` die Pakete `susehilf` und `sdb_de` installiert haben.
- Im Paket `i4l` (z. B. die ISDN-FAQ in der Datei: `/usr/share/doc/packages/i4ldoc/i4l-faq-de.txt` bzw. als HTML-Dokument: `file:/usr/share/doc/packages/i4ldoc/i4l-faq-de.html`)
- Mailingliste *ISDN mit SuSE Linux* (deutsch): suse-isdn@suse.com und die Newsgroup **`de.alt.comm.isdn4linux`**.

### 3.2.6 ISDN-Meldungen – “cause codes”

Leider sind die „Cause“-Meldungen, die man vom ISDN Subsystem erhält, auf Englisch und nicht immer leicht verständlich. Daher hier die Übersetzung.

Eine typische „Fehler“-Meldung (engl. *Cause*) von HiSaX besteht aus 2 Teilen, der **`location`** und dem **`cause code`**. Sie besteht im Falle von Euro-ISDN aus 5 Zeichen, **`Exxyy`** wobei `xx` die Quelle der Fehlermeldung (hier nicht erläutert) und `yy` die Meldungsursache angibt. Diese Ausgabe macht HiSaX immer hexadezimal. Manche Meldungen sind auch kein Fehler in diesem Sinne, sondern stellen normales Verhalten einer Telefonverbindung dar („besetzt“, „Verbindung durch Auflegen beendet“).

Im Folgenden die Erläuterung der verschiedenen Meldungsursachen, den „Causes“; mit freundlicher Genehmigung des Instituts für Elektronische Systeme und Vermittlungstechnik der Universität Dortmund: `http://www-esv.e-technik.uni-dortmund.de`. Beachten Sie bitte, dass HiSaX diesen „Cause“-Wert Hexadezimal ausgibt.

| **`Cause#`** `dez/hex` | Beschreibung |
|---|---|
| | **Gruppe 0/1 normale Gründe** |
| **`#1 01`** | Die Nummer des gerufenen Teilnehmers ist zwar komplett und kann durch das Netzwerk interpretiert werden, ist aber zur Zeit keiner Endstelle zugeordnet. |
| **`#2 02`** | Das spezifizierte Transitnetzwerk wird durch die meldende Stelle nicht erkannt. Dies kann entweder geschehen, weil das gewünschte Transitnetzwerk nicht existiert oder aber den geforderten Dienst ablehnt. |

Tabelle 3.1: Fortsetzung auf der nächsten Seite...

| | |
|---|---|
| **#3 03** | Es wurde kein Weg zum gewünschten Endteilnehmer gefunden, da dieser vermutlich an einem anderen als dem gewählten Netzwerk angeschlossen ist. |
| **#6 06** | Der gerufene Teilnehmer kann den geforderten Kanal nicht verwenden. |
| **#7 07** | Der Ruf wurde beim gerufenen Teilnehmer abgewiesen, da der geforderte Kanal bereits belegt war (virtueller Kanal, X.31 bzw. X.25). |
| **#16 10** | Dieser Grund wird verwendet, wenn einer der an dem Ruf beteiligten Endteilnehmer den Ruf beendet. |
| **#17 11** | Der Anschluss des gerufenen Teilnehmers ist besetzt und dieser nicht in der Lage, auf einen weiteren Ruf zu reagieren oder diesen anzunehmen. |
| **#18 12** | Der gerufene Anschluss wurde zwar erreicht, aber der Rufaufbauwunsch `SETUP` wurde nicht innerhalb der vorgesehenen Zeit beantwortet. |
| **#19 13** | Trotz Annahme des Endgerätes wurde der Ruf nicht durch den Endteilnehmer akzeptiert, z. B. Telefon klingelt in leerer Wohnung. |
| **#21 15** | Der gerufene Teilnehmer hat den Ruf explizit abgelehnt, z. B. als Reaktion auf einen Anklopfton. |
| **#22 16** | Als Option der Zielvermittlungsstelle kann dieser Grund gesendet werden, wenn sich die Rufnummer des gerufenen Teilnehmers geändert hat. |
| **#26 1A** | Dem gerufenen Teilnehmer konnte der Ruf nicht angezeigt werden. |
| **#27 1B** | Die gerufene Teilnehmerschnittstelle (Anschluss) ist zur Zeit außer Betrieb. |
| **#28 1C** | Die gewählte Rufnummer ist ungültig oder kann durch das Netzwerk nicht interpretiert werden. |
| **#29 1D** | Ein mit dem Rufaufbau angefordertes Dienstmerkmal kann durch das Netzwerk nicht bereitgestellt werden. |
| **#30 1E** | Es wird angezeigt, dass die STATUS Message, in der dieser Cause vorkommt, auf Grund einer STATUS ENQUIRY Message versendet wurde. |
| **#31 1F** | Wenn kein anderer der in der Klasse `Normal` vorhandenen Gründe für die Ursache des Rufabbaus anwendbar ist, wird dieser Grund gesendet. |
| | **Gruppe 2, nicht verfügbare Ressourcen** |
| **#34 22** | In der Vermittlungsstelle sind alle B-Kanäle (Sprechwege) oder alle virtuellen Kanäle (X.25) belegt. |
| **#38 26** | Das Vermittlungsnetzwerk ist nicht betriebsbereit und wird dies für eine absehbare Zeit auch nicht mehr sein. |

Tabelle 3.1: Fortsetzung auf der nächsten Seite...

| | |
|---|---|
| **#41 29** | In der Vermittlungsstelle liegt ein vorübergehender Fehler vor, der in nächster Zukunft behoben sein wird. Es macht also Sinn, den Rufaufbauversuch zu wiederholen. |
| **#42 2A** | In der Vermittlungsstelle, die diesen Grund absendet, ist zur Zeit aus Überlastgründen kein Kanal verfügbar. |
| **#43 2B** | Die durch den rufenden Teilnehmer übergebene Zugriffsinformation, wie z. B. Passwörter im UTU Element, LLC oder HLC Daten, konnten nicht an den gerufenen Teilnehmer weitergeleitet werden. |
| **#44 2C** | Der gewünschte Kanal kann durch das Interface auf der anderen Seite nicht bereitgestellt werden. |
| **#47 2F** | Sollte eine Ressource nicht verfügbar sein, die nicht durch die oben genannten Gründe der Gruppe `Ressource nicht verfügbar` beschrieben werden kann, so wird dieser Grund gesendet. |
| | **Gruppe 3, Dienst oder Option nicht verfügbar** |
| **#49 31** | Das geforderte Qualitätsmerkmal (Durchsatz oder Delay) nach X.213, kann nicht eingehalten werden. |
| **#50 32** | Der Anwender ist zur Nutzung des angeforderten Dienstes nicht berechtigt, da er als Nutzer nicht eingetragen ist. |
| **#57 39** | Der Anwender ist auf der auslösenden Anlage nicht berechtigt, den geforderten Dienst zu nutzen. |
| **#58 3A** | Der verlangte Übertragungsdienst ist zur Zeit nicht verfügbar. |
| **#59 3B** | Eine nicht verfügbare Dienstleistung oder Option, die nicht durch die vorherigen Gründe zu beschreiben ist, wird hiermit angezeigt. |
| | **Gruppe 4, Dienstleistung oder Option nicht implementiert** |
| **#65 41** | Das aussendende Gerät ist nicht in der Lage, die geforderte Eigenschaft (`bearer capabiltiy`) bereitzustellen. |
| **#66 42** | Der angeforderte Kanaltyp ist nicht verfügbar. |
| **#69 45** | Das angeforderte Dienstmerkmal ist nicht implementiert. |
| **#70 46** | Der Benutzer hat die uneingeschränkte Übertragung digitaler Information angefordert, aber nur eine eingeschränkte Übertragung ist zulässig. |

Tabelle 3.1: Fortsetzung auf der nächsten Seite...

| | |
|---|---|
| **#79 4F** | Ein Dienst oder eine Option, die sich durch die oben genannten Gründe nicht beschreiben lässt, ist nicht implementiert. |
| | **Gruppe 5, Ungültige Nachricht, unzulässiger Parameterbereich** |
| **#81 51** | Eine Nachricht wurde empfangen, die mit einem im Netzwerk zur Zeit ungültigen „Call Reference“-Wert versehen war. |
| **#82 52** | Der angeforderte Nutzkanal existiert auf der Schnittstelle, die diesen Grund liefert, nicht. Dies kann z. B. bei CHI = 26 auf einem PCM 24 Interface vorkommen. |
| **#83 53** | Es wurde versucht, einen Ruf mit einer ungültigen Call ID aus dem geparkten Zustand herauszuholen. |
| **#84 54** | Es wurde versucht, beim Übergang ins Parken eine Call ID zu verwenden, die im Bereich des zuständigen Controllers bereits in Verwendung ist. |
| **#85 55** | Es wurde versucht, einen Ruf wieder aufzunehmen, obwohl kein Ruf geparkt worden ist. |
| **#86 56** | Der Ruf mit der verlangten Call ID wurde bereits wieder ausgelöst. |
| **#88 58** | Die gerufene Endteilnehmerschnittstelle ist nicht in der Lage, den geforderten LLC oder HLC oder anderen zusätzlichen Attributen zu genügen. |
| **#91 5B** | Ein Transit Netzwerk wurde in einer inkompatiblen Weise angefordert. |
| **#95 5F** | Wenn keine andere der hier genannten Ursachen für eine ungültige Message zutrifft, wird dieser Grund versendet. |
| | **Gruppe 6, Protokollfehler** |
| **#96 60** | Ein zwingend vorgeschriebenes Informationselement ist nicht vorhanden. |
| **#97 61** | Ein unbekannter oder nicht implementierter Message-Type wurde von der auslösenden Einheit empfangen. |
| **#98 62** | Eine Message wurde empfangen, die im aktuellen Zustand des Rufes nicht zulässig war, oder es wurde eine STATUS-Message mit einem ungültigen Zustand empfangen. |
| **#99 63** | Ein Informationselement wurde empfangen, das nicht bekannt oder nicht implementiert ist. Das Informationselement kann bei einem weiteren Versuch weggelassen werden, um die gewünschte Funktion zu erreichen. |

Tabelle 3.1: Fortsetzung auf der nächsten Seite...

| | |
|---|---|
| **#100 64** | Der Inhalt eines Informationselementes ist ungültig und kann vom den Grund sendenden Gerät nicht verwendet werden. |
| **#101 65** | Eine für den aktuellen Zustand des Rufes unzulässige Message wurde empfangen. |
| **#102 66** | Eine Prozedur zur Wiederherstellung eines stabilen Zustandes wurde als Reaktion auf das Ablaufen eines Timers eingeleitet. |
| **#111 6F** | Eine in dieser Gruppe nicht näher spezifizierte Ursache ist aufgetreten. |

Tabelle 3.1: Deutschsprachige ISDN-Causes

## 3.3 Konfiguration eines ADSL / T-DSL Anschlusses

### 3.3.1 Standardkonfiguration

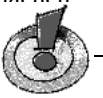
Momentan werden von SuSE Linux DSL-Zugänge unterstützt, die mit dem Point-to-Point-over-Ethernet-Protokoll (PPPoE) arbeiten. Dieser Service wird zur Zeit von T-Online und Arcor angeboten.

Wenn Sie ein Einzelplatzsystem mit grafischer Oberfläche besitzen, sollte der DSL-Zugang mit YaST2, Modul ADSL, bzw. T-DSL für T-Online Kunden, eingerichtet werden. Falls dies nicht möglich oder gewünscht ist, gehen Sie bitte folgendermaßen vor:

1. Die Pakete ppp und pppoe aus der Serie n müssen installiert sein. Verwenden Sie dazu am besten YaST.
2. Konfigurieren Sie Ihre Netzwerkkarte, z. B. mit YaST:
   a) Wählen Sie das Menü ‘`Administration des Systems`’ → ‘`Hardware in System integrieren`’ → ‘`Netzwerkkarte konfigurieren`’.
   b) Im Feld ‘`Typ des Netzwerks`’ geben Sie Ihr Netzwerkdevice an (meist eth0).
   c) In der Zeile ‘`Art der Netzwerk-Karte`’ können Sie Ihre Netzwerkkarte auswählen.
   d) Gehen Sie in das Menü ‘`Administration des Systems`’ → ‘`Netzwerk konfigurieren`’ → ‘`Netzwerk Grundkonfiguration`’.
   e) Wählen Sie Ihr Device (z. B. `eth0`) aus und drücken Sie (F6).
   f) Geben Sie in dem Feld ‘`IP-Adresse Ihres Rechners`’ eine IP-Adresse ein, die jedoch nicht im Internet oder in einem privaten Netz vergeben sein darf. Ein gute Wahl ist z. B. `192.168.22.1`.

g) Im Feld ‘Netmask’ tragen Sie 255.255.255.0 ein.

h) Sie müssen unbedingt darauf achten, dass Sie für ein Einzelplatzsystem keinen Eintrag in das Feld ‘Adresse default-Gateway’ machen.



> **Hinweis**
>
> Die Werte für die IP-Adresse Ihres Rechners und Netzmaske sind nur Platzhalter. Sie haben für den Verbindungsaufbau mit ADSL keine Bedeutung und werden nur zum Aktivieren der Netzwerkkarte benötigt.

i) Klicken Sie auf ‘Weiter’ und speichern Sie mit F10.

j) Gehen Sie in das Menü ‘Konfiguration Nameserver’ und tragen Sie hier einen beliebigen Servernamen ein. Die meisten Provider unterstützen heute dynamische DNS-Vergabe, d. h. beim Verbindungsaufbau werden aktuelle IP-Adressen der Nameserver übergeben. Dennoch muss in Ihrem Einzelplatzsystem in diesem Dialog „DNS-Server“ oder wenigstens ein Platzhalter eingetragen werden. Gut geeignet als Platzhalter ist z. B. 192.168.22.99 .

   Falls Sie den Nameserver nicht dynamisch zugewiesen bekommen, müssen Sie hier die gültigen IP-Adressen des Nameservers Ihres Providers eintragen. Außerdem können noch Werte für die Liste der Domains definiert werden.

k) Beenden Sie die Grundkonfiguration des Netzwerks.

3. Die ADSL-Parameter werden in zwei Dateien definiert: /etc/pppoed.conf und rc.dialout , wenn Sie KDE verwenden.

   Gehen Sie folgendermaßen vor:

a) Öffnen Sie die Datei /etc/pppoed.conf mit Ihrem Lieblingseditor:

b) Tragen Sie im Parameter user Ihren Loginnamen (Usernamen) ein, z. B.:

```
user = myname
```

   Oft wird an den Usernamen ähnlich einer E-Mail-Adresse noch ein @-Zeichen und der Domainname des Providers angehängt, z. B.:

```
user = myname@myisp.de
```

   Dies ist z. B. bei T-Online der Fall, bei Arcor hingegen nicht.

c) Der Parameter password muss Ihr Kennwort beinhalten, z. B.:

```
password = obhun123
```

d) In der Zeile interface tragen Sie das Device ein, das Sie bei der Konfiguration der Ethernetkarte definiert haben, z. B.:

```
interface = eth0
```

e) Falls Sie die Verbindung mit kinternet durch Mausklick herstellen wollen, öffnen Sie noch die Datei rc.dialout :

i. Tragen Sie folgende Zeile ein:
`MODE = ADSL`

ii. Beim Parameter `DEVICE` tragen Sie auch wieder das Interface ein, das Sie bei der Konfiguration der Ethernetkarte definiert haben, z. B.:
`DEVICE = eth0`

4. Starten Sie das Netzwerk oder den Rechner neu.

5. Sie können die Verbindung nun starten. Klicken Sie auf das Zahnrad-Icon in der Buttonleiste. Wählen Sie '`Kommunikation / Internet`' → '`Internet Tools`' → '`kinternet`'. Nun erscheint in der Buttonleiste das Steckersymbol. Ein Klick darauf startet die Verbindung und ein zweiter Klick beendet sie wieder.

6. Alternativ können Sie auch die Verbindung mit dem Aufruf von **`rcpppoed start`** herstellen und mit **`rcpppoed stop`** wieder beenden.

### 3.3.2 DSL Verbindung per Dial-on-Demand

Dial-on-Demand bedeutet, dass die Verbindung automatisch aufgebaut wird, sobald ein User auf das Internet zugreift, z. B. indem er eine Webseite mit einem Browser anwählt oder E-Mails verschickt. Nach einer bestimmten Zeit (Idletime), in der keine Daten gesendet oder empfangen werden, wird die Verbindung wieder getrennt. Da die Einwahl mit PPPoE, dem Protokoll für ADSL, sehr schnell geht, entsteht fast der Eindruck, als hätte man eine Standleitung in das Internet.

Dies ist aber nur sinnvoll, wenn Sie eine so genannte Flatrate besitzen. Wird Ihr Zugang zeitabhängig abgerechnet, müssen Sie darauf achten, dass kein periodischer Prozess, z. B. ein cronjob, immer wieder eine Verbindung aufbaut. Das könnte sehr teuer werden.

Obwohl mit einer DSL-Flatrate auch eine permanente Einwahl möglich wäre, sprechen doch einige Punkte für eine Verbindung, die nur kurz und nach Bedarf besteht:

- Die meisten Provider trennen die Verbindung nach einer gewissen Zeit.
- Eine permanente Verbindung kann als Ressourcenverschwendung betrachtet werden (z. B. IP-Adressen).
- Vor allem ist es ein enormes Sicherheitsrisiko permanent online zu sein, da ein Angreifer den Router systematisch auf Schwachstellen absuchen kann. Ein System, das nur bei Bedarf im Internet erreichbar ist und immer wieder eine andere IP-Adresse hat, ist viel schwieriger zu attackieren.

Dial-on-Demand können Sie mit YaST2 aktivieren (siehe auch das Buch *Konfiguration*), oder Sie richten es manuell ein:

1. Sie setzen in der Datei `/etc/pppoed.conf` den Parameter
`demand = yes`

und definieren eine Idletime mit folgender Variablen:

```
idle = 60
```

Damit wird eine unbenutzte Verbindung nach 60 Sekunden beendet.

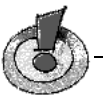
2. In der gleichen Datei haben Sie die Möglichkeit, zwei DNS-Server-Einträge zu definieren. Diese werden dann beim Start des DoD-Systems aktiv:

```
dns1 = 192.168.22.98
dns2 = 192.168.22.99
```

3. Im Prinzip können Sie die Dial-on-Demand-Verbindung nun schon nutzen. Sie müssen jedoch bei jedem Systemstart als User `'root'` das DoD-System mit **`rcpppoed start`** aktivieren. Automatisieren können Sie die Aktivierung des DoD-Systems, indem Sie in der Datei `/etc/rc.config` die Einstellungsvariable **`START_PPPOED=yes`** setzen.

### 3.3.3 DSL-Router einrichten

Erfahrene Linux-Anwender können mit der mitgelieferten Software einen DSL-Router für ein kleines privates Netzwerk einrichten. Dies ist mit relativ geringem Aufwand möglich. Achten Sie bitte unbedingt darauf, ob der Vertrag mit Ihrem Provider dies nicht ausdrücklich verbietet. Eine detaillierte Anweisung finden Sie in unserer Supportdatenbank (`http://sdb.suse.de/sdb/de/html`) unter dem Stichwort `dsl router`.



**Hinweis**

Ein Router in das Internet ist immer eine Sicherheitslücke für Ihr privates Netz. Achten Sie darauf, dass Sie keine sensiblen Daten in Ihrem Netz gespeichert haben. SuSE kann keine Gewährleistung für Schäden übernehmen, die aufgrund des Gebrauchs eines Gateways an Ihren Daten oder lokalem Netzwerk entstehen.

Hier ein Beispiel für eine `/etc/pppoed.conf` :

```
# user and password
user = "000111111112222233344556699@t-online.de"
password = "pssst"

idle = 60
demand = yes

dns1 =
dns2 =
```

Datei 3.3.1: Die Datei `/etc/pppoed.conf`

# 3.4 Kabelmodem

In einigen europäischen Ländern sowie USA und Kanada ist der Internetzugang über das Fernsehkabelnetz weit verbreitet. Hier folgt als Beispiel eine Schritt-für-Schritt-Anleitung, um mit dem österreichischen Telekabel-Dienst in das Internet zu kommen. Diese Anleitung sollte sich auch auf andere Kabelanbieter übertragen lassen.

## 3.4.1 Grundlagen

Der Telekabel-Teilnehmer bekommt von der Kabelfirma ein „Modem“, welches einerseits an das Fernsehkabel, andererseits mittels 10Base-T-Leitung (Twisted-Pair) an eine Netzwerkkarte im Computer angeschlossen wird. Dieses Modem stellt dann für den Computer eine Standleitung dar, meist mit einer festen IP-Adresse.

#### Vorgehensweise zur Installation

1. Falls Sie Ihre Netzwerkkarte schon eingerichtet haben, fahren Sie bei Punkt 8 fort.
2. Starten Sie als Benutzer `'root'` das Programm YaST – falls Sie in KDE sind: (Alt) + (F2) drücken, dann **`xterm`** eingeben; dann im neuen Fenster YaST starten.
3. Gehen Sie zu ‘`Administration des Systems`’, ‘`Hardware in System integrieren`’, ‘`Netzwerkkarte konfigurieren`’.
4. Bei ‘`Typ des Netzwerks`’ geben Sie `eth0` an.
5. Bei ‘`Art der Netzwerkkarte`’ wählen Sie Ihre Karte aus.
6. Bei ‘`Optionen zum Laden des Moduls`’ geben Sie Parameter wie IO-Port usw. an. *Achtung,* falls Sie eine PCI-Karte haben, brauchen Sie meist keine Parameter angeben.
7. Gehen Sie auf ‘`Weiter`’. Kehren Sie in das YaST-Hauptmenü zurück (durch zweimaliges Drücken der (Esc)-Taste).
8. Gehen Sie zu ‘`Administration des Systems`’, ‘`Netzwerk konfigurieren`’, ‘`Netzwerk Grundkonfiguration`’.
9. Sie befinden sich nun im Fenster ‘`Auswahl des Netzwerks`’.
10. Drücken Sie (F5), um das Device `ethernet` einzustellen (falls es noch nicht unter ‘`Device-Name`’ dort steht, z. B. `eth0`).
11. Drücken Sie (F3) und wählen Sie ‘`DHCP`’.
12. Drücken Sie (F4), um dieses Device zu aktivieren.
13. Mit (F10) wird die Konfiguration gespeichert.

14. Verlassen Sie YaST durch mehrmaliges Drücken der Esc-Taste.

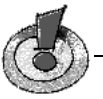
15. Sie können nun durch Eingabe von **`rcdhclient start`** Ihren Netzwerkzugang aktivieren. Danach können Sie z. B. mit **`ping www.suse.de`** den Zugang testen.

Diese Anleitung gilt für Versionen ab SuSE Linux 6.4.

Eine Alternative wäre es – falls IP-Adresse, Netzwerkmaske und Gateway bekannt und statisch sind – eine fixe Netzwerkkonfiguration einzustellen (siehe Abschnitt 2.3.2 auf Seite 20). Erkundigen Sie sich bei Ihrem Kabelbetreiber, ob Ihre IP-Adresse in Zukunft nicht geändert wird. Der Vorteil einer fixen Konfiguration: Sollte beim Booten der Kabelzugang gestört sein, so läuft der Bootvorgang problemlos weiter, und sobald die Netz-Störung behoben ist, können Sie sofort wieder in das Internet.

## 3.5 Modem-Anschluss

Der Anschluss eines Modems an den Rechner gestaltet sich ebenso wie unter anderen Betriebssystemen auch. Das Modem wird entweder über ein serielles Kabel mit dem Rechner verbunden oder (falls es sich um ein „internes“ Modem handelt) in einen freien Slot des Computers eingebaut. In YaST wird angegeben, an welcher Schnittstelle das Modem angeschlossen wird. Ein Link wird von der Device-Datei nach `/dev/modem` angelegt. Das Modem kann also über `/dev/modem` angesprochen werden, unabhängig davon, an welche Schnittstelle es angeschlossen wurde, wenn der Link richtig gesetzt wurde.



**Hinweis**

Es befinden sich auch sog. „WinModems“ im Handel. Diese können derzeit nicht unter Linux betrieben werden. Siehe dazu auch `file:/usr/share/doc/sdb/de/html/cep_winmodem.html` und `http://www.linmodems.org/`; zu Modems im Allgemeinen vgl. das `Modem-HOWTO.gz`.

Als *normale* Terminalprogramme können Sie z. B. minicom oder unter dem X Window System seyon einsetzen. Oder machen Sie die ersten Tests gleich mit wvdial (Abschnitt 3.6 auf der nächsten Seite).

### Minicom

Minicom ist ein einfach zu bedienendes Terminalprogramm, das in der Bedienung an das DOS-Programm Telix angelehnt ist. Alle Anwender, die minicom benutzen wollen, müssen vorher in die Datei `/etc/minicom.users` eingetragen werden. Hier wird festgelegt, wer mit welcher Konfiguration auf welches Modem zugreifen darf. Konfiguriert wird Minicom, indem Sie es als `'root'` folgendermaßen starten:

```
erde:/ # minicom -s
```

Die für den Betrieb erforderlichen Einstellungen sind selbsterklärend.

Hinweis

Die Tastenkombination (Strg) + (L) arbeitet nicht im xterm oder rxvt, dafür aber in kvt und in der Textkonsole.

## 3.6 Mit dem Modem in das Internet: PPP mit wvdial

Das Programm wvdial ist ein leistungsfähiges Tool, um analoge PPP-Verbindungen zu Internetprovidern, im folgenden ISP (engl. *Internet Service Provider*) genannt, herzustellen. Da die ISPs oft verschiedene Einstellungen im PPP-Protokoll verwenden, ist es manchmal sehr mühsam, die richtigen Optionen herauszufinden. Das erledigt wvdial durch intelligente Algorithmen.

In der Vergangenheit war es unter Linux erforderlich, bei der Internetkonfiguration auch den Nameserver (DNS – Domain Name System) des ISP anzugeben. Das ist mit wvdial nicht mehr nötig. Es erkennt automatisch den Nameserver des Providers, sofern dieser die entsprechende Information übermittelt.

### 3.6.1 Konfiguration von wvdial

Sie können wvdial bequem von YaST aus konfigurieren. Sie finden das Menü unter ‘`Administration des Systems`’, ‘`Netzwerk konfigurieren`’, ‘`PPP-Netzwerk konfigurieren`’. Das Menü sehen Sie in Abbildung 3.5 auf der nächsten Seite.

Gehen Sie folgendermaßen vor:

- Stellen Sie sicher, dass Sie bereits Ihr Modem eingerichtet haben (z. B. mit YaST). Dies ist entweder schon während der Erstinstallation geschehen oder kann jetzt nachgeholt werden.
- Wählen Sie nun den Menüpunkt ‘`Konfigurieren Sie Ihren Provider`’.
- Geben Sie dort Telefonnummer, Benutzerkennung und Passwort an (Abbildung 3.6 auf Seite 99).

  Wird die Verbindung über eine Nebenstellenanlage hergestellt, so tragen Sie bitte nach der Netzkennziffer („Amtsholung“, z. B. `0`) ein Komma oder ein `w` ein: `0,` oder `0w` .

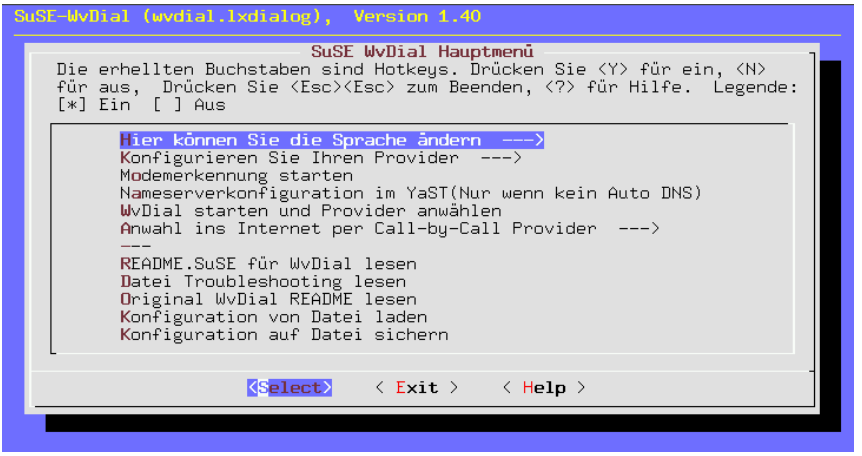



Abbildung 3.5: wvdial: Einstellung der Parameter

**Tipp**

Wenn Sie unter dem X Window System arbeiten und Schwierigkeiten haben etwas einzugeben, weil die Pfeil- (←, →) oder Rücklöschtasten (⟵) nicht wunschgemäß reagieren, sollten Sie YaST zunächst verlassen und mit Strg + Alt + F2-F6 auf eine Textkonsole wechseln, sich als `'root'` einloggen und dort wieder YaST starten.

- Wählen Sie die automatische Nameserver-Konfiguration aus. Falls diese nicht funktioniert, so müssen Sie den Nameserver in YaST wie gewohnt einstellen.
- Bestimmen Sie Ihr Wählverfahren, üblich ist die Tonwahl.
- Falls Ihr Rechner an einer Telefonanlage angeschlossen ist, wählen Sie den Punkt '`Modem an Telefonanlage`' aus. Es wird dann kein Wählton abgewartet.
- Unter '`Einwahlmodus`' sollte meist '`PPP-direkt-PAP/CHAP`' funktionieren.
- Verlassen Sie dieses Untermenü.
- Nach der Providerkonfiguration geht es zur Modemerkennung. Wählen Sie einfach den Menüpunkt '`Modemerkennung starten`' an.
- Hat das geklappt, wählen Sie den Punkt '`wvdial starten und Provider anwählen`'. Sie bekommen dann ein Fenster, in dem Sie diverse Meldungen sehen können.

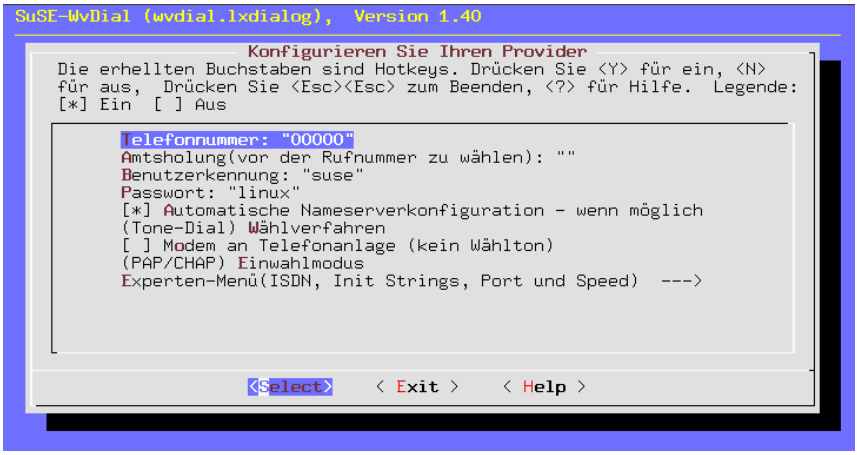



Abbildung 3.6: wvdial: Provider konfigurieren

- Wenn Sie die Meldung sehen, dass der PPP-Prozess gestartet wurde, können Sie das Internet schon benutzen.

- Prüfen Sie eventuell, ob eine Verbindung zustande gekommen ist. Dazu öffnen Sie ein weiteres Terminalfenster (in KDE: Alt + F2 und dann **xterm** eingeben), in das Sie eingeben:

  ```
  erde: # su
  ```

  Dann ist das 'root'-Passwort einzugeben und danach:

  ```
  erde: # tail -f /var/log/messages
  ```

  Jetzt können Sie die Systemmeldungen mitverfolgen. Sobald Sie Zeilen sehen mit "Local IP:" und "Remote IP:" – jeweils gefolgt von einer IP-Nummer – haben Sie die Gewissheit, dass die Verbindung in das Internet steht.

- Beenden Sie den Internetzugang mit Strg + C .

- Wenn das alles geklappt hat, können Sie mit einem einfachen Aufruf von wvdial von der Kommandozeile aus den Internetzugang starten und ihn auch mit Strg + C beenden.

- Sie können Ihre Konfiguration später auch ohne YaST mit dem Programm wvdial.lxdialog bzw. als grafisches Programm mit wvdial.tcl ändern. Dies ist jedoch nur als Benutzer 'root' möglich.

Dokumentation zu wvdial finden Sie im Verzeichnis /usr/share/doc/packages/wvdial .

#### wvdial als Benutzer und die Sicherheit

Wenn Sie wollen, dass außer ‘root’ auch normale Benutzer PPP-Verbindungen mit wvdial aufbauen können, so müssen Sie den jeweiligen Benutzer mit YaST in die Gruppen ‘uucp’ und ‘dialout’ eintragen. Diese Benutzer haben dann auch Zugriff auf die Datei /etc/wvdial.conf , die normalerweise auch das Login und das Passwort für den Internetzugang enthält. Um die Sicherheit zu erhöhen, können Sie das Passwort in eine geschützte Datei auslagern:

1. Wechseln Sie in das Verzeichnis /etc/ppp und legen Sie als Benutzer ‘root’ die Datei wvpw mit den Rechten 600 an:

   ```
   erde: # cd /etc/ppp
   erde:/etc/ppp # touch wvpw
   erde:/etc/ppp # chmod 600 wvpw
   ```

2. Laden Sie die Datei wvpw in einen Editor und tragen Sie dort nur das Passwort ein. Speichern Sie die Datei.

3. Kontrollieren Sie, ob die Rechte der Datei wvpw weiterhin stimmen. Ein

   ```
   erde:/etc/ppp # ls -l wvpw
   ```

   sollte Gewissheit schaffen (vgl. Ausgabe 3.6.1).

```
-rw-------   1 root   root       7  Jan 18 17:20 wvpw
```

Ausgabe 3.6.1: Ausgabe von **ls -l wvpw**

4. Wiederholen Sie die Konfiguration, wie in Abschnitt 3.6.1 auf Seite 97. Geben Sie jedoch als Passwort @/etc/ppp/wvpw an, mit dem vorangestellten „Klammeraffen“ teilen Sie wvdial mit, dass das Passwort dieser Datei zu entnehmen ist.

#### Modem piepst immer laut

Falls die Modem-Lautstärke während der Verbindung zu laut ist, können Sie /etc/wvdial.conf editieren und die Zeile

```
Init3 = ATM0
```

einfügen. Dieser Befehl schaltet den Modem-Lautsprecher ab.

### 3.6.2 Mehrere Provider mit wvdial

wvdial kann eine beliebige Anzahl von Parametersätzen verwalten. Dazu können Sie in der Datei /etc/wvdial.conf neben dem Abschnitt Dialer Default noch Zusatzabschnitte anlegen. Beim Starten von **wvdial** mit dem Namen eines solchen zusätzlichen Abschnitts werden dann zuerst die Parameter aus „Default“

gelesen. Alle Parameter, die in dem genannten Zusatzabschnitt nochmal angegeben sind, überschreiben die vorherigen Werte.

Hier ein kleines Beispiel für T-Online und den Call-by-Call-Provider Arcor (Datei 3.6.1). Darin wird die Konfiguration von YaST erstellt. Erweitert wird die Datei manuell um die Zeilen in Datei 3.6.2.

```
[Dialer Defaults]
Modem = /dev/ttyS0
Init1 = ATZ
Init2 = ATQ0 V1 E1 S0=0 &C1 &D2 S11=55 +FCLASS=0
Init3 = ATM0
Compuserve = 0
Tonline = 1
Dial Command = ATX3DT
Baud = 115200
Auto DNS = 1
Stupid Mode = 0
New PPPD = 1

Phone =0,0191011
Username = ????????
Password = ???????
```

Datei 3.6.1: `/etc/wvdial.conf`: Standard-Abschnitt

```
[Dialer arcor]
Phone =    010700192070
Username = arcor
Password = internet
```

Datei 3.6.2: `/etc/wvdial.conf`: Zusatz-Abschnitt

Wird wvdial ohne Parameter aufgerufen, wird eine Verbindung zu T-Online aufgebaut. Der Aufruf von **`wvdial arcor`** baut eine Verbindung zu Arcor auf. Lesen Sie dazu bitte auch die Manual-Page von **`wvdial`** (**`man wvdial`**).

### 3.6.3 ISDN-Terminaladapter

Diese Geräte ermöglichen den Anschluss an ISDN. Im Gegensatz zu gewöhnlichen ISDN-Karten werden Rechner und Terminaladapter über ein serielles Kabel verbunden.

### Hinweis

ISDN-Terminaladapter dürfen nicht mit TK-Anlagen mit eingebauter ISDN-Karte verwechselt werden. Diese sind zwar auch über ein serielles Kabel angeschlossen, verwenden aber ein proprietäres Protokoll über die serielle Schnittstelle und können deshalb nicht unter Linux betrieben werden! Im PC muss ein mitgelieferter CAPI-Treiber installiert werden, der von den Herstellern nicht für Linux zur Verfügung gestellt wird. Bekannt sind hier die Geräte:

- *Eumex 404 PC*
- *Eumex 322 PCi*
- *AVM Fritz!XPCDr.*
- *Neuhaus Triccy Data LCR*

Obwohl die Adapter im Prinzip ein analoges Modem simulieren, zeigen diese Adapter Besonderheiten; z. B.

- benötigen sie spezielle Befehle, um eine Point-to-Point-Verbindung zu ermöglichen und
- geben sie in der Voreinstellung erweiterte `CONNECT`-Meldungen aus.

Deshalb muss die Modemkonfiguration angepasst werden:

1. Verwenden Sie nicht die automatische Modemerkennung, die sonst mit YaST durchgeführt wird.
2. Gehen Sie in YaST in das Menü ‘`Administration des Systems`’ → ‘`Netzwerk konfigurieren`’ → ‘`PPP-Netzwerk konfigurieren`’ → ‘`Konfigurieren Sie Ihren Provider`’ → ‘`Experten-Menü`’ (ISDN, Init Strings, Port und Speed) → (Standard-Analog-Modem/non-ISDN) Modem Typ (analoges Modem/ISDN).
3. Passen Sie im ‘`Experten-Menü`’ die serielle Modemschnittstelle an; vgl. Abschnitt 3.6.1 auf Seite 97.
4. Loggen Sie sich als `‘root’` ein.
5. Erstellen Sie die Datei `/etc/wvdial.conf` per Hand; diese Datei wird sonst automatisch generiert. Die Datei sollte den Inhalt wie in Datei 3.6.3 auf der nächsten Seite haben.

   Bei <spezieller Eintrag1> und <spezieller Eintrag2> müssen Sie – je nach Gerät – die folgenden Werte eintragen:

   **Hersteller ELSA:** ELSA MicroLink ISDN/TLpro und ISDN/TLV.34:

   ```
   Init1 = AT&F\N10%P1
   Init2 = AT\V0
   ```

```
[Dialer Defaults]
Modem = /dev/modem
Baud = 115200
Init1 = <spezieller Eintrag1>
Init2 = <spezieller Eintrag2>
; Phone =
; Username =
; Password =
```

Datei 3.6.3: `/etc/wvdial.conf: Terminal-Adapter`

**Hersteller ELSA:** ELSA TanGo 1000 und ELSA TanGo 2000:

```
Init1 = AT&F$IBP=HDLCP
```

**`Init2`** entfällt hier.

**Hersteller Zyxel:** alle Modelle:

```
Init1 = AT&FB40
Init2 = ATXO
```

**Hersteller Hagenuk:** Speed/Viper Dragon:

```
Init1 = ATZ
Init2 = AT&FB8X0
```

**Andere Hersteller:** Sie können den vom Hersteller angegebenen „Initstring“ in der Dokumentation des Adapters nachschlagen. Manchmal sind auch Skripten für Unix oder Linux beigefügt, denen dieser String entnommen werden kann. Oder Sie schauen nach, mit welchem Initstring sich der Adapter unter einem anderen Betriebssystem einwählt, z. B. unter MS-Windows.

Alle anderen Konfigurationsschritte führen Sie wie im Handbuch Ihres Adapters beschrieben durch.

### 3.6.4 Konfiguration von PCI-Modems

Der IRQ und die IO-Adresse der seriellen Schnittstellen sind in Linux auf die Werte voreingestellt, die von ISA-Karten benutzt werden. Diese Werte sind ein Quasi-Standard und sorgen in vielen PCs dafür, dass kein Ressourcen-Konflikt auftritt. Die Ressourcen von PCI-Karten werden jedoch vom BIOS beim Booten vergeben und stimmen, wenn man das BIOS alleine entscheiden lässt, nicht mit den traditionellen Werten überein.

Gehen Sie deshalb vor der Konfiguration von wvdial folgendermaßen vor:

1. Bestimmen Sie die tatsächlichen Werte, die das BIOS den seriellen Schnittstellen zugeordnet hat, mit dem Befehls **`scanpci -v`**. Sie benötigen den Interrupt (IRQ) und die IO-Adresse (IO-port).

2. Integrieren Sie das Modem mit YaST über die Punkte 'Administration des Systems' / 'Hardware in System integrieren' / 'Modem konfigurieren'.

   Beachten Sie dabei eine eventuell vorhandene serielle Maus und andere serielle Schnittstellen; im Zweifelsfall wählen Sie `/dev/ttyS2`, um einen Konflikt mit einer zusätzlich eingebauten, herkömmlichen Schnittstellenkarte zu vermeiden.

3. Der Befehl **`setserial`** kann dazu verwendet werden, die Konfiguration der seriellen Schnittstelle zu verändern. Belegt das Modem z. B. den IRQ 5 und die IO-Adresse `0x220`, der Kernel erwartet jedoch den Interrupt 4 und Port `0x02f8`, so kann mit dem Befehl

   ```
   erde: # setserial /dev/ttyS2 irq 5 port 0x220
   ```

   Abhilfe geschaffen werden.

   Dieser Befehl muss allerdings fortan bei jedem Booten ausgeführt werden. Hier bietet sich ein Eintrag in die Datei `/etc/init.d/boot.local` an. Alternativ ist die Anpassung der Datei `/etc/init.d/serial`, im Abschnitt `start` möglich:

   ```
   run_setserial /dev/ttyS2 irq 5 port 0x220
   ```

   Nähere Informationen zu setserial findet man in der Manual-Page von **`setserial`** (**`man setserial`**).

4. Um zu testen, ob die Konfiguration der Schnittstelle erfolgreich war, können Sie **`wvdialconf /dev/null`** eingeben. Dabei werden alle `ttySx`-Schnittstellen geprüft und Ihr Modem sollte erkannt werden.

   *Hinweis:* Alternativ zur Konfiguration mit setserial können Sie die IRQ-Einstellungen im BIOS ändern. Dies ist nur möglich, wenn Ihr BIOS das zulässt und die IO-Adresse nicht geändert werden muss:

   Stellen Sie fest, in welchem PCI-Slot Ihre Schnittstellenkarte steckt. In manchen BIOS-Setup-Programmen gibt es ein Untermenü, in dem die Einstellungen der PCI-Schnittstellen festgelegt werden. Hier kann jedem Slot ein fester Interrupt (IRQ) zugewiesen werden. Tragen Sie hier den voreingestellten IRQ ein. In den meisten Fällen wird dies IRQ 3 oder IRQ 4 sein. Beim nächsten Start passt sich der tatsächliche IRQ der Voreinstellung an.

## 3.7 Sendmail-Konfiguration

Ist der Online-Anschluss erst einmal hergestellt, sei es über UUCP, PPP oder ISDN, soll dieser natürlich auch genutzt werden. Eine typische Anwendung hierfür ist E-Mail, elektronische Post. Dieser Abschnitt beschreibt die Konfiguration des Paketes sendmail. Eine Alternative zu sendmail ist postfix oder qmail – auf beide Pakete wird hier aber nicht weiter eingegangen.

Bei der Zustellung von E-Mail-Nachrichten entscheidet sendmail, wie die Nachrichten weiter transportiert werden sollen: über ein TCP/IP-Netzwerk mit dem Protokoll SMTP, in den lokalen E-Mail-Folder eines Benutzers oder über andere Transferprogramme wie UUCP.

### 3.7.1 Konfiguration mit YaST2

Eine einfache Sendmail-Konfiguration können Sie bequem mit YaST2 erledigen. Unter ‘Netzwerk/Erweitert’ finden Sie den Konfigurationsdialog für Sendmail. Hier können Sie aus folgenden Einstellungen wählen:

- ‘Rechner mit permanenter Netzverbindung (SMTP)’
  Hierbei handelt es sich um eine so genannte „Standleitung“, wie sie häufig in Firmen oder sonstigen Institutionen, die viel mit dem Internet arbeiten, zu finden ist. Die Verbindung zum Internet steht permanent – es ist also keine Einwahl nötig.

  Dieser Menüpunkt gilt jedoch auch für Nutzer eines lokalen Netzwerks, in dem es keine permanente Internet–Verbindung gibt, jedoch ein zentraler Mailserver zum E–Mail–Versand verwendet wird.

- ‘Einzelplatzrechner ohne Netzverbindung’
  Wenn Sie keinen Internetzugang haben und auch keinem Netz angehören, dann können Sie nur lokale E–Mails verschicken.

- ‘Rechner mit temporärer Netzverbindung (Modem oder ISDN)’
  Dies betrifft die meisten Benutzer, die zuhause einen Rechner haben, der keinem Netzwerk angehört, die aber gelegentlich in das Internet gehen – per Einwahl über das Modem, T–DSL/ADSL oder ISDN.

- ‘UUCP zur Mail-Übertragung benutzen’
  „UUCP“ bedeutet „Unix to Unix Copy Program“. Früher wurde es häufig für das Senden von E–Mails verwendet. Dieses Protokoll ist für Dialup–Verbindungen und wird heute nur noch sehr selten verwendet.

- ‘Expertenmodus für die Konfiguration von Sendmail’
  Dies ist der einzige Punkt, bei dem es ‘Weiter’ geht mit einer eigenen Konfigurationsmaske. Hier wird der Domain–Name für die lokale Zustellung erfragt (also der Name des Rechners, an dem man arbeitet), der Rechner für die ausgehende E–Mail (also der Mailserver, an den die E–Mails zunächst geschickt werden) und der Rechner für die gesamte E–Mail (oft identisch mit dem eben genannten Mail–Server).

  Im folgenden haben Sie die Möglichkeit, drei Punkte auszuwählen: ‘Mail nur in die Queue stellen’: Dies bewirkt, dass E–Mails nicht mehr sofort verschickt werden, sondern nur nach den Sendmail–Intervallen (z. B. alle 30 Minuten) oder nach manuellem Aufruf (`sendmail -q`). Bei Dialup–Verbindungen ist dies praktisch immer der Fall, da nicht bei jedem E–Mail–Versand sofort eine Verbindung aufgebaut wird. Lokale E–Mails sind nicht davon betroffen – sie werden sofort verschickt. ‘Rechnernamen nicht kanonifizieren’: Damit verhindern Sie einen automatischen Dialup–Verbindungsaufbau bei DNS–Anfragen. ‘Sendmail als SMTP-Daemon starten’: Dies brauchen Sie bei der Verwendung als Mailserver und im Zusammenhang mit fetchmail. Ferner sind noch ‘Parameter für den Sendmail-Aufruf’ einzugeben: Die wichtigsten sind `-bd`, `-q` und `-om`. ‘Domains, die über die "generics table" verändert werden können’: Das sind die Domains, die als „Localhost“ eingetragen

sind. Wenn Sie auch andere Domains modifizieren lassen wollen, dann müssen Sie sie hier eintragen. Mit ‘Weiter’ kommt die Konfiguration zum Abschluss.

- ‘Keine Modifikation von /etc/sendmail.cf’
  Diesen Punkt wählen Sie, wenn bereits eine Konfiguration besteht und diese nicht verändert werden soll.

YaST2 konfiguriert die Datei /etc/rc.config.d/sendmail.rc.config automatisch entsprechend Ihren Angaben. Nur beim Expertenmodus haben Sie (indirekten) Zugriff auf den Datei–Inhalt und können ihn manuell verändern. Aus dieser Datei wird mit Hilfe eines Skriptes die Datei /etc/sendmail.cf erzeugt, die von Sendmail eingelesen wird. Mit ‘Beenden’ schließen Sie die Konfiguration ab.

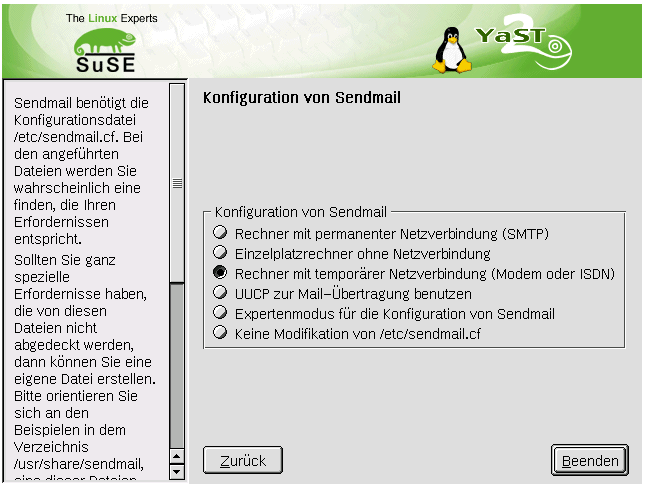



Abbildung 3.7: YaST2 Sendmailkonfiguration

### 3.7.2 Konfiguration mit YaST

Die Hauptkonfigurationsdatei von sendmail ist /etc/sendmail.cf. Für eine einfache Konfiguration kann man mit YaST ein paar Parameter setzen und damit eine gültige /etc/sendmail.cf erstellen lassen; die Eintragungen stehen dann in der /etc/rc.config.d/sendmail.rc.config und SUSEconfig schreibt anhand dieser Eintragungen unter Verwendung von /sbin/conf.d/SuSEconfig.sendmail die Datei /etc/sendmail.cf.

Da die Konfigurationsdateien des sendmail-Paketes sehr komplex sind, hat SuSE Linux zwei Konfigurationen vorbereitet, die die in der Regel vorkommenden Fälle weitgehend abdecken:

Wenn sendmail innerhalb eines TCP/IP-Netzwerkes verwendet werden soll, sollte man unbedingt einen gültigen DNS-Server besitzen. Dort sollte man für jeden Namen einen Extra-Eintrag („MX record“, „mail exchange record“) für E-Mail

machen. Die aktuellen Einstellungen kann man mit dem **host**-Befehl (aus dem Paket bind) überprüfen:

```
erde: # host sonne.kosmos.all
sonne.kosmos.all address 192.168.0.1
sonne.kosmos.all mail is handled (pri=10) by sonne.kosmos.all
sonne.kosmos.all mail is handled (pri=100) by mail-relay.kosmos.all
```

Falls dort kein Eintrag für Mail existiert, sollte man seinen DNS-Administrator um Hilfe bitten.

Folgende Parameter für eine E-Mail-Konfiguration können über YaST in der /etc/rc.config.d/sendmail.rc.config (vgl. die Datei /etc/mail/README) eingestellt werden:

- **SENDMAIL_TYPE="yes"**
  Diese Variable muss auf yes stehen, wenn die sendmail-Konfigurationsdatei aus den in der /etc/rc.config.d/sendmail.rc.config gesetzten Werten gebildet werden soll. Wenn man die /etc/sendmail.cf selbst herstellen möchte, dann ist no der richtige Wert.

- **SENDMAIL_LOCALHOST=**
  **"localhost sonne.kosmos.all www.kosmos.all"**
  sendmail muss wissen, welche E-Mail lokal abgespeichert und welche an einen anderen Zielrechner verschickt werden muss. Nur E-Mail an den lokalen Hostnamen wird per default als lokale E-Mail abgespeichert. Mit **SENDMAIL_LOCALHOST** kann man weitere Rechner-Namen – durch Leerzeichen getrennt – angeben, die auch als lokal angesehen werden sollen.

  Beispiel: Der Rechner heißt sonne.kosmos.all und ist zugleich WWW-Server für www.kosmos.all. Damit E-Mail an www.kosmos.all auch akzeptiert wird, muss man folgendes eintragen:
  **SENDMAIL_LOCALHOST="localhost www.kosmos.all"**

- **FROM_HEADER=kosmos.all**
  Als Absenderadresse wird normalerweise der lokale Rechnername verwendet. Die Adresse kann aber mit diesem Parameter beliebig verändert werden.

  Beispiel: Der Rechner heißt zwar erde.kosmos.all, E-Mail soll aber nur in der Form tux@kosmos.all verschickt werden (also ohne Rechnernamen). Das geht über den Eintrag: **FROM_HEADER=kosmos.all**.

- **SENDMAIL_SMARTHOST=mail-server.provider.de**
  Für alle nicht-lokale E-Mail fragt sendmail nach den DNS-Daten und will dann die E-Mail über das SMTP-Protokoll an den zuständigen Rechner schicken. Dieser Rechner kann irgendwo im Internet sein und hat u. U. nur eine langsame Verbindung zu unserem Rechner. Über diesen Parameter kann man daher einen Zwischenrechner angeben, der alle nicht-lokale E-Mail bekommt und diese dann weiter an den Zielrechner abliefert.

  Beispiel 1: Damit kann man auch bei einer Dialup-Verbindung alle E-Mail beim Provider abgeben, der dann für die Auslieferung ins Internet zuständig ist:
  **SENDMAIL_SMARTHOST=smtp:mail-server.provider.de**.

Beispiel 2: Ist man über UUCP angeschlossen, kann man alle nicht-lokale E-Mail an den UUCP-Server weitergeben:
**SENDMAIL_SMARTHOST=uucp-dom:uucp.kosmos.all**.

- **SENDMAIL_NOCANONIFY=no**
  sendmail schaut alle E-Mail-Adressen im Mail-Header nach und ersetzt die Namen mit den „Fully Qualified Domain Names“ (FQDN). Falls man beim E-Mail-Schreiben immer den vollständigen E-Mail-Namen angibt und vielleicht wegen einer Dialup-Verbindung nicht immer einen DNS-Server erreichbar hat, kann man das mit `yes` abschalten.

- **SENDMAIL_ARGS="-bd -q30m -om"**
  Mit diesen Parametern wird sendmail beim Booten des Rechners gestartet.
  Mit `-q30m` schaut sendmail alle 30 Minuten nach, ob im Queue-Verzeichnis `/var/spool/mqueue` noch E-Mail liegt, die ausgeliefert werden muss.
  `-bd` startet sendmail im „daemon mode“, damit wird E-Mail über das TCP/IP-Netzwerk von anderen Rechnern akzeptiert.
  Für Dialup-Verbindungen könnte man z. B. `-q30m` weglassen und E-Mail nur über einen direkten Aufruf von **`sendmail -q`** ausliefern; diesen Aufruf könnte man z. B. über einen crontab-Eintrag einmal pro Tag tätigen. Eine andere Möglichkeit wäre, **`sendmail -q`** noch in den Skripten zum Verbindungsaufbau unterzubringen. Dann wird bei jedem Verbindungsaufbau zusätzlich noch E-Mail übertragen.

- **SENDMAIL_EXPENSIVE=no**
  sendmail versucht, sofort eine E-Mail über SMTP an den nächsten Rechner weiterzugeben. Falls man nur zeitweise eine Verbindung zum Internet hat („Dial-On-Demand“), möchte man u. U. nicht für jede E-Mail eine Verbindung zum Provider starten.
  Mit `yes` wird alle E-Mail zunächst im Queue-Verzeichnis `/var/spool/mqueue` gehalten und nicht sofort weitergeschickt.

Alle lokale E-Mail wird über das Programm procmail in die lokalen E-Mail-Folder `/var/mail/<name>` abgespeichert. Bitte lesen Sie die Manual-Page von **`procmailrc`** (**`man procmailrc`**) und die Manual-Page von **`procmailex`** (**`man procmailex`**) sowie die Manual-Page von **`procmail`** (**`man procmail`**) für eine genaue Beschreibung dieses sehr flexiblen Programms.

Falls E-Mail nicht an den nächsten Rechner weitergegeben werden kann, wird sie in dem Queue-Verzeichnis `/var/spool/mqueue` gespeichert und beim nächsten „Queue-Run“ von sendmail erneut übertragen. Das Zeitintervall der „Queue-Runs“ wird beim Starten von sendmail angegeben oder das Übertragen der Nachrichten wird explizit durch den Aufruf von **`sendmail -q`** gestartet.

Weitere Einstellungen von sendmail kann man in den Dateien `/etc/aliases` und einigen Dateien im Verzeichnis `/etc/mail/` vornehmen. In den Dateien stehen auskommentierte Beispiele. Einige der Dateien müssen von den Textdateien mit dem Programm makemap in Datenbankdateien übersetzt werden. Das geschieht automatisch beim Aufruf von SuSEconfig oder beim Verlassen von YaST.

Für komplexere Konfigurationen sollte man die automatische Generierung von `/etc/sendmail.cf` durch **`SENDMAIL_TYPE=no`** abstellen und dann `/etc/`

`mail/linux.mc` als Vorlage für eine eigene Konfiguration nehmen. `linux.mc` enthält m4-Anweisungen und

```
erde: # m4 /etc/mail/linux.mc > /etc/sendmail.cf
```

erstellt über die Makros im Verzeichnis `/usr/share/sendmail` eine gültige sendmail-Konfiguration.

Weitere Dokumentation ist in den Verzeichnissen `/etc/mail`, `/usr/share/sendmail` und `/usr/share/doc/packages/sendmail` zu finden. Als Startadresse für WWW sollte man bei `http://www.sendmail.org/` anfangen. Für komplexere Aufgaben kommt man sicher nicht um das Sendmail-Buch aus dem O'Reilly-Verlag herum, das eine sehr gute und ausführliche Dokumentation zur sendmail-Konfiguration bietet.

## 3.8 Externe Mailboxen abrufen mit fetchmail

Ein sehr nützliches Programm zum Abrufen der eigenen Mail von externen Mailservern ist fetchmail. Die Konfiguration wird in einer Textdatei (`.fetchmailrc` im Homeverzeichnis) gespeichert. Das Programm selbst ist von der Kommandozeile aus bedienbar. Oftmals verwendet man **`fetchmail`** in Skripten, wenn man die Mail auto- oder halbautomatisch holen möchte.

Eine einfache Konfigurationsdatei könnte so aussehen:

```
poll pop.provider.net proto pop3 user otto pass secRET \
     options ssl
```

Da das Passwort in der Datei im Klartext gespeichert ist, *muss* diese Datei mit

```
erde: # chmod 600 .fetchmailrc
```

nur für den Benutzer lesbar gemacht werden. Der Aufruf von fetchmail kann dann periodisch über einen Cronjob erfolgen. Alternativ kann auch für mehrere Benutzer eine Fetchmail-Konfiguration erstellt werden und von `'root'` z. B. im **`ip-up`**-Skript Mail gesammelt abgeholt werden. Die `.fetchmailrc` von `'root'` könnte dann so aussehen:

```
poll pop.provider.net proto pop3 user otto pass secRET \
     options ssl is user1
poll pop3.mails.net proto pop3 user helga pass geHeim \
     is user2
```

Das `is user` am Ende sagt fetchmail, welchem lokalen Benutzer die Mail zugestellt werden soll. `/etc/ppp/ip-up` kann um den Fetchmail-Aufruf ergänzt werden; vgl. Datei 3.8.1.

```
ip-up)
      /usr/bin/fetchmail >/var/log/fetchmail 2>&1
```

Datei 3.8.1: `/etc/ppp/ip-up`: Eintrag für fetchmail

Auf diese Weise wird bei PPP-Verbindungen immer zu Beginn für `'user1'` und `'user2'` Mail abgerufen. Als weitere Alternative findet sich im Dokumentationsverzeichnis `/usr/share/doc/packages/fetchmail` ein Startup-Skript, um fetchmail als Daemon zu starten.

## 3.9 News: Die neuesten Meldungen des USENET

Einer der wichtigsten Dienste, die das Internet zur Verfügung stellt, ist die Übermittlung und Verteilung von Nachrichten, die in verschiedenen Gruppen (engl. *Newsgroups*) organisiert sind; dieser Teil des Internet wird als das USENET bezeichnet. Erst durch die Existenz dieses Mediums war die Entwicklung von Linux überhaupt möglich, und nur durch diese hocheffiziente Art der Kommunikation ist die rapide Weiterentwicklung sowie das schnelle Entfernen von Fehlern aus dem System möglich. Weiterhin ist das USENET ein wichtiges Medium, wenn es um die gegenseitige Unterstützung der Linux-Anwender untereinander geht.

Da eine komplette Beschreibung eines Newssystems mit all seinen vielfältigen Möglichkeiten (wie das Weiterreichen an andere Rechner) den Rahmen dieses Buches bei weitem sprengen würde, soll hier nur die Konfiguration eines lokalen Newssystems beschrieben werden.

Tipp

Größere Systeme sollten auf Paket `inn`, Serie n zurückgreifen; Hinweise zur INN-Installation liegen unter `/usr/share/doc/packages/inn`; der INN ist auch für UUCP-Systeme zu bevorzugen.

### 3.9.1 Das News-System Leafnode

Das Paket `leafnode` ist ein bestens geeignetes News-System für kleinere Netze oder Einzelplatz-Rechner mit einer einfachen, nicht unbedingt schnellen Verbindung ins Internet. Das Paket besteht aus mehreren Teilen: dem eigentlichen NNTP-Server leafnode, dem Programm fetchnews (früher: `fetch`) zum Holen der Nachrichten und dem Programm texpire zum Löschen alter bzw. nicht mehr relevanter Nachrichten; als Add-Ons gibt es Tools zum Verwalten des Datenbestands unter `/var/spool/news`. Dokumentation zu all diesen Komponenten finden Sie unter `/usr/share/doc/packages/leafnode` sowie in Manual-Page von **`leafnode`** (**`man 8 leafnode`**) und den dort genannten Manual-Pages.

#### Voraussetzung für den Einsatz von Leafnode

- Einen externen NNTP-Server müssen Sie über Modem (PPP), eine ISDN-Verbindung oder eine andere Netzverbindung (z. B. Ethernet) direkt erreichen können; von einem solchen NNTP-Server können Sie dann die „News“ beziehen. Fragen Sie gegebenenfalls Ihren „Internet Service Provider“ (ISP) nach den Daten des zu benutzenden NNTP-Servers.
- Das Paket `leafnode`, Serie n muss installiert sein.
- Plattenplatz unter `/var/spool/news`

- Die im Folgenden genannten Konfigurationsschritte müssen durchgeführt werden.

#### Lokaler NNTP-Server

Zunächst ist sicherzustellen, dass Leafnode als *lokaler* NNTP-Server läuft.

1. In der Datei `/etc/rc.config` die Variable <NNTPSERVER> auf den Wert `localhost` setzen. Freilich können Sie auch den „richtigen“ Namen Ihres Linux-Rechners anstelle von `localhost` verwenden (z. B. `erde`), wenn Sie Ihren Rechner entsprechend konfiguriert haben; in vernetzten Umgebungen ist dies zwingend erforderlich! Das Setzen der Variablen <NNTPSERVER> erledigen Sie am sichersten mit YaST; denn YaST ruft nach dem Verlassen der Maske mit (F10) das Skript **`SuSEconfig`** automatisch auf.
2. In der Datei `/etc/leafnode/config` mit einem Editor notwendige bzw. gewünschte Anpassungen vornehmen. Dort muss unbedingt der Name des NNTP-Servers Ihres Providers eingetragen werden (bei **`server =`**).
3. Treffen Sie Vorkehrungen, damit leafnode vom **`inetd`** gestartet wird. Schalten Sie zu diesem Zweck den `nntp`-Eintrag in `/etc/inetd.conf` durch Entfernen des Kommentarzeichens (`'#'`, alles in einer Zeile bitte!) frei; vgl. Datei 3.9.1.
4. Starten Sie inetd von Hand erneut, damit diese Konfiguration zum Tragen kommt; dazu kann der Befehl **`rcinetd restart`** verwendet werden.

```
nntp   stream  tcp   nowait  news     /usr/sbin/tcpd
                                      /usr/sbin/leafnode
```

Datei 3.9.1: **`inetd`**-Eintrag für leafnode

Nun ist lokal alles vorbereitet, damit zum ersten Mal Kontakt zum Newsserver des Providers aufgenommen werden kann.

**Tipp**

Mit **`telnet localhost 119`** können Sie überprüfen, ob leafnode sich meldet; falls ja, geben Sie **`quit`** ein, um wieder zur Kommandozeile zurückzukommen.

#### Das Newssystem initialisieren und betreiben

Jetzt kann das System initialisiert werden. Falls noch nicht geschehen, stellen Sie eine IP-Verbindung zu Ihrem ISP her (in der Regel per Modem oder

ISDN). Bei der ersten Kontaktaufnahme mittels fetchnews werden vom entfernten Newsserver die Informationen zu den verfügbaren Newsgroups geholt und unter `/var/spool/news/interesting.groups` abgelegt; wenn Sie im Detail verfolgen wollen, was **`fetchnews`** tut, verwenden Sie die Option **`-vvv`**:

```
erde:~ # fetchnews -vvv
```

Es sind noch keine Artikel verfügbar, dennoch muss man nun einen NNTP-fähigen Newsreader aufrufen und in die (noch leeren) Gruppen einmal hineinschauen (vgl. Abschnitt 3.9.1). leafnode registriert dies; beim nächsten **`fetchnews`**-Aufruf werden genau diese angewählten Gruppen mit Nachrichten gefüllt werden.

Wenn nicht jedes Mal beim „Online-Gehen“ der **`fetchnews`**-Aufruf von Hand eingegeben werden soll, dann nehmen Sie ihn z. B. in Ihr `/etc/ppp/ip-up`-Skript auf.

#### Das Newssystem verwalten

leafnode wurde nach dem Prinzip einer weitgehenden Selbstverwaltung entworfen. Wenn daher bestimmte Newsgruppen von keinem Benutzer mehr gelesen werden, dann werden diese nach einer vorgegebenen Frist nicht mehr von **`fetchnews`** geholt.

Man hat im Grunde nur dafür zu sorgen, dass alte Artikel entfernt werden; diese Aufgabe erledigt texpire; in `/etc/crontab` ist ein passender Eintrag bereits vorgesehen; entfernen Sie das Kommentarzeichen `'#'`, wie in Datei 3.9.2 gezeigt (alles in einer Zeile bitte!).

```
0 22 * * * root  test -x /usr/sbin/texpire && /usr/sbin/texpire
```

Datei 3.9.2: Expire-Eintrag für leafnode in `/etc/crontab`

Erklärungen zu Einstellmöglichkeiten, die über die Datei `/etc/leafnode/config` vorgenommen werden können, finden Sie – wie bereits gesagt – in Manual-Page von **`leafnode`** (**`man leafnode`**).

### Lesen der News

Für das Lesen der News stehen verschiedene Programme zur Verfügung, z. B. nn, tin oder pine; auch Netscape oder Emacs können zum Newslesen verwendet werden. Die Wahl des Newsreaders ist oftmals eine Frage des persönlichen Geschmackes. Die Newsreader können sowohl für den Zugriff auf einen Newsserver – wie in einem Netzwerk üblich – als auch für den Zugriff auf das lokale Spoolverzeichnis konfiguriert werden. Entsprechend vorkonfigurierte Pakete finden sich in der Serie n von SuSE Linux. Wenn Sie mit tin auf den leafnode-NNTP-Server zugreifen wollen (vgl. Abschnitt 3.9.1 auf Seite 110 ff.), dann rufen Sie diesen Newsreader mit dem Kommando **`rtin`** auf.

# 4 FTP

## 4.1 Allgemeines

In diesem Kapitel wird das „File Transfer Protocol“ (FTP) besprochen. FTP ist ein vergleichsweise alter Internet-Dienst, der es erlaubt, Dateien zwischen zwei Rechnern zu übertragen. Zwischen den Rechnern muss dazu eine TCP/IP-Verbindung bestehen (wie es z. B. im Internet der Fall ist).

### 4.1.1 Intention dieses Kapitels

Nach der vollständigen Lektüre dieses Kapitels sollten Sie

- wissen, wozu FTP dient und in welchem Zusammenhang es mit TCP/IP steht
- wissen, was ein FTP-Client und ein FTP-Server ist
- eine Auswahl von FTP-Clients und FTP-Servern kennen
- einen FTP-Client bedienen können
- einen FTP-Server grundlegend konfigurieren und starten können
- grundlegende Sicherheitserwägungen im Bezug auf FTP anstellen können

Sollten Sie einen FTP-Server im Internet oder einem großen Firmennetz administrieren, kann die Lektüre des vorliegenden Kapitels natürlich keinen professionellen FTP-Administrator aus Ihnen machen. Für diesen Fall empfiehlt sich eine Schulung und das Studium der im letzten Abschnitt genannten Bücher.

### 4.1.2 Einführung

FTP funktioniert als klassischer Client/Server-Dienst: Ein FTP-Client vom Rechner A greift, nach erfolgter Anmeldung mittels Login und Passwort, auf den FTP-Server des Rechners B zu und kann nun Dateien von B nach A übertragen (FTP-get, Download). Eine Übertragung von A nach B ist ebenfalls möglich (FTP-put, Upload). Für beide Übertragungsmöglichkeiten und alle anderen Dateioperationen müssen die Zugriffsrechte auf dem Server B entsprechend konfiguriert sein. Der FTP-Client darf nichts, das der FTP-Server nicht explizit erlaubt.

Zum Zeitpunkt der Entwicklung von FTP war Sicherheit im Internet noch kein großes Thema (so wird z. B. das Passwort der Anmeldung des Clients beim Server unverschlüsselt übertragen). Auch aus diesem Grund wird FTP im Internet

häufig nur noch für öffentliche Dateiarchive verwendet. Die FTP-Server dieser Archive erlauben sog. „Anonymous-FTP“, d. h. Login und Passwort sind allgemein bekannt, die auf dem FTP-Server zur Verfügung gestellten Daten nicht vertraulich. Für das Aktualisieren von (privaten) Homepages bei Massenprovidern, die persönliche Domains anbieten (`eigener-name.de`), ist FTP ebenfalls verbreitet.

## 4.2 FTP-Clients

Ein FTP-Client ist ein Anwenderprogramm und dient dazu, mit dem FTP-Server Verbindung aufzunehmen. SuSE Linux enthält eine Reihe von FTP-Clients, die zwar alle im Prinzip das gleiche können, aber von Aussehen und Bedienung recht unterschiedlich daherkommen.

Welchen FTP-Client Sie benutzen, ist letztendlich Geschmackssache. Eine Ausnahme ist sicherlich GNU wget, da dieser sich einfach innerhalb eines Skripts aufrufen lässt und außer FTP noch HTTP (das Protokoll des WWW) beherrscht. Diesen Client sollten Sie neben dem FTP Command-Line Client auf jeden Fall installieren.

Ein FTP-Client sollte FTP-reget beherrschen, damit eine abgebrochene Übertragung wieder aufgenommen werden kann, ohne von vorn anfangen zu müssen. Netscape und einige andere (meist ältere) Clients können kein FTP-reget. Wenn Sie Ihr SuSE Linux hinter einer Firewall betreiben, muss der Client in der Regel passives FTP können, falls FTP überhaupt erlaubt ist – viele Firmenfirewalls erlauben nur Mail und WWW. Eine Firewall ist eine Art Filter für Datenpakete, der interne Netze vor dem unberechtigten Zugriff von außen schützen soll. Lesen Sie hierzu auch das Kapitel 8.1 auf Seite 155.

Die Pakete mit den FTP-Clients befinden sich in Serie `n`, Serie `gnm` und Serie `xap`.

### 4.2.1 FTP Command-Line Client (`lukemftp`)

Als FTP Command-Line Client wird bei SuSE Linux standardmäßig lukemftp installiert. Bei diesem Client erfolgt die Bedienung durch Eingabe von Kommandos an einem Kommandoprompt. Trotz dieser scheinbar komplizierten Handhabung ist der Client einfach bedienbar – ein halbes Dutzend Befehle sind für eine typische Sitzung ausreichend. Der `lukemftp` ist standardmäßig auf binäre Übertragung („Binary Transfer“) und passives FTP (s. u.) eingestellt.

Beispiel 4.2.1 auf der nächsten Seite zeigt das Protokoll einer typischen Sitzung (der Übersichtlichkeit halber gekürzt und vereinfacht) mit Verbindung zu `ftp.suse.com`.

```
tux@erde: > ftp ftp.suse.com
Connected to ftp2.suse.com.
220- Welcome to the SuSE ftp server ftp.suse.com.
Please make sure to read pub/INDEX before reporting any
problems to ftpadmin@suse.com.
Login as ‘ftp´ or ‘anonymous´ and use e-mail address as password!
220 SuSE FTP Server ready
Name (ftp.suse.com:tux): ftp
331 Anonymous login ok, send complete e-mail address as password.
Password:
230-
+---------------------------------------------------------------+
|   Welcome to the SuSE Linux FTP archives in California, USA    |
+---------------------------------------------------------------+
You are visitor 401 out of 650 connections in your class.
230 Anonymous access granted, restrictions apply.
Remote system type is UNIX.
ftp> cd pub/suse/i386/update/7.1/a1
250 CWD command successful.
ftp> ls
227 Entering Passive Mode (202,58,118,12,13,84).
150 Opening ASCII mode data connection for file list
-rw-r--r--   1 suse  suse  263310 Mar 6 01:43 package.i386.rpm
-rw-r--r--   1 suse  suse     590 Mar 6 06:03 package.i386_de.info
-rw-r--r--   1 suse  suse     535 Mar 6 06:03 package.i386_en.info
226 Transfer complete.
ftp> binary
200 Type set to I.
ftp> get package.i386.rpm
227 Entering Passive Mode (202,58,118,12,13,168).
150 Opening BINARY mode data connection for package.i386.rpm
(263310 bytes).
226 Transfer complete.
263310 bytes received in 00:17 (14.44 KB/s)
ftp> ?
Commands may be abbreviated.  Commands are:
[...] Gekürzt
ftp> quit
221 Goodbye.
```

Datei 4.2.1: Beispiel einer typischen FTP-Sitzung

Nach erfolgter Anmeldung werden die Kommandos am Kommandoprompt `ftp>` eingegeben. Die wichtigsten Kommandos sind

- **`cd`** – **`cd Verzeichnis`** wechselt in ein neues Verzeichnis auf dem FTP-Server.
- **`dir`** – Zeigt Verzeichnisinhalte an (**`ls`** funktioniert auch).
- **`binary`** – Stellt den Übertragungsmodus auf binäre Übertragung ein. Dies ist für alle Dateien, die nicht ausschließlich Text enthalten (z. B. RPM-Pakete), erforderlich.
- **`get`** – **`get Dateiname`** überträgt eine Datei vom Server zum Client (**`mget`** kann mehrere Dateien auf einmal übertragen).

- **put** – **put Dateiname** überträgt eine Datei vom Client zum Server (**mput** kann mehrere Dateien auf einmal übertragen).
- **?** – Zeigt eine Liste der verfügbaren Kommandos an.
- **quit** – Schließt die Verbindung zum FTP-Server.
- **passive** – Schaltet zwischen aktivem (normalem) und passivem FTP hin und her.

### 4.2.2 Nicht-grafische Clients

Unter dem Begriff „nicht-graphische Clients“ werden Clients verstanden, die kein X-Window System zum Betrieb benötigen. Darunter fallen u. a.

- Der bereits im vorangegangenen Abschnitt besprochene **lukemftp**
- GNU **wget**
- NcFTP (**ncftp**)
- Der Midnight Commander (**mc**)

NcFTP funktioniert ähnlich wie `lukemftp` und wird hier nicht im Detail besprochen. Der Midnight Commander ist eigentlich ein Filebrowser, hat aber auch eine FTP-Funktion. Er versucht, die subjektiv komfortable Oberfläche eines graphischen Clients nachzubilden (vgl. Abb. 4.1).

Der zweifellos interessanteste FTP-Client dieses Abschnitts ist GNU **wget**. Das Programm **wget** beherrscht neben FTP auch HTTP und lässt sich sehr gut in Skripten verwenden, ist also richtig Unix-like. Eine Zusammenfassung der Kommandozeilenoptionen, die **wget** beherrscht, erhalten Sie mit dem Kommando **wget --help**.

Beispiel 4.2.2 auf der nächsten Seite zeigt ein Skript, das eine einfache Alarmierung ausführt (Mail an ein Mobiltelefon), falls der FTP-Server `ftp.meinefirma.de` ausfällt.

Man kann das Skript per **cron** z. B. alle Viertelstunde ausführen lassen und erhöht damit beträchtlich die Chance, einen Ausfall beizeiten zu entdecken.

### 4.2.3 Graphische Clients

Unter den Begriff „graphische Clients“ werden Clients verstanden, die das X-Window System zum Betrieb benötigen. Darunter fallen u. a.

- gFTP (neu, wird aktiv weiterentwickelt)
- IglooFTP (neu, kommerzielle Variante erhältlich)
- LLNL XFTP (alt, wird nicht mehr weiterentwickelt)
- xmftp (alt, wird nicht mehr weiterentwickelt)

```
#!/bin/bash
#
PATH=/bin:/sbin:/usr/sbin:/usr/bin
TARGETHOST="ftp.meinefirma.de"
TARGETURL="ftp://ftp.meinefirma.de/pub/INDEX"
TESTFILE=/tmp/INDEX
ALARMSENT=/tmp/INDEX.checked
ALARMADR="01231234567@mobiltelefonfirma.de"

function ftpbroken () {
 wget --timeout=300 --quiet --directory-prefix=/tmp --passive-ftp \
 $TARGETURL > /dev/null 2>&1
}

test -f $TESTFILE && rm -f $TESTFILE
# do not mailbomb the cellphone ;-)
test -f $ALARMSENT && exit 0

# main program

ftpbroken || {
 touch $ALARMSENT
 echo "$TARGETHOST ftp failed on ‘date‘." | mail $ALARMADR;
}
```

Datei 4.2.2: Skript für Ausfall-Alarm

- Webbrowser (z. B. Netscape, Konqueror etc.)

Mit Ausnahme der Webbrowser funktionieren alle diese Clients ähnlich: Es gibt ein zweigeteiltes Fenster, in dem nebeneinander das lokale Verzeichnis und das aktuelle Verzeichnis auf dem FTP-Server angezeigt wird. Alle Einstellungen lassen sich via Popup-Menü oder Texteingabe-Fenster vornehmen. Meist kann man per Mausoperation Dateien einfach hin- und herschieben. FTP-Kommandos müssen nicht direkt eingegeben werden. gFTP und IglooFTP sind relativ neu und haben umfangreiche Konfigurationsmöglichkeiten. Sofern die genannten Clients keine direkte Hilfefunktion haben, finden Sie Hinweise zur Benutzung in den entsprechenden Verzeichnissen in `/usr/share/doc/packages/PAKETNAME/`.

## 4.3 FTP-Protokoll

Das FTP-Protokoll ist in RFC 959 und einigen späteren Ergänzungen (u. a. RFCs 1579, 1635 und 2228) definiert. Ein RFC („Request For Comment“) ist eine öffentlich zugängliche Spezifikation, die Programmierer brauchen, um eine bestimmte Funktion (im vorliegenden Fall FTP) richtig zu implementieren (quasi in ein funktionierendes Programm zu „gießen“). Die Lektüre eines RFC ist ziemlich trocken und für einen Anwender nicht unbedingt erforderlich. Für den Administrator eines FTP-Servers mag die Lektüre bisweilen aus Verständnisgründen nützlich sein.

Die Abbildung 4.3 zeigt den Aufbau einer Client/Server-Verbindung sowohl für

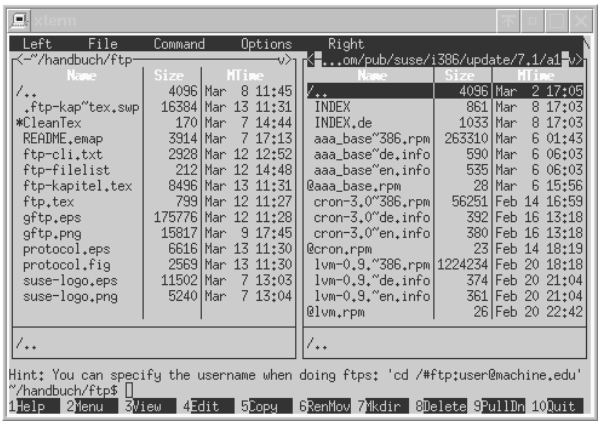



Abbildung 4.1: Der Filebrowser Midnight Commander als FTP-Client

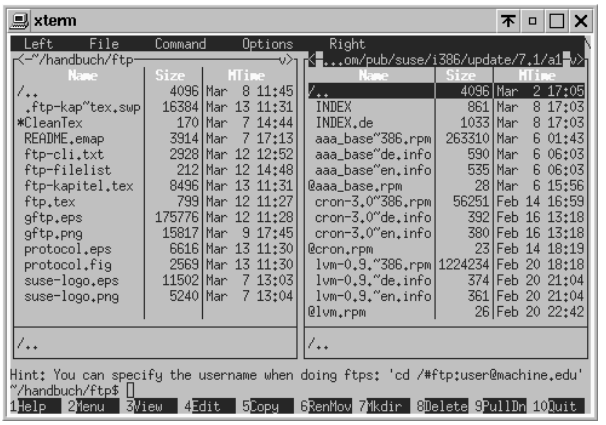



Abbildung 4.2: Der FTP-Client gFTP

aktives (normales) als auch passives FTP. Passives FTP ist aufgrund der Verbreitung von Firewalls heute der häufigere Fall. Die eigentliche Übertragung der (Nutz-) Daten beginnt in beiden Fällen nach Schritt 4.

Die TCP-Ports 20 (FTP Daten) und 21 (FTP Kommando) auf dem FTP-Server sind die standardmäßig hierfür vorgesehenen Ports. Alle anderen Portnummern in der Abbildung 4.3 sind willkürlich gewählt. Sie werden von TCP/IP bei jedem Verbindungsaufbau neu ausgehandelt. Der wesentliche Unterschied zwischen aktivem und passivem FTP ist, dass bei aktivem FTP der FTP-Server eine Datenverbindung zum FTP-Client aufzubauen versucht. Im Falle von passivem FTP versucht stattdessen der Client eine Datenverbindung zum Server aufzubauen (jeweils Schritt 3 in beiden Teilen der Abbildung). Dieser Vorgang steht nicht im Zusammenhang mit der späteren Übertragungsrichtung der Daten (Up- bzw. Download).

Neben aktivem (normalem) und passivem FTP beim Aufbau der Verbindung

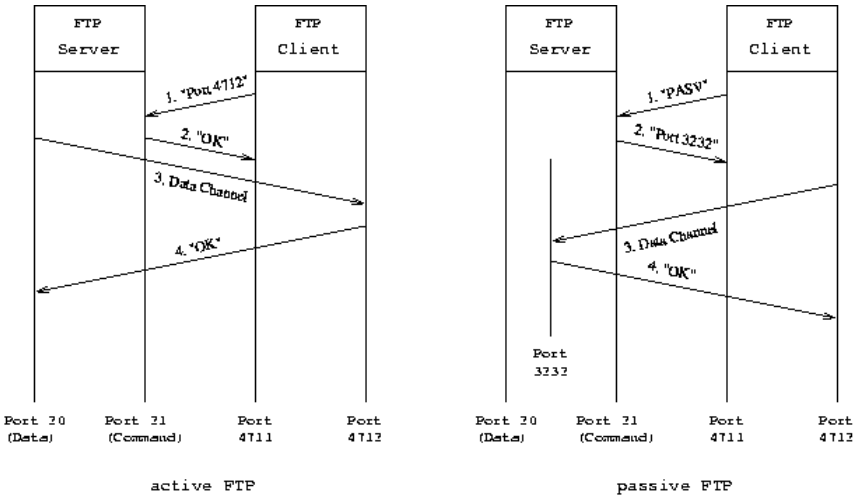



Abbildung 4.3: Verbindungsaufbau bei FTP

kennt FTP noch zwei verschiedene Übertragungsmodi: ASCII- und Binary-Transfer. Der ASCII-Transfer ist dafür gedacht, eine Umwandlung der unterschiedlichen Text-Formate verschiedener Betriebssysteme quasi während der Übertragung zu ermöglichen. So gibt es z. B. unterschiedliche Zeilenende-Zeichen bei Unix (nur Line Feed), MS-DOS (Line Feed und Carriage Return) und Apple Macintosh (nur Carriage Return). Weiterhin existieren verschiedene Zeichensatztabellen (ASCII, EBCDIC etc.). Der ASCII-Transfer ist heute praktisch bedeutungslos. Ein guter Text-Editor (gibt es für alle Betriebssysteme) kann Texte aller Typen richtig darstellen und bearbeiten. Benutzen Sie immer Binary-Transfer. Binary-Transfer überträgt Dateien ohne irgendwelche Änderungen vorzunehmen. Wenn Sie versehentlich z. B. ein RPM-Paket im ASCII-Mode übertragen haben, ist das Paket dadurch kaputt und nicht installierbar.

## 4.4 FTP-Server

Ein FTP-Server läuft als Dienst (unter Unix häufig „Daemon“ genannt) auf einem Rechner (der oft ebenfalls als FTP-Server bezeichnet wird) und ermöglicht FTP-Clients, mit ihm Verbindung aufzunehmen. SuSE Linux enthält eine Reihe von FTP-Servern, die sich im Wesentlichen durch ihre Konfigurierbarkeit unterscheiden.

Die verschiedenen FTP-Server werden entweder direkt als Masterprozess, der für jeden zu bedienenden Client Kindprozesse erzeugt, oder über den **`inetd`**-Daemon gestartet. In letzterem Fall übernimmt der **`inetd`**-Daemon das Starten der Prozesse für die einzelnen Clients.

Eine besondere Sicherungsmöglichkeit bei FTP-Servern ist der sog. „Change Root“-Mechanismus, der bei Anonymous-FTP zum Einsatz kommt: Nach er-

folgter Anmeldung beim FTP-Server führt dieser ein **chroot**-Kommando aus. Mit diesem Kommando kann man das Root- („Wurzel“)-Verzeichnis des Linux-Filesystems im laufenden Betrieb ändern, d. h. der Benutzer des FTP-Clients ist in einer Change Root-Umgebung quasi „gefangen“ – auf Verzeichnisse, die im Verzeichnisbaum oberhalb der Change Root-Umgebung liegen, besteht kein Zugriff mehr. üblicherweise wird als Change Root das Homeverzeichnis des Benutzers `'ftp'` gesetzt.

Die Pakete mit den FTP-Servern befinden sich in der Serie `n`. Neben den hier beschriebenen gibt es eine Reihe weiterer, frei verfügbarer FTP-Server. Weitere Informationen hierzu finden Sie in den im letzten Abschnitt dieses Kapitels genannten Verweisen.

### 4.4.1 BSD FTP Daemon (`in.ftpd`)

Dieser FTP-Server ist in der Standard-Installation von SuSE Linux enthalten, d. h. er ist nach Abschluss einer Standard-Installation sofort verfügbar. Sollten Sie keine Standard-Installation durchgeführt haben, installieren Sie einfach das Paket `ftpd`, Serie `n`.

Der BSD FTP Daemon wird vom **inetd** gestartet und hat aus Sicherheitsgründen noch **tcpd** (s. u.) vorgeschaltet. Den zugehörigen Eintrag in `/etc/inetd.conf` sehen Sie in Beispiel 4.4.1.

```
ftp     stream  tcp     nowait  root    /usr/sbin/tcpd  in.ftpd
```

Datei 4.4.1: Eintrag für FTP in der Datei `/etc/inetd.conf`

Der BSD FTP Daemon ist vorwiegend für den Einsatz in Intranets und hier für kleine, einfache Installationen geeignet, da er nur wenige Konfigurationsmöglichkeiten bietet. Für Server im Internet und hohem Datendurchsatz werden eher der WU-FTPD oder der ProFTPD eingesetzt.

### 4.4.2 WU-FTPD (`wu.ftpd`)

Der ursprünglich an der Washington University entwickelte WU-FTPD eignet sich, insbesondere in der älteren, als weitgehend sicher geltenden Version 2.4.2, gut für einen FTP-Server im Internet. Außer der Version 2.4.2 enthält das Paket `wuftpd`, Serie `n`, noch die neuere Version 2.6.1, die umfangreichere Konfigurationsoptionen bietet (z. B. bessere Unterstützung sog. „virtueller“ FTP-Server).

Das folgende Beispiel zeigt die Konfiguration für einen WU-FTPD Version 2.4.2, der via **inetd** gestartet wird und ausschließlich Anonymous-FTP erlauben soll. Im Verzeichnis `pub/incoming` sollen Benutzer Dateien ablegen können, die weder per FTP übertragen, noch anzeigbar sein sollen. Hierzu müssen neben passenden Einträgen in `/etc/ftpaccess` auch die Zugriffsrechte auf das Verzeichnis `pub/incoming` richtig gesetzt sein (mittels des **chmod**-Kommandos, z. B. auf oktal `1733`).

Den passenden Eintrag in `/etc/inetd.conf` sehen Sie in Beispiel 4.4.2.

```
ftp   stream  tcp     nowait  root    /usr/sbin/tcpd  wu.ftpd -a
```

Datei 4.4.2: Eintrag für FTP in der Datei `/etc/inetd.conf`

Die passende Konfigurationsdatei `/etc/ftpaccess` sehen Sie in Beispiel 4.4.3.

```
# /etc/ftpaccess - example for WU-FTPD

class  anon     anonymous  *
limit  anon     100  Any /etc/ftp/dead.msg

# email of the responsible person for the %E-cookie
email ftpadmin@meinefirma.de

# show messages...
banner  /etc/ftp/banner.msg
message /etc/ftp/welcome.msg login
message INDEX cwd=*

# passwd-check  <none|trivial|rfc822>  [<enforce|warn>]
passwd-check    none

log transfers anonymous inbound,outbound
log security anonymous

shutdown /etc/shutmsg

# global access rules

delete          no     anonymous         # delete permission?
overwrite       no     anonymous         # overwrite permission?
rename          no     anonymous         # rename permission?
chmod           no     anonymous         # chmod permission?
umask           no     anonymous         # umask permission?

noretrieve core .notar /etc/group /etc/passwd /pub/incoming

# upload rules for ftp.meinefirma.de

upload  /ftp  *                 no
upload  /ftp  /pub/incoming     yes    root root   0440  nodirs
upload  /ftp  /pub/incoming/*   yes    root root   0440  nodirs
```

Datei 4.4.3: Konfigurationsdatei `/etc/ftpaccess`

Zeilen, die mit dem Zeichen # beginnen, sind Kommentare. Nach dem Ändern des Eintrages in `/etc/inetd.conf` ist ein Signal an den **`inetd`** erforderlich, damit dieser seine Konfiguration neu einliest. Am einfachsten erfolgt dies mit dem Kommando **`rcinetd reload`**. Dieses Kommando schickt ein sog. SIGHUP-Signal an den **`inetd`**, was auch von Hand mit dem Kommando **`kill -HUP inetd-pid`** erfolgen kann. **`inetd-pid`** ist die Prozessnummer (PID, Process ID) des **`inetd`**, die Sie mithilfe des **`ps`**-Kommandos

(**`ps aux | grep inetd`**) herausfinden können. Diese Vorgehensweise beschränkt sich übrigens nicht nur auf den WU-FTPD, sondern gilt für alle Daemonen, die via **`inetd`** gestartet werden (das sind ziemlich viele).

Die Dateien, die in der o.g. Beispiel-Konfigurationsdatei aufgeführt sind (z. B. unter den Schlüsselwörtern `banner` und `message`) sollten natürlich existieren und einen sinnvollen Inhalt haben. Es handelt sich um Meldungen, die von FTP-Clients angezeigt werden, die auf Ihren FTP-Server zugreifen. Ausführliche Erklärungen zu den einzelnen Konfigurationsoptionen finden Sie in der Manpage von `/etc/ftpaccess` (`man ftpaccess`).

Weitere Kenndaten dieser Konfiguration: Der FTP-Server erlaubt 100 Client-Verbindungen gleichzeitig (darüber hinaus gehende Verbindungsversuche werden mit der Fehlermeldung `dead.msg` abgewiesen). Vor der Anmeldung wird der Text `banner.msg` im Client angezeigt, nach erfolgter Anmeldung `welcome.msg`. Wechselt der Benutzer des Clients in ein anderes Verzeichnis auf dem Server und befindet sich in diesem eine Datei namens `INDEX`, so wird `INDEX` im Client angezeigt. Der Benutzer kann sich mit einem beliebigen Passwort anmelden (Mailadressen lassen sich ohnehin nicht auf Gültigkeit prüfen). Der Transfer von Dateien wird komplett (Up- und Download) in `/var/log/xferlog` protokolliert (geloggt). Außerdem werden sicherheitsrelevante Kommandos des Clients geloggt (z. B. Versuche, Zugriffsrechte von Dateien auf dem Server zu ändern). Existiert die Datei `/etc/shutmsg`, werden Verbindungsversuche unterbunden. Benutzer dürfen keine Dateien auf dem Server löschen, überschreiben, umbenennen oder Zugriffsrechte ändern. Alle Dateien nach dem Schlüsselwort `noretrieve` darf der Client nicht zu sich übertragen. Die Vereinbarungen nach `upload` konfigurieren das für Benutzer schreibbare Verzeichnis `pub/incoming`.

### 4.4.3 ProFTPD (`proftpd`)

Der ProFTPD wird vom ProFTPD Project Team entwickelt und gepflegt. Bei der Entwicklung wurde Wert darauf gelegt, dass die Konfiguration möglichst ähnlich der des bekannten Webservers Apache ist (vgl. Kapitel 5 auf Seite 129). Die gesamte Konfiguration erfolgt in `/etc/proftpd.conf` und bietet wesentlich mehr Möglichkeiten als die des WU-FTPD. So ermöglicht der ProFTPD z. B. eine Beschränkung der Anzahl von Clients je IP, was sich gut gegen einen Missbrauch des FTP-Servers einsetzen lässt.

Das folgende Beispiel entspricht in Sachen Konfiguration weitgehend dem vorangegangenen Beispiel für den WU-FTPD. Der ProFTPD wird allerdings direkt (also nicht via **`inetd`**) gestartet. Das Kommando zum Starten lautet **`rcproftpd start`**. Dieses Kommando startet den Master-Daemon, der dann selbsttätig Kindprozesse für jeden sich verbindenden Client startet.

Die passende Konfigurationsdatei `/etc/proftpd.conf` sehen Sie in Beispiel 4.4.3 auf der nächsten Seite

Zeilen, die mit dem Zeichen # beginnen, sind Kommentare. Zusätzlich zu dem Konfigurationsbeispiel für den WU-FTPD gibt es hier die Beschränkung der maximalen Anzahl der von einer einzelnen IP kommenden Clients. Weiterhin wird

```
# /etc/proftpd.conf - example for ProFTPD
ServerName                  "ftp.meinefirma.de"
ServerType                  standalone
ServerAdmin                 ftpadmin@meinefirma.de
ServerIdent                 on    "MeineFirma FTP Server ready"
DeferWelcome                on
DefaultServer               on
DisplayConnect              msgs/banner.msg
DisplayGoAway               msgs/dead.msg
DisplayLogin                msgs/welcome.msg
AuthPAM                     on
AuthPAMAuthoritative        off
AuthPAMConfig               proftpd
Port                        21
SocketBindTight             on
Umask                       022
User                        nobody
Group                       nogroup

<Directory /*>
  AllowOverwrite            off
  HiddenStor                off
</Directory>

Classes on
Class default limit 100
PathAllowFilter "^[a-zA-Z0-9_.-]+"
PathDenyFilter "(\.ftp)|(\.ht)[a-z]+$"
DenyFilter                  "%"
MaxInstances                110
UseReverseDNS               on
IdentLookups                off
TimeoutStalled              120

ScoreboardPath       /var/run/proftpd
TransferLog          /var/log/xferlog
LogFormat            default "%h %l %u %t \"%r\" %s %b"
ExtendedLog          /var/log/proftpd.log  ALL default
RequireValidShell              no

<Anonymous ~ftp>
  User                    ftp
  Group                   nogroup
  UserAlias               anonymous ftp
  MaxClients              100
  MaxClientsPerHost  2 "Sorry, max. clients per IP: %m"
  DisplayLogin            msgs/welcome.msg
  AuthAliasOnly on
  RequireValidShell off
```

```
   <Directory pub/incoming>
        Umask  017
        <Limit STOR CWD CDUP>
             AllowAll
        </Limit>
        <Limit READ WRITE DIRS RMD DELE MKD>
             DenyAll
        </Limit>
   </Directory>
</Anonymous>
```

Datei 4.4.4: Konfigurationsdatei `/etc/proftpd.conf`

ein Filter für Zeichen vereinbart, die ein Dateiname enthalten darf bzw. nicht enthalten darf. Eine genaue Erklärung der einzelnen Schlüsselworte der ProFTPD-Konfiguration finden sie in */usr/share/doc/packages/proftpd/Configuration.html* sowie auf der im Abschnitt 4.7.2 auf Seite 126 genannten Entwickler-Website.

## 4.5 Grundlegende Sicherheitsaspekte

Im Rahmen dieses Kapitels kann das Thema Sicherheit nicht erschöpfend behandelt werden. Eine Reihe von Dingen, die es zu bedenken gilt, sollen nachfolgend jedoch kurz angerissen werden. Deshalb sei nochmals das separate Kapitel zum Thema Sicherheit 8 auf Seite 155 im vorliegenden Handbuch erwähnt.

Sicherheitsprobleme und -maßnahmen im Zusammenhang mit FTP können direkter (z. B. durch falsch gesetzte Schreibrechte auf einem FTP-Server) als auch indirekter Natur sein (z. B. durch eine Firewall zwischen FTP-Client und FTP-Server).

Direkte Sicherheitsprobleme und -maßnahmen:

- Der „TCP/IP Daemon Wrapper“ **`tcpd`**: Der **`tcpd`** lässt sich einem FTP-Server (und anderen Diensten) als zusätzliche Sicherungsmaßnahme quasi „vorschalten“. Mit dem **`tcpd`** können Sie mittels der Dateien `/etc/hosts.allow` und `/etc/hosts.deny` regeln, welcher Rechner oder welches Subnetz bestimmte Dienste benutzen darf, d. h. Sie können beispielsweise den Zugriff auf Ihren FTP-Server auf Rechner Ihres eigenen Subnetzes beschränken. Weitere Informationen finden Sie in den im letzten Abschnitt genannten Quellen und mit `man 5 hosts_access`.
- Zugriffsrechte auf Dateien und Verzeichnisse: Auf Ihrem FTP-Server im Internet sollten Sie keinen Upload erlauben. Ist dies jedoch zwingend erforderlich, achten Sie peinlichst darauf, dass die Schreibrechte auf ein einzelnes Verzeichnis beschränkt sind, dass keine weiteren Verzeichnisse innerhalb dieses Verzeichnisses angelegt werden können und dass nicht beliebig viel geschrieben werden kann. Außerdem sollten die Inhalte des schreibbaren Verzeichnisses nicht vom FTP-Client angezeigt und übertragen werden können. Ansonsten kann es passieren, dass Ihr FTP-Server ggf. als Zwischen-

station für raubkopierte Software („Warez“) und Pornographie missbraucht wird.

- Sicherheitsrelevante Software-Updates: Bleiben Sie auf dem Laufenden, was Sicherheitslücken in dem von Ihnen im Internet betriebenen FTP-Server angeht. Wenn eine Sicherheitslücke bekannt wird, führen Sie schnellstmöglich ein Update durch. Nichts ist peinlicher als ein „gehackter“ Server. Informationen über Sicherheitslücken und sicherheitsrelevante Software-Updates erhalten Sie von einschlägigen Mailinglisten. Lesen Sie hierzu auch das Kapitel zum Thema Sicherheit 8 auf Seite 155

Indirekte Sicherheitsprobleme und -maßnahmen:

- Firewall zwischen FTP-Server und FTP-Client: Wie im Abschnitt zum FTP-Protokoll beschrieben, versucht der FTP-Server im Falle von aktivem (normalem) FTP eine Verbindung zum FTP-Client zurück aufzubauen. Eine Firewall wird dies in der Regel nicht zulassen. Als Folge davon funktioniert ein FTP-Client hinter einer Firewall erst dann, wenn passives FTP eingeschaltet ist. Der Rückverbindungsversuch des FTP-Servers im Falle von aktivem FTP wird unter Umständen sogar von der Firewall mitprotokolliert und sieht dann wie ein sog. „Portscan“ aus. Mit einem Portscan versucht ein Angreifer, evtl. Schwachstellen an Ihrem Rechner zu finden.
- Passworte: Wie bereits eingangs erwähnt werden Passworte bei FTP unverschlüsselt übertragen. Als Konsequenz daraus sollten Sie im Internet oder in größeren Firmennetzen keine Dateien vertraulichen Inhalts mit FTP übertragen. Lassen Sie als FTP-Administrator nur Anonymous-FTP zu. Reale Benutzer sollten anstelle von FTP **`scp`** oder **`rsync`** via SSH benutzen.

## 4.6 Sonstiges

### 4.6.1 TFTP

Das „Trivial File Transfer Protocol“ (TFTP) ist eines der einfachsten Dateiübertragungsprotokolle überhaupt. TFTP unterscheidet sich von FTP im Wesentlichen in folgenden Punkten:

- TFTP kennt keinen Authentifizierungsmechanismus, d. h. eine Anmeldung mit Login und Passwort erfolgt nicht.
- TFTP benutzt das verbindungslose „User Datagram Protocol“ (UDP) anstelle des verbindungsorientierten „Transmission Control Protocol“ (TCP).
- Befehlsumfang und Konfigurationsmöglichkeiten sind auf ein Minimum reduziert.

TFTP hat, außer seinem minimalistischen Ansatz, keine Vorteile gegenüber FTP und wird in der Regel nur für Nischenanwendungen benutzt. Als Beispiel wären hier X-Terminals anzuführen, die mithilfe von TFTP booten und Fonts laden.

In diesem und ähnlichen Fällen ist TFTP sehr nützlich, da es sich einfach in einem EPROM unterbringen lässt (X-Terminals und ähnliche Netzwerk-Terminals haben in der Regel kein Disketten- oder Plattenlaufwerk).

Einen TFTP-Server (`in.tftpd`) nebst passendem TFTP-Client finden Sie im Paket `tftp`, Serie n. Das TFTP-Protokoll ist in RFC 1350 (und einer Reihe Ergänzungen dazu) beschrieben.

## 4.7 Weiterführende Literatur und Links

Die im folgenden genannten Quellen sind meist in englischer Sprache abgefasst. Die Einarbeitung in das Thema dieses Kapitels ist ohne grundlegende Kenntnisse der englischen Sprache und einer Reihe von Fachtermini leider nur eingeschränkt möglich.

### 4.7.1 Bücher


- Fischer, S., Walther, U.: „Linux Netzwerke“
  SuSE PRESS, Nürnberg, 2000 (ISBN 3-934678-20-3)
- Sery, P.G., Kabir, M.J.: „The SuSE Linux Server“
  Hungry Minds, Inc., 2001 (ISBN 0-7645-4765-8)
- Hunt, C.: „TCP/IP Network Administration“, 2nd Ed.
  O’Reilly & Assoc., Inc., 1998 (ISBN 1-56592-322-7)
- Garfinkel, S., Spafford, G.: „Practical Unix & Internet Security“, 2nd Ed.
  O’Reilly & Assoc., Inc., 1996 (ISBN 1-56592-148-8)


### 4.7.2 WWW-Links

WWW-Links zum Thema FTP:

- ProFTPD Project
  `http://www.proftpd.net/`
- WU-FTPD Development Group
  `http://www.wu-ftpd.org/`
- GNU wget
  `http://www.gnu.org/directory/wget.html`
- gFTP
  `http://gftp.seul.org/`
- IglooFTP
  `http://www.iglooftp.com/unix/`
- FTP mini-HOWTO
  `http://www.linuxdoc.org/HOWTO/mini/FTP.html`

- Lycos Advanced FTP Search
  `http://download.lycos.com/static/advanced_search.asp`
  Suchmaschine für Dateien auf FTP-Servern.

Allgemeine WWW-Links zum Stöbern:

- Woven Goods for Linux
  `http://www.fokus.gmd.de/linux/`
  Information und Dokumentation zu Linux und Linux-Software, u. a. auch zum Thema FTP.
- Linux Documentation Project
  `http://www.linuxdoc.org/`
  Aktuelle Informationen und HOWTOs.
- RFCs
  `ftp://ftp.isi.edu/in-notes/` oder
  `http://www.ietf.org/rfc.html`

# 5 Der Webserver Apache

## 5.1 Einführung

Mit einem Marktanteil von weltweit rund 60 % ist Apache die mit Abstand erfolgreichste Webserver-Software, um Dokumente im Inter- bzw. Intranet per HTTP-Protokoll zur Verfügung zu stellen, und gleichzeitig ein Vorzeigeobjekt der Open-Source-Bewegung.

Während Sie nun mit der bei SuSE Linux mitgelieferten Apache-Version 1.3 das Resultat mehrjähriger Entwicklungsarbeit in Händen halten, befindet sich Apache 2.0, eine grundlegend überarbeite Fassung der Serie 1.3, bereits im fortgeschrittenen Entwicklungsstadium. Neben einer besseren Unterstützung für Nicht-Linux-Systeme soll die Version 2.0 vor allem Mehrprozessor-Rechner deutlich performanter machen; auch die neuen IPv6-Adressen werden nun endlich unterstützt.

Hier möchten wir Ihnen einen Überblick über einige grundsätzliche Anwendungsmöglichkeiten des Servers geben. Möchten Sie als Privatanwender mehr über Apache wissen, sind im Buchhandel verschiedene Titel über den HTTP-Server erschienen, für kommerzielle Projekte steht natürlich auch der SuSE Linux Business Support mit Rat und Tat zur Seite.

## 5.2 Ein erster Schritt

Überprüfen Sie zunächst, ob Apache auf Ihrem System installiert ist. Nachdem dies seit SuSE Linux 7.1 bei neu installierten SuSE-Linux-Systemen nicht mehr standardmäßig der Fall ist, sollten Sie mit dem Befehl **`rcapache status`** überprüfen, ob das Programm im Hintergrund läuft. Bitte beachten Sie, dass Sie hierzu als `'root'` eingeloggt sein müssen.

Falls Apache nicht auf Ihrem Rechner geladen ist (`"Checking for httpd: No process"`), lässt er sich durch die Eingabe von **`rcapache start`** aktivieren.

Sollten Sie bei der Eingabe des obigen Befehls mitgeteilt bekommen, dass das Kommando nicht gefunden wurde (`"bash: rcapache: command not found"`) sollten Sie die Installation mit YaST bzw. YaST2, nachholen. Sie finden den WWW-Server im Paket `apache` der Serie `n`.

Wer regelmäßig auf die Dienste des Apache zurückgreifen muss, kann im Übrigen auch in der Datei `/etc/rc.config` die Variable `START_HTTPD=yes` setzen bzw. diese Option im YaST2 rc.config-Editor entsprechend abändern, dann wird Apache bei jedem Systemstart automatisch geladen.

Dieser einfache Schritt genügt und schon läuft auf Ihrem Rechner ein eigener Webserver. Wenn Sie nun auch noch eine Verbindung zum Internet aufbauen würden, könnte sogar jeder, der Ihre IP-Adresse kennt, darauf zugreifen.

Hat man den Server erst einmal gestartet, steht gleich eine Beispielseite bereit, die sich mit jedem beliebigen Webbrowser (z. B. Konqueror oder Netscape) auf Ihrem Rechner durch die Eingabe von `http://localhost` als Adresse (so heißt standardmäßig der Rechner, an dem Sie jeweils gerade sitzen.) starten lässt.

## 5.3 Background

Nun wird man sich oftmals nicht damit begnügen, Standardseiten von Apache an den Webbrowser ausliefern zu lassen, sondern auch eigene Dokumente, z. B. im eigenen Netz zur Verfügung stellen wollen. Was sich anfangs vielleicht recht kompliziert anhört, ist in Wirklichkeit jedoch sehr einfach zu handhaben.

Normalerweise stammen alle Dateien, die Sie mittels Apache von Ihrem Rechner zur Verfügung stellen, aus dem Verzeichnis `/usr/local/httpd/htdocs/`. Auch das Testdokument, das Sie gerade eben gesehen haben, ist als Datei `index.html` in diesem Verzeichnis gespeichert. Apache sucht standardmäßig nach einer solchen Index-Datei, wenn kein Dateiname angegeben wurde.

Wenn Sie nun beliebige (i. d. R. HTML-) Dokumente nach `/usr/local/httpd/htdocs/` kopieren bzw. in diesem Verzeichnis erstellen, stehen diese auch auf Ihrem eigenen Webserver zum Abruf bereit. Beispielsweise würde der Inhalt der Datei `/usr/local/httpd/htdocs/test.html` ausgeliefert werden, wenn ein Client die URL `http://localhost/test.html` nachfragen würde. Selbstverständlich können Sie beliebige Unterverzeichnisse anlegen, hierfür sind jedoch in aller Regel root-Rechte erforderlich. Bitte beachten Sie, dass Apache auf die von Ihnen abgelegten Dateien zugreifen können muss, es sollte also der Einfachheit halber ein Leserecht für „andere Benutzer“ gesetzt sein.

## 5.4 Konfiguration

Die Konfiguration des Servers erfolgt über Dateien im Verzeichnis `/etc/httpd`, wobei die dort liegende Datei `httpd.conf` eine zentrale Rolle einnimmt. In diesem über 1000 Zeilen langen File werden nicht nur Grundeinstellungen vorgenommen und Zusatzmodule geladen, hier findet auch der Großteil der darüber hinaus gehenden Konfigurationsparameter seinen Platz.

Die bei SuSE Linux mitgelieferte Version von Apache ist bereits für viele Anwendungsmöglichkeiten vorkonfiguriert und ausführlich dokumentiert, so dass wir an dieser Stelle nur auf einige Ausschnitte in der Datei eingehen möchten:

## 5.5 Virtual Hosts

Eines der wichtigsten und beliebtesten Features des Apache ist die Möglichkeit, mit Hilfe von so genannten *virtual hosts* mehrere IP-Adressen oder Domains von ein- und demselben Server aus zu versorgen, wobei der Eindruck entsteht, es handle sich jeweils um komplett unterschiedliche Maschinen.

Doch bevor man nun Apache so konfiguriert, dass er Inhalt für 10 000 Domains auf einmal ausliefert, muss überlegt werden, ob für jeden virtuellen Host eine IP-Adresse zur Verfügung steht („IP-based virtual hosts“) oder ob sich mehrere virtuelle Angebote eine Adresse teilen müssen („Name-based virtual hosts“).

Prinzipiell gibt es mit dem neuen HTTP-1.1-Protokoll die Möglichkeit, mehrere virtuelle Hosts auf einer IP-Adresse laufen zu lassen und nur anhand des eingegebenen Namens zu entscheiden, ob nun der Inhalt der einen oder der anderen Seite ausgeliefert werden soll. Dies ist vor allem immer dann notwendig, wenn mehr Virtuelle Hosts eingerichtet werden sollen als IP-Adressen vorhanden sind.

Sind hingegen ausreichend IP-Adressen vorhanden, liegt es nahe, jedem Virtuellen Host eine eigene Nummer zuzuweisen, damit auch bei (sehr seltenen) Anfragen von älteren Browsern, die kein HTTP 1.1 „sprechen“, klar ist, welche Seite es denn nun sein darf.

Übrigens muss nicht unbedingt für jede IP-Adresse eine separate Netzwerkkarte erworben und konfiguriert werden. Es ist durchaus möglich, einer (physikalischen) Netzwerkkarte mehrere Adressen zuzuweisen, indem man hinter den Namen des Gerätes einen Doppelpunkt sowie eine Zahl einfügt. So könnte eine Netzwerkkarte eth0 mit Adresse `192.168.1.1` durchaus auch als `eth0:0` unter `192.168.1.2` im Netz zur Verfügung stehen.

Im folgenden Beispiel möchten wir unseren Apache dazu bringen, parallel zu `erde.suse.de` auch unter `mond.suse.de` Anfragen entgegen zu nehmen. Dafür könnte man der Datei `/etc/httpd/httpd.conf` z. B. folgenden Eintrag hinzufügen:

```
<VirtualHost 192.168.1.22>
 DocumentRoot /www/docs/mond
 ServerAdmin webmaster@mond.suse.de
 ServerName mond.suse.de
 CustomLog /var/log/host.mond.suse.de-access_log common
 ErrorLog /var/log/host.mond.suse.de-error_log
</VirtualHost>
```

Datei 5.5.1: Die Datei `httpd.conf`

Nachdem in der ersten Zeile die Adresse des *virtual host* definiert wird, folgen einige grundsätzliche Informationen zu diesem:

- Unter *DocumentRoot* wird das Verzeichnis angegeben, in dem sich die (HTML-) Dateien für den virtuellen Host `mond.suse.de` befinden. Wenn nun ein Client eine Anfrage unter dieser Adresse stellt, wird Apache versuchen, im Verzeichnis `/www/docs/mond` eine `index.html`-Datei zu finden und auszuliefern.

- Mit *ServerAdmin* wird die E-Mail-Adresse des Server-Administrators definiert. Diese wird z. B. auch dann angezeigt, wenn eine Seite unter `http://mond.suse.de` nicht auffindbar ist.
- Durch *ServerName* wird der Domain-Name des virtuellen Hosts definiert. Dieser sollte auch im DNS entsprechend eingetragen sein.
- Die obigen Einstellungen bei `CustomLog` führen dazu, dass Apache in der Datei `/var/log/host.mond.suse.de-access_log` ein Protokoll über alle Zugriffe auf diesen virtuellen Server führt.
- *ErrorLog* gibt die Datei an, in die evtl. auftretende Fehler mitprotokolliert werden.

## 5.6 Weitere Informationen

Als Einstieg zu Apache und das dahinter stehende Projekt empfehlen wir die Seite `http://httpd.apache.org/`. Dort finden Sie stets aktuelle Informationen rund um die Apache-Entwicklung sowie auch unzählige FAQs und Anleitungen. Sollten Sie an zusätzlichen Modulen für Ihren Webserver interessiert sein, wäre `http://modules.apache.org` ein guter Anfang.

Auch die Anleitung des HTTP-Servers unter `/usr/share/doc/packages/apache/manual/index.html` könnte sich als nützlich erweisen. Für wirklich tiefgreifende Fragen und höhere Anforderungen sind inzwischen mehrere Bücher erschienen, wobei hier vor allem das im O'Reilly-Verlag erschienene „Apache: The Definitive Guide“ zu nennen wäre, das von einigen Apache-Kern-Programmierern geschrieben wurde.

Inzwischen existiert mit ApacheWeek (`http://www.apacheweek.com`) sogar ein wöchentlicher Info-Newsletter, der nicht nur die neuesten Infos rund um den Fortschritt des Projektes in die Benutzer-Gemeinde bringt, sondern auch immer das Neueste von verschiedenen Apache-Kongressen zu berichten weiß.

# 6 Dokumentationsserver: SuSE Hilfe im Netz

In einem Netzwerk ist es vorteilhaft, umfangreiche oder seltener benutzte Dokumentationen und die zugehörigen Indizes (Suchdatenbanken) zentral auf einem Dokumentationsserver zu halten. Diese Funktionalität (früher im Paket `dochost`, Serie `n`) ist jetzt im Paket Paket `susehelp`, Serie `doc` integriert. Die folgenden Abschnitte zeigen, wie die SuSE Hilfe für Dokumentationsserver und -Client konfiguriert wird.

Wie Sie mit der *SuSE Hilfe* arbeiten, lesen Sie im Buch *Konfiguration* im Kapitel *Die SuSE Hilfe*.

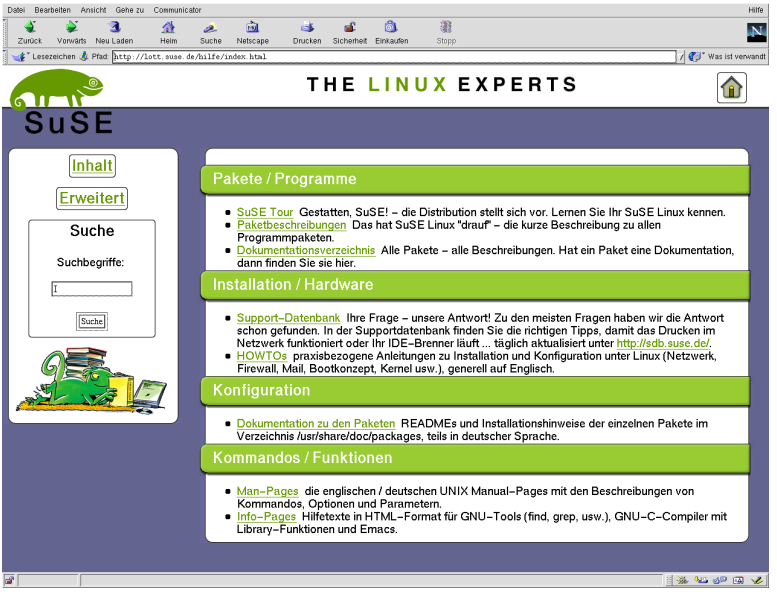



Abbildung 6.1: Die Startseite der SuSE Hilfe (Browser Netscape)

## 6.1 Vorbemerkungen zu Apache und susehelpcenter

Die SuSE Hilfe verwendet für ihre Suchfunktion einen Webserver. Für lokale Anwendung (Einzelplatzsystem) leistet der im Paket `susehelpcenter`, Serie

k2de implementierte Wizard optimale Serverdienste; es ist also nicht zu empfehlen für diese Zwecke den Apache Webserver zu installieren, der die Dokumente auch für externen Zugriff zugänglich macht.

Soll für die SuSE Hilfe ein Dokumentationsserver eingerichtet werden, ist hier die Verwendung eines echten Webservers (z. B. Apache) unverzichtbar.

Die Installation von Paket susehelpcenter, Serie k2de auf dem Server ist zusätzlich zu empfehlen, nicht nur für komfortablen lokalen Zugriff, sondern auch für die menügesteuerte Hilfe-Konfiguration (zu erreichen über das KDE-Kontrollzentrum, siehe Abschnitt *Hilfe* im Buch *Konfiguration*).

## 6.2 Die Dokumentationsquellen / Pakete

Im Nachfolgenden geben wir Ihnen eine Übersicht über die Dokumentationspakete, die in die SuSE Hilfe integriert werden können.

- Paket howtodeh, bzw. howtoenh, Serie doc: Die deutschsprachigen bzw. englischen Howto-Dokumente sind praxisbezogene Anleitungen zu Installation und Konfiguration von Linux. Die englischen sind aufgrund ihres größeren Umfangs und ihrer Aktualität vorzuziehen.

- Paket sdb, sdb_de, Serie doc: Die Support-Datenbank (SDB) ist auch auf unserer CD enthalten, damit Sie nicht immer online zugreifen müssen.

- Paket susetour_de, Serie doc: Die SuSE-Tour gibt einen Überblick über einige unserer Programme.

- Paket applbook_de, confbook_de, netbook_de, refbook_de, Serie doc: Die SuSE-Handbücher im HTML-Format.

- Paket rman, Serie ap: Das Paket wird für den Webserver zur Umwandlung der Manual-Pages in das HTML-Format benötigt. Die Dokumente werden „on-the-fly“ konvertiert.

- Paket inf2htm, Serie doc: Das Paket wird für den Webserver zur Umwandlung der *Info-Pages* in das HTML-Format benötigt. Die Dokumente werden „on-the-fly“ konvertiert.

Wenn Sie weitere Dokumentationen (z. B. von Drittanbietern) über die SuSE Hilfe erreichen wollen, sollten Sie dafür Konfigurationsdateien erstellen, analog zu denen in /etc/susehelp.d/*.conf. Weitere Einzelheiten hierzu gibt’s in /usr/share/doc/packages/susehelp/README.

## 6.3 Einrichten des Dokumentationsservers

1. Stellen Sie sicher, dass folgende Pakete installiert sind:

- Paket `susehelp`, Serie `doc`. Das zentrale Paket der SuSE Hilfe stellt Suchfunktionen in den installierten Dokumentationsquellen bereit und Routinen zur Darstellung von Paketbeschreibungen, Man-Pages, Info-Pages und den Programmdokumentationen in `/usr/share/doc/packages/`.
- Paket `apache`, Serie `n`, vergleiche Abschnitt 6.1 auf Seite 133
- Paket `susehelpcenter`, Serie `k2de`, vergleiche Abschnitt 6.1 auf Seite 133
- die Dokumentationsquellen Ihrer Wahl, vergleiche Abschnitt 6.2 auf der vorherigen Seite

2. Ändern Sie in der Datei `/etc/rc.config` die Variablen wie in Datei 6.3.1 angegeben. Statt `sonne.kosmos.all` und `kosmos.all` tragen Sie natürlich den Namen Ihres Rechners, bzw. Ihrer Domain ein.

```
START_INETD="yes"
START_HTTPD="yes"
DOC_SERVER="yes"
DOC_HOST="sonne.kosmos.all "
DOC_ALLOW="LOCAL .kosmos.all "
```

Datei 6.3.1: `/etc/rc.config` für Dokumentationsserver

Im Anschluss ein paar Erläuterungen zu den Einträgen in der Datei `/etc/rc.config`.

**`START_INETD`** Das Programm inetd (engl. *inet daemon*) wird u. a. für den Zugriff auf die Manual-Page im HTML-Format via http-rman benötigt.

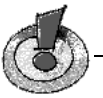
**`START_HTTPD`** Um den Webserver (z. B. Apache) bei jedem Booten automatisch zu starten, muss der Wert bei **`START_HTTPD`** auf `yes` stehen.

**`DOC_SERVER`** Der Eintrag legt fest, von welchem Rechner die Dokumente zur Verfügung gestellt werden sollen. Die Variable für den **`DOC_SERVER`** muss für den Server auf den Wert `yes` und für Clients auf den Wert `no` gesetzt werden.

**`DOC_HOST`** Der Eintrag legt den Namen des Dokumentationsservers fest. In unserem Beispiel `sonne.kosmos.all`.

**`DOC_ALLOW`** Hier geht es um die Sicherheit des Systems: Es sind die Rechner bzw. Domains einzutragen, denen Zugriff auf die Manual-Pages gestattet werden soll. Wenn Sie den Zugriff für eine komplette Domain freigeben wollen, vergessen Sie nicht den führenden Punkt `'.'` vor dem Namen.

Am einfachsten nehmen Sie die Änderungen über YaST vor. Wenn Sie die Datei „händisch“ ändern, vergessen Sie nicht, abschließend SuSEconfig zu starten. SuSEconfig stößt dann im Hintergrund das Programm zum Aufbau der Suchdatenbanken an. Das kann eine Weile dauern.



Hinweis

Die *lokale Hilfe* kann über susehelpcenter immer gestartet werden, unabhängig von der Konfiguration des Rechners als Dokumentationsserver oder -Client.

Über die menügesteuerte Hilfe-Konfiguration (zu erreichen über das KDE-Kontrollzentrum, siehe Abschnitt *Hilfe* in dem Buch *Konfiguration*) können Sie einzelne Dokumentationsquellen von der Suche ausschließen, falls Sie dafür keine Indizes erstellen lassen möchten, bzw. einbeziehen. Ihr „maßgeschneidertes“ Hilfe-System können Sie zusammenstellen, wenn Sie die Konfigurationsdateien `/etc/susehelp.d/*.conf` speziell Ihren Bedürfnissen anpassen. Lesen Sie dazu die Beschreibungen in `/usr/share/doc/packages/susehelp/README`.

## 6.4 Konfiguration für einen Client-Rechner

Auf den Clients die auf einen Dokumentationsserver zugreifen sollen, installieren Sie das Paket `susehelp`, Serie `doc` und ändern in `/etc/rc.config` die Variablen wie in Datei 6.4.1.

Ersetzen Sie hierbei `sonne.kosmos.all` durch den vollständigen Namen Ihres Dokumentationsservers (FQDN - engl. Fully Qualified Domain Name). Die Einstellungen funktionieren nur dann, wenn die Dokumentation tatsächlich auf `sonne.kosmos.all` installiert ist. Vgl. auch die Erläuterungen zu Datei 6.3.1 auf der vorherigen Seite.

```
DOC_SERVER="no"
DOC_HOST="sonne.kosmos.all"
DOC_ALLOW=""
```

Datei 6.4.1: `/etc/rc.config` für einen Client-Rechner

Jetzt können Sie das SuSE Hilfesystem des Dokumentationsservers starten. Geben Sie in der Konsole folgendes ein:

```
erde:~ # hilfe
```

# 7 Proxy-Server: Squid

Im folgenden Kapitel wird erläutert, wie das Caching von Webseiten mit Hilfe eines Proxy-Servers funktioniert und welchen Nutzen Squid für Ihr System bietet.

Squid ist der am weitesten verbreitete Proxy-Cache für Linux/UNIX-Plattformen. Wir werden beschreiben, wie er zu konfigurieren ist, welche Systemanforderungen bestehen, wie das eigene System konfiguriert sein muss, um transparentes Proxying durchzuführen, wie man Statistiken über den Nutzen des Cache mithilfe von Programmen wie Calamaris und cachemgr erhält oder wie man Web-Inhalte mit squidgrd filtert.

## 7.1 Was ist ein Proxy-Cache?

Squid fungiert als Proxy-Cache. Es verhält sich wie ein Makler, der Anfragen von Clients erhält (in diesem Fall Web-Browser) und an den zuständigen Server-Provider weiterleitet. Wenn die angeforderten Objekte beim Vermittler ankommen, behält er eine Kopie davon in einem Festplatten-Cache.

Der Vorteil zeigt sich, wenn mehrere Clients dasselbe Objekt anfordern: Sie können nun direkt aus dem Festplatten-Cache bedient werden, also wesentlich schneller als aus dem Internet. Dies spart gleichzeitig eine Menge Systembandbreite.

Tipp

Squid bietet ein großes Spektrum an Features, z. B. die Festlegung von Hierarchien für die Proxy-Server zum Verteilen der Systemlast, Aufstellen fester Zugriffsregeln an alle Clients, die auf den Proxy zugreifen wollen, Erteilen oder Verweigern von Zugriffsrechten auf bestimmte Webseiten mit Hilfe anderer Applikationen oder die Ausgabe von Statistiken der meistbesuchten Webseiten, wie z. B. das Surfverhalten der Benutzer u. v. m.

Squid ist kein generischer Proxy. Normalerweise vermittelt er nur zwischen HTTP-Verbindungen. Außerdem unterstützt er die Protokolle FTP, Gopher, SSL und WAIS, jedoch keine anderen Internet-Protokolle wie Real Audio, News oder Videokonferenzen. Squid greift auf das UDP-Protokoll nur zur Unterstützung der Kommunikation zwischen verschiedenen Caches zurück. Aus diesem Grund werden auch andere Multimedia-Programme nicht unterstützt.

## 7.2 Informationen zu Proxy-Cache

### 7.2.1 Squid und Sicherheit

Man kann Squid zusammen mit einer Firewall verwenden, um interne Netzwerke durch den Einsatz von Proxy-Cache nach außen zu schützen. Die Firewall verweigert mit Ausnahme von Squid alle externen Dienste, alle WWW-Verbindungen müssen durch den Proxy aufgebaut werden.

Im Falle einer Firewall-Konfiguration mit einem DMZ würden wir dort unseren Proxy setzen. In diesem Fall ist es wichtig, dass alle Rechner im DMZ ihre Protokolldateien an Rechner innerhalb des gesicherten Netzwerks senden.

Ein Möglichkeit der Implementierung dieser Features mit Hilfe eines so genannten „transparenten“ Proxy wird in Abschnitt 7.6 auf Seite 147 behandelt.

### 7.2.2 Mehrere Caches

Man kann mehrere Caches so konfigurieren, dass Objekte zwischen ihnen ausgetauscht werden können, um die Systemlast zu reduzieren und die Möglichkeit zu steigern, ein bereits im lokalen Netzwerk vorhandenes Objekt zu finden. Möglich sind auch Cache-Hierarchien, so dass ein Cache in der Lage ist, Objektanfragen an Caches der gleichen Hierarchie weiterzuleiten oder einen übergeordneten Cache zu veranlassen, die Objekte von einem anderen Cache im lokalen Netzwerk oder direkt aus der Quelle herunterzuladen.

Die Wahl der richtigen Topologie für die Cache-Hierarchie ist sehr wichtig, da Netzwerkverkehr insgesamt nicht erhöht werden soll. In einem großen Netzwerk z. B. ist es möglich, für jedes Subnetz einen Proxy-Server zu konfigurieren und diesen dann mit einem übergeordneten Proxy zu verbinden, der wiederum an den Proxy-Cache vom ISP angeschlossen wird.

Die gesamte Kommunikation wird vom ICP (engl. *Internet Cache Protocol*) gesteuert, das auf dem UDP-Protokoll aufgesetzt ist. Der Datenaustausch zwischen Caches geschieht mittels HTTP (engl. *Hyper Text Transmission Protocol*) basierend auf TCP. Allerdings sollten für solche Verbindungen schnellere und einfachere Protokolle verwendet werden, die innerhalb von maximal einer oder zwei Sekunden auf eingehende Anfragen reagieren können.

Um den besten Server für die gewünschten Objekte zu finden, schickt ein Cache an alle Proxies der gleichen Hierarchie eine ICP-Anfrage. Die Proxies werden mittels ICP-Antworten mit dem Code „HIT“ auf die Anfragen reagieren, falls das Objekt gefunden wurde oder, falls nicht, mit dem Code „MISS“. Im Falle mehrerer HIT-Antworten wird der Proxy-Server einen Server für das Herunterladen bestimmen. Diese Entscheidung wird unter anderem dadurch bestimmt, welcher Cache die schnellste Antwort sendet oder welcher näher ist. Bei einer nicht zufrieden stellenden Antwort gesendet wurde, wird die Anfrage an den übergeordneten Cache geschickt.

Tipp

Zur Vermeidung von mehrfacher Speicherung von Objekten in verschiedenen Caches unseres Netzwerks werden andere ICP-Protokolle verwendet, wie z. B. CARP (engl. *Cache Array Routing Protocol*) oder HTCP (engl. *Hyper-Text Cache Protocol*).
Je mehr Objekte sich im Netzwerk befinden, desto leichter wird es, das Gesuchte zu finden.

### 7.2.3 Zwischenspeichern von Internetobjekten

Nicht alle im Netzwerk verfügbaren Objekte sind statisch. Es existieren viele dynamisch generierte CGI-Seiten, Zugriffszähler oder verschlüsselte SSL-Dokumente für eine höhere Sicherheit. Aus diesem Grund werden solche Objekte nicht im Cache gehalten: Bei jedem neuen Zugriff hat sich das Objekt bereits wieder verändert.

Für alle anderen im Cache befindlichen Objekte stellt sich jedoch die Frage, wie lange sie dort bleiben sollen. Für diese Entscheidung werden alle Objekte im Cache drei verschiedenen Stadien zugeordnet:

1. **FRESH:** Wenn dieses Objekt angefordert wird, wird es gesendet, ohne dass ein Abgleich mit dem Originalobjekt im Web stattfindet.
2. **NORMAL:** Der Server, von dem das Objekt ursprünglich stammt, wird daraufhin überprüft, ob sich das Objekt geändert hat. Falls dies der Fall ist, wird die Kopie im Cache aktualisiert.
3. **STALE:** Das Objekt wird als veraltet angesehen und wird neu vom Server heruntergeladen.

Durch Header wie „Last modified“ („zuletzt geändert“) oder „Expires“ („läuft ab“) und dem entsprechenden Datum informieren sich Web- und Proxy-Server über den Status eines Objekts. Es werden auch andere Header verwendet, die z. B. anzeigen, dass ein Objekt nicht zwischengespeichert werden muss.

Objekte im Cache werden normalerweise aufgrund fehlenden Speichers ersetzt, und zwar durch Algorithmen wie LRU (engl. *Last Recently Used*), die zum Ersetzen von Cache-Objekten entwickelt wurden. Das Prinzip besteht im Wesentlichen darin, zuerst die am seltensten gewünschten Objekte zu ersetzen.

## 7.3 Systemanforderungen

Zuerst sollte die maximale Systemlast bestimmt werden. Es ist wichtig, den Systemspitzen besondere Aufmerksamkeit zu schenken, da diese mehr als viermal so hoch wie der Tagesdurchschnitt sein können. Im Zweifelsfall ist es besser, die

Systemanforderungen zu überschätzen, vorausgesetzt, dass ein am Limit arbeitender Squid zu einem ernsthaften Qualitätsverlust des Dienstes führen kann.

Geordnet nach Wichtigkeit werden in den folgenden Abschnitten die verschiedenen Systemfaktoren aufgezeigt.

### 7.3.1 Festplatte

Für das Zwischenspeichern spielt Geschwindigkeit eine hohe Rolle. Man sollte sich also um diesen Faktor besonders kümmern. Bei Festplatten ist dieser Parameter als „zufällige Positionierzeit“ in Millisekunden beschrieben. Als Faustregel gilt: Je niedriger dieser Wert, desto besser. Für eine hohe Geschwindigkeit empfiehlt es sich, schnelle Festplatten zu wählen.

Nach dem Squid-Benutzer-Guide (`http://www.squid-cache.org`) ist bei Systemen mit nur einer Festplatte die Formel für die Berechnung der Anzahl von Anfragen pro Sekunde von der Positionierzeit der Festplatten ganz einfach:

Anfragen pro Sekunde = 1000 / Positionierzeit

Squid erlaubt die gleichzeitige Verwendung von mehreren Festplatten und damit eine höhere Anzahl von Anfragen pro Sekunde. Hat man z. B. drei Festplatten mit der gleichen Positionierzeit von 12 Millisekunden, ergibt sich unter Verwendung der vorherigen Formel folgendes:

Anfragen pro Sekunde = 1000 / (Positionierzeit / Anzahl der Festplatten) =
1000 / (12/3) = 250 Anfragen pro Sekunde

Im Vergleich zum Einsatz von IDE-Festplatten sind SCSI-Festplatten zu bevorzugen. Allerdings haben neuere IDE-Festplatten ähnliche Positionierzeiten wie SCSI und zusammen mit DMA-kompatiblen IDE-Controllern erreichen sie eine ähnliche Geschwindigkeit für den Datentransfer ohne dabei die Systemlast beträchtlich zu steigern.

#### Größe des Festplatten-Cache

In einem kleinen Cache ist die Wahrscheinlichkeit eines HIT (das gewünschte Objekt befindet sich bereits dort) sehr gering, da der Cache schnell gefüllt sein wird. In diesem Fall werden die selten gewünschten Objekte durch neue ersetzt. Steht jedoch `1 GB` für den Cache zur Verfügung und die Benutzer benötigen nur `10 MB` pro Tag zum Surfen, dann dauert es mehr als hundert Tage, bis der Cache voll ist.

Am leichtesten lässt sich die Größe des Cache durch die maximale Übertragungsrate der Verbindung bestimmen. Mit einer Verbindung von `1 MB/Sek` wird die maximale Übertragungsrate bei `125 KB/Sek` liegen. Landet der gesamte Datenverkehr im Cache, kommen innerhalb einer Stunde `450 MB` zusammen. Wenn man nun annimmt, dass der gesamte Datenverkehr lediglich während acht Arbeitsstunden erzeugt wird, erreicht man innerhalb eines Tages `3,6 GB`. Da die Verbindung nicht bis zur Kapazitätsgrenze ausgeschöpft wurde, konnten wir davon ausgehen, dass die gesamte Datenmenge, die durch den Cache geht, bei

ungefähr 2 GB liegt. In unserem Beispiel werden 2 GB Speicher für Squid benötigt, um die Daten aller aufgerufenen Seiten *eines* Tages im Cache zu halten.

Zusammenfassend lässt sich sagen, dass Squid dazu tendiert, kleinere Datenblöcke von der Festplatte zu lesen oder darauf zu schreiben, so dass es wichtiger ist, wie schnell er diese Objekte auf der Festplatte findet, als eine Festplatte mit hohem Durchsatz zu haben.

### 7.3.2 RAM

Der von Squid benötigte Speicher ist abhängig von der Anzahl der im Cache zugewiesenen Objekte. Squid speichert Cache-Objektverweise und häufig angeforderte Objekte zusätzlich im Speicher, damit diese Daten schneller abgefragt werden können. Der Speicher ist eine Million mal schneller als eine Festplatte!

Jedes Objekt im RAM-Speicher hat eine Größe von 72 Byte (für „kleine“ Pointer-Architekturen wie **Intel**, **Sparc**, **MIPS**; für Alpha sind es 104 Byte), wenn die Durchschnittsgröße eines Objekts im Internet ungefähr 8 KB beträgt und wir 1 GB Festplattenspeicher für den Cache haben, werden wir ungefähr 130.000 Objekte speichern, was alleine für die Meta-Daten fast 10 MB RAM ergibt.

Squid hält auch andere Daten im Speicher, z. B. eine Tabelle mit allen vergebenen IP-Adressen, einen genau festgelegten Domainnamen-Cache, die am häufigsten gewünschten Objekte, Puffer, Zugriffskontrolllisten, etc.

Es ist sehr wichtig, dass ausreichend Speicher für den Squid-Prozess zur Verfügung steht. Sollte er ausgelagert werden müssen, wird sich die Systemleistung nämlich drastisch reduzieren. Für die Cache-Speicherverwaltung wird das Tool cachemgr.cgi verwendet. Es wird im Abschnitt 7.7.1 auf Seite 150 erläutert.

### 7.3.3 CPU

Das Programm Squid benötigt nicht viel CPU. Nur beim Start und während der Überprüfung des Cache-Inhalts ist die Prozessorlast höher. Der Einsatz eines Multiprozessorrechners steigert nicht die Systemleistung. Zur Effektivitätssteigerung ist es besser, schnellere Festplatten zu verwenden oder mehr Speicher hinzuzufügen.

Einige Beispiele von konfigurierten Systemen, auf denen Squid läuft, finden sich unter http://wwwcache.ja.net/servers/squids.html .

## 7.4 Squid starten

Der Squid auf SuSE Linux ist bereits soweit vorkonfiguriert, dass man ihn problemlos sofort nach der Installation starten kann. Als Voraussetzung für einen reibungslosen Start sollte das Netzwerk soweit konfiguriert sein, dass mindestens ein Nameserver und sinnvollerweise auch das Internet erreichbar sind. Probleme kann es bereiten, wenn man eine Wählverbindung mit dynamischer

DNS-Konfiguration verwendet. In so einem Fall sollte mindestens der Nameserver fest eingetragen sein, da Squid erst gar nicht startet, wenn er in der `/etc/resolv.conf` keinen DNS findet.

Um Squid zu starten, gibt man auf der Kommandozeile (als `'root'`)

```
rcsquid start
```

ein. Beim ersten Mal wird zunächst die Verzeichnisstruktur in `/var/squid/cache` angelegt. Dies wird vom Startskript `/etc/init.d/squid` automatisch durchgeführt und kann ein paar Sekunden bis Minuten dauern. Erscheint rechts in grün `done`, wurde Squid erfolgreich gestartet. Auf dem lokalen System kann man die Funktionsfähigkeit von Squid sofort testen, indem man im Browser als Proxy `localhost` und `Port 3128` einträgt. Um den Zugriff auf Squid und somit das Internet für alle zu ermöglichen, braucht man in der Konfigurationsdatei `/etc/squid.conf` lediglich den Eintrag `http_access deny all` auf `http_access allow all` zu ändern. Allerdings sollte man dabei bedenken, dass man den Squid damit komplett für jedermann öffnet. Von daher sollte man unbedingt so genannte „ACL's" definieren, die den Zugriff auf den Proxy regeln. Dazu mehr im Abschnitt 7.5 auf Seite 145 .

Hat man Änderungen an der Konfigurationsdatei `/etc/squid.conf` vorgenommen, muss man Squid dazu bringen, diese neu einzulesen. Das gelingt mit:

```
rcsquid reload
```

Alternativ kann man Squid auch komplett neu starten:

```
rcsquid restart
```

Wichtig ist noch folgendes Kommando:

```
rcsquid status
```

Damit kann man feststellen, ob der Proxy läuft und mit

```
rcsquid stop
```

wird Squid beendet. Letzteres kann eine Weile dauern, da Squid bis zu einer halben Minute (`shutdown_lifetime`) wartet, bevor die Verbindungen zu den Clients unterbrochen werden und er dann noch seine Daten auf Platte schreiben muss. Beendet man Squid mit einem **`kill`** oder **`killall`**, kann das einen zerstörten Cache zur Folge haben, den man dann löschen muss, um Squid wieder starten zu können.

Beendet sich Squid nach kurzer Zeit, obwohl er scheinbar erfolgreich gestartet wurde, kann das an einem fehlerhaften Nameserver-Eintrag oder einer fehlenden `/etc/resolv.conf` liegen. Den Grund für einen gescheiterten Start protokolliert Squid dabei in der Datei `/var/squid/logs/cache.log` . Soll Squid bereits beim Booten automatisch gestartet werden, braucht man in `/etc/rc.config` lediglich den Eintrag `START_SQUID=no` auf `START_SQUID=yes` abzuändern.

Bei einer Deinstallation von Squid werden weder Cache noch Log-Dateien entfernt. Man muss das Verzeichnis `/var/squid` manuell löschen.

#### Lokaler DNS-Server

Einen lokalen DNS-Server wie **`BIND-8`** oder **`BIND-9`** aufzusetzen, ist durchaus sinnvoll, auch wenn er keine eigene Domain verwaltet. Er fungiert dann lediglich

als „Caching-only DNS“ und kann ohne spezielle Konfiguration DNS-Anfragen über die Root-Nameserver auflösen. Trägt man diesen in der `/etc/resolv.conf` mit der IP-Adresse `127.0.0.1` für **`localhost`** ein, findet Squid beim Starten immer einen gültigen Nameserver. Es reicht aus, das Paket zu installieren und BIND zu starten. Den Nameserver des Providers sollte man in der Konfigurationsdatei `/etc/named.conf` unter **`forwarders`** mit seiner IP-Adresse eintragen. Falls man eine Firewall laufen hat, und sei es nur die Personal-Firewall, muss man aber darauf achten, dass die DNS-Anfragen auch durchgelassen werden.

## 7.5 Die Konfigurationsdatei /etc/squid.conf

Alle Einstellungen zum Squid Proxyserver sind in der Datei `/etc/squid.conf` vorzunehmen. Um Squid erstmalig starten zu können, sind darin keine Änderungen erforderlich, der Zugriff von externen Clients ist jedoch erst einmal gesperrt. Für **`localhost`** ist der Proxy freigegeben und als Port wird standardmäßig `3128` verwendet. Die Optionen sind ausführlich und mit vielen Beispielen in der vorinstallierten `/etc/squid.conf` dokumentiert. Annähernd alle Einträge sind am Zeilenanfang durch ein #-Zeichen auskommentiert und immer am Ende der zugehörigen Beschreibung zu finden. Die angegebenen Werte entsprechen fast immer den voreingestellten Werten, so dass das Entfernen des Kommentarzeichens, ohne den Parameter der Option zu ändern, bis auf wenige Ausnahmen keine Wirkung hat. Besser ist es, das Beispiel stehen zu lassen und die Option mit dem geänderten Parameter in der Zeile darunter neu einzufügen. So kann man die voreingestellten Werte und Änderungen problemlos nachvollziehen.

Hat man ein Update von einer älteren Squid-Version durchgeführt, ist es unbedingt zu empfehlen, die neue `/etc/squid.conf` zu verwenden und nur die Änderungen von der ursprünglichen Datei zu übernehmen. Versucht man die alte `squid.conf` weiter zu verwenden, läuft man Gefahr, dass die Konfiguration nicht mehr funktioniert, da Optionen immer wieder geändert werden und neue hinzukommen.

#### Allgemeine Konfigurations-Optionen

**http_port 3128** Das ist der Port, auf dem Squid auf Anfragen der Clients lauscht. Voreingestellt ist `3128`, gebräuchlich ist auch `8080`. Es ist möglich, hier mehrere Portnummern, durch Leerzeichen getrennt, anzugeben.

**cache_peer <hostname> <type> <proxy-port> <icp-port>** Hier kann man einen übergeordneten Proxy als „Parent“ eintragen, z. B. wenn man den des Providers nutzen will oder muss. Als `<hostname>` trägt man den Namen bzw. die IP-Adresse des zu verwendenden Proxies und als `<type>` `parent` ein. Für `<proxy-port>` trägt man die Portnummer ein, die der Betreiber des Parent auch zur Verwendung im Browser angibt, meist `8080`. Den `<icp-port>` kann man auf `7` oder `0` setzen, wenn man den ICP-Port des Parent nicht kennt und die Benutzung dieses mit dem Provider nicht vereinbart wurde. Zusätzlich sollte man dann noch `default` und `no-query` nach den Port-

nummern angeben, um die Verwendung des ICP-Protokolls ganz zu unterbinden. Squid verhält sich dann gegenüber dem Proxy des Providers wie ein normaler Browser.

**cache_mem 8 MB** Dieser Eintrag gibt an, wie viel Arbeitsspeicher von Squid für das Cachen maximal verwendet wird. Voreingestellt sind `8 MB`.

**cache_dir ufs /var/squid/cache 100 16 256** Der Eintrag `cache_dir` gibt das Verzeichnis an, in dem alle Objekte auf Platte abgelegt werden. Die Zahlen dahinter geben den maximal zu verwendenden Plattenplatz in `MB` und die Anzahl der Verzeichnisse in erster und zweiter Ebene an. Den Parameter `ufs` sollte man unverändert lassen. Voreingestellt sind `100 MB` Plattenplatz im Verzeichnis `/var/squid/cache` zu belegen und darin 16 Unterverzeichnisse anzulegen, die jeweils wiederum 256 Verzeichnisse enthalten. Bei Angabe des zu verwendenden Plattenplatzes sollte man genügend Reserven lassen, sinnvoll sind Werte zwischen 50 und maximal 80 Prozent des verfügbaren Platzes. Die beiden letzten Zahlen für die Anzahl der Verzeichnisse sollte man nur mit Vorsicht vergrößern, da zu viele Verzeichnisse auch wieder auf Kosten der Performance gehen können. Hat man mehrere Platten, auf die der Cache verteilt werden soll, kann man entsprechend viele `cache_dir`-Zeilen eintragen.

**cache_access_log /var/squid/logs/access.log** Pfadangabe für Log-Datei.

**cache_log /var/squid/logs/cache.log** Pfadangabe für Log-Datei.

**cache_store_log /var/squid/logs/store.log** Pfadangabe für Log-Datei.
Diese drei Einträge geben den Pfad zur Protokolldatei von Squid an. Normalerweise wird man daran nichts ändern. Wird der Squid stark beansprucht, kann es sinnvoll sein, den Cache und die Log-Dateien auf verschiedene Platten zu legen.

**emulate_httpd_log off** Ändert man diesen Eintrag auf `on`, erhält man lesbare Log-Dateien. Allerdings kommen manche Auswerteprogramme damit nicht zurecht.

**client_netmask 255.255.255.255** Mit diesem Eintrag kann man die protokollierten IP-Adressen in den Log-Dateien maskieren, um die Identität der Clients zu verbergen. Trägt man hier `255.255.255.0` ein, wird die letzte Stelle der IP-Adresse auf Null gesetzt.

**ftp_user Squid@** Hiermit kann man das Passwort setzen, welches Squid für den anonymen FTP-Login verwenden soll. Beim Zugriff auf öffentliche FTP-Server wird im allgemeinen als Login `'anonymous'` und als Passwort die eigene Mail-Adresse verwendet, was das Eingeben von Benutzername und Passwort für jeden FTP-Download erspart. Voreingestellt ist `Squid@` ohne Domain, da die Clients aus beliebigen Domains kommen können. Es kann aber sinnvoll sein, hier eine gültige E-Mail-Adresse in der eigenen Domain anzugeben, da einige FTP-Server diese auf Gültigkeit überprüfen.

**cache_mgr webmaster** Eine E-Mail-Adresse, an die Squid eine Nachricht schickt, wenn er unerwartet abstürzt. Voreingestellt ist `webmaster`.

**logfile_rotate 0** Squid ist in der Lage, die gesicherten Log-Dateien zu rotieren, wenn man **`squid -k rotate`** aufruft. Die Dateien werden dabei, entsprechend der angegebenen Anzahl, durchnummeriert, und nach Erreichen des angegebenen Wertes wird die jeweils älteste Datei wieder überschrieben. Dieser Wert steht standardmäßig auf `0`, weil das Archivieren und Löschen der Log-Dateien bei SuSE Linux von einem eigenen Cronjob durchgeführt wird, dessen Konfiguration man in der Datei `/etc/logfiles` findet. Der Zeitraum, nach dem die Dateien gelöscht werden, wird in der `/etc/rc.config` mit dem Eintrag `MAX_DAYS_FOR_LOG_FILES` festgelegt.

**append_domain <domain>** Mit `append_domain` kann man angeben, welche Domain automatisch angehängt wird, wenn keine angegeben wurde. Meist wird man hier die eigene Domain eintragen, dann genügt es, im Browser `www` einzugeben, um auf den eigenen Webserver zu gelangen.

**forwarded_for on** Setzt man diesen Eintrag auf `off`, entfernt Squid die IP-Adresse bzw. den Systemnamen des Clients aus den HTTP-Anfragen.

**negative_ttl 5 minutes; negative_dns_ttl 5 minutes** Normalerweise braucht man diese Werte nicht zu verändern. Hat man aber eine Wählleitung, kann es vorkommen, dass das Internet zeitweilig nicht erreichbar ist. Squid merkt sich dann die erfolglosen Anfragen und weigert sich, diese neu anzufragen, obwohl die Verbindung in das Internet wieder steht. Für diesen Fall sollte man die `minutes` in `seconds` ändern, dann führt auch ein `Reload` im Browser, wenige Sekunden nach der Einwahl, wieder zum Erfolg.

**never_direct allow <acl_name>** Will man verhindern, dass Squid Anfragen direkt aus dem Internet fordert, kann man hiermit die Verwendung eines anderen Proxies erzwingen. Diesen muss man zuvor unter `cache_peer` eingetragen haben. Gibt man als `<acl_name> all` an, erzwingt man, dass sämtliche Anfragen direkt an den `parent` weitergegeben werden. Das kann zum Beispiel nötig sein, wenn man einen Provider verwendet, der die Verwendung seines Proxies zwingend vorschreibt oder die Firewall keinen direkten Zugriff auf das Internet durchlässt.

#### Optionen zur Zugriffskontrolle

Squid bietet ein ausgeklügeltes System, um den Zugriff auf den Proxy zu steuern. Durch die Verwendung so genannter „ACLs“ ist es einfach und vielseitig konfigurierbar. Dabei handelt es sich um Listen mit Regeln, die der Reihe nach abgearbeitet werden. ACLs müssen zuerst definiert werden, bevor sie verwendet werden können. Einige Standard-ACLs wie `all` und `localhost` sind bereits vorhanden. Das Festlegen einer ACL an sich bewirkt aber noch gar nichts. Erst wenn sie tatsächlich eingesetzt wird, z. B. in Verbindung mit `http_access`, werden die definierten Regeln abgearbeitet.

**acl <acl_name> <type> <data>** Eine ACL benötigt zur Definition mindestens drei Angaben. Der Name `<acl_name>` kann frei gewählt werden. Für `<type>` kann man aus einer Vielzahl unterschiedlicher Möglichkeiten auswählen, die man im Abschnitt `ACCESS CONTROLS` in der `/etc/squid.conf`

nachlesen kann. Was für `<data>` anzugeben ist, hängt vom jeweiligen Typ der ACL ab und kann auch aus einer Datei, z. B. mit Rechnernamen, IP-Adressen oder URLs eingelesen werden. Im folgenden einige einfache Beispiele:

```
acl meinesurfer srcdomain .meine-domain.com
acl lehrer src 192.168.1.0/255.255.255.0
acl studenten src 192.168.7.0-192.168.9.0/255.255.255.0
acl mittags time MTWHF 12:00-15:00
```

**http_access allow <acl_name>** Mit `http_access` wird festgelegt, wer den Proxy verwenden darf und auf was er im Internet zugreifen darf. Dabei sind ACLs anzugeben, `localhost` und `all` sind weiter oben bereits definiert, die mit `deny` oder `allow` den Zugriff sperren oder freigeben. Man kann hier eine Liste mit vielen `http_access`-Einträgen erstellen, die von oben nach unten abgearbeitet werden; je nachdem, was zuerst zutrifft, wird der Zugriff auf die angeforderte URL freigegeben oder gesperrt. Als letzter Eintrag sollte immer `http_access deny all` stehen. Im folgenden Beispiel hat `localhost`, also der lokale Rechner, freien Zugriff auf alles, während er für alle anderen komplett gesperrt ist:

```
http_access allow localhost
http_access deny all
```

Noch ein Beispiel, in dem die zuvor definierten ACLs verwendet werden: Die Gruppe `'lehrer'` hat jederzeit Zugriff auf das Internet, während die Gruppe `'studenten'` nur Montags bis Freitags, und da nur mittags, surfen darf:

```
http_access deny localhost
http_access allow lehrer
http_access allow studenten mittags
http_access deny all
```

Die Liste mit den eigenen `http_access`-Einträgen sollte man der Übersichtlichkeit halber nur an der dafür vorgesehenen Stelle in der `/etc/squid.conf` eintragen. Das bedeutet zwischen dem Text

```
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
```

und dem abschließenden

```
http_access deny all
```

**redirect_program /usr/bin/squidGuard** Mit dieser Option kann man einen „redirector“, wie z. B. SquidGuard angeben, der in der Lage ist, unerwünschte URLs zu sperren. In Verbindung mit Proxy-Authentifizierung und den passenden ACLs kann man so den Zugriff auf das Internet für verschiedene Benutzergruppen sehr differenziert steuern. SquidGuard ist ein eigenes Paket, das separat zu installieren und konfigurieren ist.

**authenticate_program /usr/sbin/pam_auth** Sollen sich die Benutzer am Proxy authentifizieren müssen, kann man hier ein entsprechendes Programm wie z. B. pam_auth angeben. Bei der Verwendung von pam_auth öffnet sich für den Anwender beim ersten Zugriff ein Loginfenster, in dem er Benutzername und Passwort eingeben muss. Zusätzlich ist noch eine ACL erforderlich, damit nur Clients mit gültigem Login surfen können:

```
acl password proxy_auth REQUIRED

http_access allow password
http_access deny all
```

Das `REQUIRED` nach `proxy_auth` kann man auch durch eine Liste von erlaubten Benutzernamen oder einen Pfad zu solch einer Liste ersetzen.

**ident_lookup_access allow <acl_name>** Hiermit erreicht man, dass auf alle durch die ACL definierten Clients eine Ident-Anfrage ausgeführt wird, um die Identität des jeweiligen Benutzers zu ermitteln. Setzt man für `<acl_name>` `all` ein, erfolgt dies generell für alle Clients. Auf den Clients muss dazu ein Ident-Daemon laufen, bei Linux kann man dafür das Paket pidentd installieren, für **Windows** gibt es freie Software, die man sich aus dem Internet besorgen kann. Damit nur Clients mit erfolgreichem Ident-Lookup zugelassen werden, ist auch hier wieder eine entsprechende ACL zu definieren:

```
acl identhosts ident REQUIRED

http_access allow identhosts
http_access deny all
```

Auch hier kann man das `REQUIRED` wieder durch eine Liste erlaubter Benutzernamen ersetzen. Die Verwendung von `Ident` kann den Zugriff merklich verlangsamen, da die Ident-Lookups durchaus für jede Anfrage wiederholt werden.

## 7.6 Transparente Proxy-Konfiguration

Normalerweise schickt der Web-Browser an einen bestimmten Port des Proxy-Servers Anfragen und der Proxy stellt die angeforderten Objekte zur Verfügung, ob sie nun im Cache sind oder nicht. Innerhalb eines echten Netzwerks können verschiedene Situationen auftreten:

- Aus Sicherheitsgründen ist es besser, wenn alle Clients zum Surfen im Internet einen Proxy verwenden.
- Es ist notwendig, dass alle Clients einen Proxy verwenden, egal ob sie sich dessen bewusst sind oder nicht.
- In großen Netzwerken, die bereits einen Proxy verwenden, ist es möglich, veränderte Konfigurationen der einzelnen Rechner zu speichern, falls sich Änderungen am System ergeben.

In jedem dieser Fälle kann ein transparenter Proxy eingesetzt werden. Das Prinzip ist denkbar einfach: Der Proxy nimmt die Anfragen des Web-Browsers entgegen und bearbeitet sie, sodass der Web-Browser die angeforderten Seiten erhält ohne zu wissen, woher sie kommen. Der gesamte Prozess wird transparent ausgeführt, daher der Name für den Vorgang.

### 7.6.1 Kernel-Konfiguration

Zuerst sollte sicherstellt sein, dass der Kernel des Proxy-Servers einen transparenten Proxy unterstützt. Andernfalls muss man dem Kernel diese Optionen hinzufügen und ihn neu kompilieren. Weitere Informationen dazu entnehmen Sie bitte dem SuSE Linux Referenzhandbuch.

Wählen Sie im entsprechenden Eintrag zu den Netzwerkoptionen '`Network Firewalls`', und dann die Optionen '`IP: firewalling`' und '`IP: Transparent proxying`'. Jetzt muss nur noch die neue Konfiguration gespeichert, der neue Kernel kompiliert und installiert, ggf. LILO neu konfiguriert und das System neu gestartet werden.

### 7.6.2 Konfigurationsoptionen in /etc/squid.conf

Folgende Optionen in der Datei `/etc/squid.conf` müssen aktiviert werden, um einen transparenten Proxy aufzusetzen:

- **`httpd_accel_host`** virtual
- **`httpd_accel_port`** 80 # Port, auf dem sich der tatsächliche HTTP-Server befindet.
- **`httpd_accel_with_proxy`** on
- **`httpd_accel_uses_host_header`** on

Die Standardkonfiguration der Datei `/etc/squid.conf` erlaubt nur Zugriff auf den Proxy von **`localhost`** aus, deshalb müssen Sie gegebenenfalls weitere Zugriffsregeln definieren;vgl. Abschnitt 7.5 auf Seite 145.

### 7.6.3 Firewall-Konfiguration mit SuSEfirewall

Alle durch die Firewall eingehenden Anfragen müssen mithilfe einer Port-Weiterleitungsregel an den Squid-Port weitergeleitet werden.

Dafür eignet sich ein SuSE-eigenes Tool: SuSEfirewall. Dessen Konfigurationsdatei findet man in der Datei `/etc/rc.config.d/firewall.rc.config`. Die Konfigurationsdatei wiederum setzt sich aus gut dokumentierten Einträgen zusammen. Auch wenn wir nur einen transparenten Proxy einrichten wollen, müssen wir einige Firewall-Optionen konfigurieren. Beispielsweise:

- Gerät zeigt auf Internet: **`FW_DEV_WORLD`**=„eth1“
- Gerät zeigt auf Netzwerk: **`FW_DEV_INT`**=„eth0“

Auf Ports und Dienste (siehe `/etc/exports`) in der Firewall wird von nicht vertrauenswürdigen Netzwerken also dem Internet zugegriffen. In diesem Beispiel bieten wir lediglich Web-Dienste nach außen hin an:

**`FW_SERVICES_EXTERNAL_TCP`**="www"

Auf Ports/Dienste (siehe `/etc/exports`) in der Firewall wird vom sicheren Netzwerk, sowohl TCP und UDP, zugegriffen.

**`FW_SERVICES_INTERNAL_TCP`**="domain www 3128"

**`FW_SERVICES_INTERNAL_UDP`**="domain"

Wir greifen auf Web-Dienste und Squid (dessen Standardport ist 3128) zu. Der oben beschriebene Dienst „Domain“ steht für DNS oder Domain Name Server. Es ist üblich, diesen Dienst zu nutzen. Andernfalls entfernen wir ihn einfach aus obigem Eintrag und setzen folgende Option auf **no**:

**`FW_SERVICE_DNS`**="yes"

Die wichtigste Option ist die Ziffer 15:

```
#
# 15.)
# Welcher Zugriff auf die einzelnen Dienste soll an einen lokalen
# Port auf dem Firewall-Rechner umgeleitet werden?
#
# Damit können alle internen Benutzer gezwungen werden, über den
# Squid-Proxy zu surfen oder es kann eingehender Webverkehr
# transparent an einen sicheren Web-Server umgeleitet werden.
#
# Wahl: keinen Eintrag vornehmen oder folgend erklärte Syntax von
# Umleitungsregeln, getrennt durch Leerzeichen, verwenden.
# Eine Umleitungsregel besteht aus 1) Quelle IP/Netz, 2) Ziel
# IP/Netz, 3) ursprünglicher Zielport und 4) lokaler Port, an den
# der Verkehr umgeleitet werden soll, getrennt durch Kommata, z.~B.:
# "10.0.0.0/8,0/0,80,3128 0/0,172.20.1.1,80,8080"
#
```

Im obigen Kommentar wird die einzuhaltende Syntax gezeigt. Zuerst greifen die IP-Adresse und die Netzwerkmaske der „internen Netzwerke“ auf die Proxy-Firewall zu. Dann die IP-Adresse und die Netzwerkmaske, an die Anfragen von den Clients „gesendet“ werden. Im Fall von Web-Browsern wählt man die Netzwerke **`0/0`**. Dies ist eine Wildcard und bedeutet „überallhin“. Danach kommt der „ursprüngliche“ Port, an den diese Anfragen geschickt wurden und schließlich folgt der Port, an den die Anfragen „umgeleitet“ wurden.

Im konkreten Fall werden Web-Dienste (Port `80`) auf den Proxy-Port (hier `3128` umgeleitet. Falls mehrere Netzwerke oder Dienste hinzugefügt werden sollen, müssen diese durch ein Leerzeichen im entsprechenden Eintrag getrennt werden.

**`FW_REDIRECT_TCP`**="192.168.0.0/16,0/0,80,3128"

**`FW_REDIRECT_UDP`**="192.168.0.0/16,0/0,80,3128"

Zum Starten der Firewall und der neuen Konfiguration muss man einen Eintrag in der Datei `/etc/rc.config` editieren. Der Eintrag **`START_FW`** muss auf "yes"gesetzt werden:

**`START_FW`**="yes"

Hinweis

Nachdem die Konfigurationsdatei `/etc/rc.config` manuell geändert wurde, muss das SuSEconfig-Skript manuell ausgeführt werden, damit die Änderungen in Ihrem System wirksam werden. Am besten verwendet man dafür YaST ('`System administration`' -> '`Change configuration file`'). SuSEconfig wird dann automatisch ausgeführt.

Starten Sie Squid wie in Abschnitt 7.4 auf Seite 141 beschrieben. Anhand der Protokolldateien in `/var/squid/logs/access.log` kann überprüft werden, ob alles richtig funktioniert. Um zu überprüfen, ob alle Ports korrekt konfiguriert wurden, kann von jedem beliebigen Rechner außerhalb unserer Netzwerke auf dem Rechner ein Portscan ausgeführt werden. Nur der Web-Dienst-Port (80) solte offen sein. Der Portscan führt über **`nmap`**:

```
nmap -O IP_address
```

## 7.7 Squid und andere Programme

In diesem Abschnitt wird gezeigt, wie andere Applikationen mit Squid interagieren.

cachemgr.cgi ermöglicht dem Systemadministrator, den benötigten Speicher für das Zwischenspeichern von Objekten zu überprüfen. Squidgrd filtert Webseiten, und calamaris ist ein Berichtsgenerator für Squid.

### 7.7.1 cachemgr.cgi

Der Cache-Manager (cachemgr.cgi) ist ein CGI-Hilfsprogramm zur Ausgabe von Statistiken über den benötigten Speicher des laufenden Squid-Prozesses. Im Gegensatz zum Protokollieren erleichtert dies die Cache-Verwaltung und die Anzeige von Statistiken.

#### Einrichten

Zuerst wird ein lauffähiger Web-Server auf dem System benötigt. Als Benutzer `'root'` gibt man folgendes eingeben, um herauszufinden, ob Apache bereits läuft: **`rcapache status`**.

Erscheint eine Nachricht wie diese:

```
Checking for service httpd: OK
Server uptime: 1 day 18 hours 29 minutes 39 seconds
```

dann läuft Apache auf unserem Rechner. Andernfalls müssen Sie folgendes eingeben: **`rcapache start`**

So wird Apache mit den SuSE Linux-Standardeinstellungen gestartet. Weitere Details zu Apache finden sich in diesem Handbuch.

Als letzten Schritt muss man die Datei `cachemgr.cgi` in das Verzeichnis `cgi-bin` von Apache kopieren:

```
cp /usr/share/doc/packages/squid/scripts/cachemgr.cgi
   /usr/local/httpd/cgi-bin
```

#### Cache-Manager ACLs in /etc/squid.conf

Folgende Standardeinstellungen sind für den Cache-Manager erforderlich:

```
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
```

Folgende Regeln sollten enthalten sein:

```
http_access allow manager localhost
http_access deny manager
```

Die erste ACL ist am wichtigsten, da der Cache-Manager versucht, mit dem Squid über das cach_object-Protokoll zu kommunizieren. Die folgenden Regeln setzen voraus, dass der Web-Server und Squid auf demselben Rechner laufen. Die Kommunikation zwischen dem Cache-Manager und Squid entsteht beim Web-Server, nicht beim Browser. Befindet sich der Web-Server also auf einem anderen Rechner, müssen Sie extra eine ACL wie in der Beispieldatei 7.7.1 hinzufügen.

```
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl webserver src 192.168.1.7/255.255.255.255 # IP des Webservers
```

Datei 7.7.1: Zugriffsregeln

Dann werden noch folgende Regeln aus Datei 7.7.2 auf der nächsten Seite benötigt.

Es ist auch möglich, ein Passwort für den Manager zu konfigurieren, wenn auf mehrere Optionen zugegriffen werden soll, wie z. B. Schließen des Cache von Remote oder Anzeigen weiterer Informationen über den Cache. Dann müssen Sie den Eintrag **`cachemgr_passwd`** und die Optionenliste, die angezeigt werden soll, mit einem Passwort für den Manager konfigurieren. Diese Liste erscheint als Teil der Eintragkommentare in `/etc/squid.conf`.

Immer wenn sich die Konfigurationsdatei geändert hat, muss Squid mit dem Kommando **`rcsquid reload`** neu gestartet werden.

#### Statistiken anzeigen

Gehen Sie zur entsprechenden Web-Seite, z. B.:
`http://webserver.example.org/cgi-bin/cachemgr.cgi`

```
http_access allow manager localhost
http_access allow manager webserver
http_access deny manager
```

Datei 7.7.2: Zugriffsregeln

Drücken Sie auf ‘continue’ und lassen Sie sich die verschiedenen Statistiken anzeigen. Weitere Informationen über die einzelnen Einträge, die vom Cache-Manager ausgegeben werden, finden sich in den FAQs zu Squid: http://www.squid-cache.org/Doc/FAQ/FAQ-9.html

### 7.7.2 SquidGuard

Dieses Kapitel soll lediglich eine Einführung zur Konfiguration von SquidGuard sowie ein paar Ratschläge zu dessen Einsatz geben. Auf eine umfangreiche Erklärung wird an dieser Stelle verzichtet. Tiefer gehende Informationen finden sich auf den Webseiten zu SquidGuard: http://www.squidguard.org

SquidGuard ist ein freier (GPL), flexibler und ultraschneller Filter, ein Umleiter und „Zugriffs-Controller-PlugIn“ für Squid. Er ermöglicht das Festlegen einer Vielzahl von Zugriffsregeln mit unterschiedlichen Beschränkungen für verschiedene Benutzergruppen für einen Squid-Cache. SquidGuard verwendet die Standardschnittstelle von Squid zum Umleiten.

squidGuard kann u. a. für Folgendes verwendet werden:

- Beschränkung des Internetzugriffs für einige Benutzer auf bestimmte akzeptierte/bekannte Web-Server und/oder URLs.
- Zugriffsverweigerung für einige Benutzer auf bestimmte Web-Server und/oder URLs.
- Zugriffsverweigerung für einige Benutzer auf URLs, die bestimmte reguläre Ausdrücke oder Wörter enthalten.
- Umleiten gesperrter URLs an eine „intelligente“ CGI-basierte Infoseite
- Umleiten nicht registrierter Benutzer an ein Registrationsformular.
- Umleiten von Bannern an ein leeres GIF.
- Unterschiedliche Zugriffsregeln abhängig von der Uhrzeit, dem Wochentag, dem Datum etc.
- Unterschiedliche Regeln für die einzelnen Benutzergruppen

Weder mit squidGuard noch mit Squid ist folgendes möglich:

- Text innerhalb von Dokumenten filtern/zensieren/editieren
- In HTML eingebettete Skriptsprachen wie JavaScript oder VBscript filtern/zensieren/editieren

#### Verwendung von SquidGuard

Installieren Sie das Paket `squidgrd` aus der Serie n. Editieren Sie die Konfigurationsdatei `/etc/squidguard.conf`. Es gibt zahlreiche andere Konfigurationsbeispiele unter `http://www.squidguard.org/config/`. Sie können später mit komplizierteren Konfigurationsenstellungen experimentieren.

Der nächste Schritt besteht darin, eine Dummy-Seite „Zugriff verweigert“ oder eine mehr oder weniger intelligente CGI-Seite zu erzeugen, um Squid umzuleiten, falls der Client eine verbotene Webseite anfordert. Der Einsatz von Apache wird auch hier wieder empfohlen.

Nun müssen wir Squid sagen, dass er SquidGuard benutzen soll. Dafür verwenden wir folgende Einträge in der Datei `/etc/squid.conf`:

```
redirect_program /usr/bin/squidGuard
```

Eine andere Option namens **`redirect_children`** konfiguriert die Anzahl der verschiedenen auf dem Rechner laufenden „redirect“, also Umleitungsprozesse (in diesem Fall SquidGuard). SquidGuard ist schnell genug, um eine Vielzahl von Anfragen (SquidGuard ist wirklich schnell: 100.000 Anfragen innerhalb von 10 Sekunden auf einem 500MHz Pentium mit 5900 Domains, 7880 URLs,gesamt 13780) zu bearbeiten. Es wird daher nicht empfohlen, mehr als 5 Prozesse festzusetzen, da die Zuweisung dieser Prozesse unnötig viel Speicher braucht.

```
redirect_children 5
```

Als Letztes senden Sie ein HUP-Signal zum Squid, damit die neue Konfiguration eingelesen wird:

```
rcsquid reload
```

Nun können Sie Ihre Einstellungen in einem Browser testen.

### 7.7.3 Erzeugen von Cache-Berichten mit Calamaris

Calamaris ist ein Perl-Skript, das zur Erzeugung von Aktivitätsberichten des Cache im ASCII- oder HTML-Format verwendet wird. Es arbeitet mit Squid-eigenen Zugriffsprotokolldateien. Die Homepage zu Calamaris befindet sich unter: `http://Calamaris.Cord.de/`

Das Programm ist einfach zu verwenden. Melden Sie sich als `'root'` an und geben Sie folgendes ein:

```
cat access.log.files | calamaris [options] > reportfile
```

Beim Verketten mehrerer Protokolldateien ist die Beachtung der chronologischen Reihenfolge wichtig, d.h. ältere Dateien kommen zuerst.

Die verschiedenen Optionen:

**-a** wird normalerweise zur Ausgabe aller verfügbaren Berichte verwendet, mit

**-w** erhält man einen HTML-Bericht und mit

**-l** eine Nachricht oder ein Logo im Header des Berichts.

Weitere Informationen über die verschiedenen Optionen finden Sie in der Manual Page zu calamaris: **man calamaris**

Ein übliches Beispiel:

```
cat access.log.2 access.log.1 access.log | calamaris -a -w >
```

**/usr/local/httpd/htdocs/Squid/squidreport.html**

Der Bericht wird im Verzeichnis des Web-Servers abgelegt. Wieder wird Apache benötigt, um die Berichte anzeigen zu können.

Ein weiteres, leistungsstarkes Tool zum Erzeugen von Cache-Berichten ist SARG (Squid Analysis Report Generator), das Sie in der Serie n finden. Weitere Informationen dazu gibt es auf der entsprechenden Internetseite unter: http://web.onda.com.br/orso/

## 7.8 Weitere Informationen zu Squid

Besuchen Sie die Homepage von Squid: http://www.squid-cache.org/. Hier finden Sie den Squid User Guide und eine sehr umfangreiche Sammlung von FAQs zu Squid.

Das Mini-Howto zu einem transparenten Proxy im Paket howtoen, unter: /usr/share/doc/howto/en/mini/TransparentProxy.gz

Des Weiteren gibt es Mailinglisten für Squid unter: squid-users@squid-cache.org.

Das Archiv dazu befindet sich unter: http://www.squid-cache.org/mail-archive/squid-users/

# 8 Sicherheit im Netzwerk

## 8.1 Masquerading und Firewall

Wegen seiner herausragenden Netzwerkfähigkeiten wird Linux immer häufiger als Router für Wählleitungen oder auch Standleitungen verwendet. Der Begriff „Router“ bezieht sich hierbei auf einen Rechner, der mehr als ein Netzwerkinterface hat und der Pakete, die nicht für eines seiner eigenen Netzwerkinterfaces bestimmt sind, an seine jeweiligen Kommunikationspartner weiterleitet. Ein Router wird häufig auch „gateway“ genannt. Die im Linux-Kernel vorhandenen Paketfilter ermöglichen eine präzise Steuerung dafür, welche Pakete des Datenverkehrs nun passieren dürfen und welche nicht.

Das Festlegen der genauen Filterregeln für diesen Paketfilter erfordert in der Regel etwas Erfahrung seitens des Administrators. SuSE Linux enthält für den weniger erfahrenen Benutzer zwei voneinander unabhängige Pakete, die das Einstellen dieser Regeln erleichtern sollen: Das traditionsreichere Paket `SuSEfirewall` und das neuere Paket `personal-firewall`. Die beiden Pakete unterscheiden sich grundsätzlich in der Konfigurierbarkeit, und in der Folge natürlich auch in der Flexibilität und im Zweck.

Die SuSEfirewall ist sehr flexibel konfigurierbar und eignet sich deswegen auch zum Aufbau von komplexeren Paketfilterkonstrukten. Die personal-firewall lässt sich nur mit einer einzigen Variablen konfigurieren und hat den Zweck, sämtliche Versuche zu unterbinden, Verbindungen zu dem Rechner aufzubauen. Beide Paketfilter-Pakete erlauben es, einen Linux-Rechner mittels Masquerading als Router zur Anbindung eines internen Netzwerks mit nur einer einzigen von außen sichtbaren IP-Adresse zu betreiben. Masquerading wird also mit Hilfe von Regeln eines Paketfilters realisiert.

**Achtung**

Die hier vorgestellten Verfahren gelten als Standard und funktionieren in der Regel. Es gibt jedoch keine Garantie dafür, dass sich nicht doch in diesem Buch oder woanders ein Fehler eingeschlichen hat. Sollten Cracker trotz umfassender korrekter Schutzmaßnahmen Ihrerseits in Ihr System eindringen, dann machen Sie bitte nicht die Buchautoren verantwortlich. Auch wenn Sie nicht direkt eine Antwort erhalten, können Sie sicher sein, dass wir für Kritik und Anregungen dankbar sind und Verbesserungen einbringen werden.

### 8.1.1 Grundlagen des Masquerading

Masquerading ist der Linux-Speziallfall von NAT (engl. *Network Address Translation*) der Übersetzung von Netzwerkadressen. Das Prinzip dahinter ist nicht sonderlich kompliziert: Ihr Router hat mehr als ein Netzwerkinterface, typischerweise sind das eine Netzkarte und ein Modem (oder ein ISDN-Interface). Eines dieser Interfaces wird Sie nach außen anbinden, eines oder mehrere andere verbinden Ihren Rechner mit den weiteren Rechnern in Ihrem Netz. In einem Beispiel soll nun per ISDN nach außen eingewählt werden, das äußere Netzwerkinterface ist `ippp0`. Sie haben mehrere Rechner im lokalen Netz mit der Netzwerkkarte Ihres Linux-Routers verbunden, die in diesem Beispiel `eth0` heißt. Als Netzwerkadresse für Ihr inneres Netz verwenden Sie das Netz `192.168.0.0` und der Router hört auf die Adresse `192.168.0.1`. Die anzubindenden Rechner haben die Adressen `192.168.0.2`, `192.168.0.3` usw. Diese Rechner senden alle Pakete, die nicht für das eigene Netz bestimmt sind, an die Adresse `192.168.0.1`, also das Netzwerkinterface von Ihrem Router, der damit als Default-Router oder auch default-gateway verwendet wird.

**Hinweis**

Achten Sie beim Konfigurieren Ihres Netzwerks immer auf übereinstimmende broadcast-Adressen und Netzwerkmasken!

Wird nun einer der Rechner in Ihrem Netz ein Paket fürs Internet abschicken, dann landet es beim Default-Router. Dieser muss so konfiguriert sein, dass er solche Pakete auch weiterleitet. Aus Sicherheitsgründen wird eine SuSE-Linux Installation dies nicht tun! Ändern Sie die Variable `IP_FORWARD` in der Datei `/etc/rc.config` auf `IP_FORWARD=yes`. Nach einem reboot oder dem folgenden Kommando ist das Weiterleiten aktiviert:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Jetzt beginnt Masquerading. Da der Router von außen gesehen nur eine einzige IP-Adresse haben soll (in unserem Beispiel die seines ISDN-Interfaces nach erfolgter Einwahl), muss er nun die Absenderadresse des ausgehenden Pakets gegen die eigene austauschen und dann über sein „äußeres“ Netzwerkinterface nach außen weiterleiten. Würde er die Absenderadresse nicht austauschen, dann würde der Empfänger nichts zurückzuschicken können. Dies ist vor allem dann unmöglich, wenn Sie den IP-Adressbereich `192.168.x.x` verwendet haben, denn es handelt sich dabei zwar um gültige IP-Adressen, doch werden diese von den Routern im Internet nicht weitergeleitet.

Der Zielrechner der Verbindung kennt nur Ihrem Router, nicht aber den eigentlichen Absender-Rechner in Ihrem inneren Netzwerk, der hinter Ihrem Router versteckts. Daher kommt der Begriff „Masquerading“. Die Zieladresse für Anwortpakete ist wegen der Adressübersetzung wieder unser Router. Dieser muss die Pakete erkennen und die Zieladresse so umschreiben, dass sie zum richtigen Rechner im lokalen Netz gelangen.

Diese Erkennung von Paketen, die zu Verbindungen gehören, die durch Masquerading durch den Router entstanden sind, geschieht mit Hilfe einer Tabelle,

die direkt im Kernel Ihres Routers gehalten wird, solange die dazugehörigen Verbindungen aktiv sind. Diese Tabelle kann der Superuser (`'root'`) sogar mit den Kommandos **`ipchains`** oder **`iptables`** einsehen. Bitte konsultieren Sie die Manpages dieser Kommandos für genauere Anleitungen. Für die Identifizierung einzelner Masquerade Verbindungen sind neben Absender- und Zieladresse auch Port-Nummern und die beteiligten Protokolle an sich relevant. Damit ist es möglich, dass Ihr Router für jeden einzelnen Ihrer lokalen Rechner viele Tausend Verbindungen gleichzeitig „verstecken“ kann.

Da der Weg der Pakete von außen nach innen von der Masquerading-Tabelle abhängt, gibt es keine Möglichkeit, von außen eine Verbindung nach innen zu öffnen. Für diese Verbindung gäbe es keinen Eintrag in der Tabelle, weil dieser erst dann zustande kommt, wenn von innen eine Verbindung nach außen geöffnet wird. Eine etablierte Verbindung hat darüber hinaus in der Tabelle einen zugeordneten Status, so dass dieser Tabelleneintrag nicht von einer zweiten Verbindung genutzt werden kann. Diese zweite Verbindung müsste ja einen anderen Status durchlaufen.

In der Folge ergeben sich nun Probleme mit manchen Anwendungen: Die Protokolle, über die sich die Programme unterhalten, öffnen manchmal weitere Verbindungen oder schicken Pakete vom Server zu Ihrem Client, die nicht ohne weiteres von einem einfachen Paketfilter als legitim erkannt werden können.

Beispiele für diese Protokolle sind ICQ, cucme, IRC (DCC, CTCP), Quake und FTP (im PORT-Mode). Netscape, das Standard-FTP-Programm und viele andere benutzen den PASV-Modus, der im Zusammenhang mit Paketfiltern und Masquerading weit weniger problembehaftet ist. Das FTP-Protokoll öffnet für jede zu übertragende Datei neben der Datenverbindung zusätzlich eine Kontrollverbindung. Beim PORT-Modus öffnet der Server eine Verbindung zum Client, beim PASV (passive) Modus öffnet der Client die Verbindung. Wie wir wissen, können Verbindungen nur von innen heraus geöffnet werden, was die Schwierigkeiten des PORT-Modus bei FTP erklärt.

### 8.1.2 Grundlagen Firewalling

„Firewall“ ist wohl der am weitesten verbreitete Begriff für einen Mechanismus, der zwei Netze miteinander verbindet, aber für möglichst kontrollierten Datenverkehr sorgt. Es gibt verschiedene Bauarten von Firewalls, die sich hauptsächlich in der logisch-abstrakten Ebene unterscheiden, auf der sie den Datenverkehr untersuchen und regulieren. Die Methode, die wir hier vorstellen, müsste sich eigentlich genauer „Paketfilter“ nennen. Wie andere Bauarten auch kann ein Paketfilter allein nicht als vollständiger, sicherer Schutz gelten. Ein Paketfilter regelt den Durchlass anhand von Kriterien wie Protokoll, Port und IP-Adresse. Auf diese Weise können Sie also Pakete abfangen, die aufgrund ihrer Adressierung nicht in Ihr Netz durchdringen sollen. Beispielsweise sollten Sie Pakete abfangen, die den telnet-Dienst Ihrer Rechner auf port 23 zum Ziel haben. Wenn Sie beispielsweise Zugriffe auf Ihren Webserver zulassen wollen, müssen Sie den dazugehörigen Port freischalten. Der Inhalt dieser Pakete, falls sie legitimnn adressiert sind (also beispielsweise mit Ihrem Webserver als Ziel), wird nicht

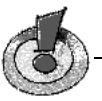
untersucht. Das Paket könnte insofern einen Angriff auf ein CGI-Programm auf Ihrem Webserver enthalten und wird vom Paketfilter durchgelassen.

Ein wirksameres — wenn auch komplexeres — Konstrukt ist die Kombination von mehreren Bauarten, beispielsweise ein Paketfilter mit zusätzlichem Application Gateway/Proxy. Der Paketfilter wehrt Pakete ab, die zum Beispiel an nicht freigeschaltete Ports gerichtet sind. Nur Pakete für ein Application Gateway sollen durchgelassen werden. Dieses Proxy tut nun so, als wäre es der eigentliche Kommunikationspartner des Servers, der mit uns eine Verbindung herstellt. In diesem Sinne kann ein solches Proxy als eine Masquerading-Maschine auf der Ebene des Protokolls der jeweiligen Anwendung angesehen werden. Ein Beispiel für solch ein Proxy ist Squid, ein HTTP Proxy Server, für den Sie Ihren Browser so konfigurieren müssen, dass Anfragen für HTML-Seiten zuerst an den Speicher des Proxy gehen und nur, wenn dort die Seite nicht zu finden ist, in das Internet geschickt werden. Die SuSE proxy suite (das Paket proxy-suite aus der Serie sec) enthält übrigens einen Proxy-Server für das FTP-Protokoll.

Im Folgenden wollen wir uns auf die zwei Paketfilter-Pakete bei SuSE-Linux konzentrieren. Für mehr Informationen und weitere Links zum Thema Firewall lesen Sie bitte das Firewall-HOWTO, enthalten im Paket `howtode`, Serie `doc`. Es lässt sich mit dem Kommando **`less /usr/share/doc/howto/de/DE-Firewall-HOWTO.txt.gz`** lesen, wenn das Paket `howtode` installiert ist.

### 8.1.3 Personal-firewall

Wie bereits erwähnt gibt es bei SuSE-Linux zwei verschiedene Pakete, um Filterregeln aufzusetzen: Personal-firewall und SuSEfirewall. Der Hauptunterschied zwischen den beiden ist die Konfigurierbarkeit: Die Personal-firewall ist dafür gedacht, konfigurationsfrei und wartungsarm Verbindungen ins Internet öffnen zu können, aber ansonsten keine Pakete durchzulassen. Der geschützte Rechner (bzw. das geschützte Netzwerkinterface) soll nicht als Server (der Dienste anbietet) im Internet erscheinen. Die Personal-firewall wird daher allgemeinen Anforderungen durchaus gerecht.



Hinweis

Wie bereits beim Masquerading erwähnt ergeben sich durch die Ablehnung von Anfragen zu einem Verbindungsaufbau von außen eine Reihe von Problemen mit Protokollen, die eine zweite Verbindung zum Client hin öffnen. Die Personal-firewall ist für derartige Dienste also ebenso problematisch wie die Masquerading Funktion.

Die Personal-firewall ist mit nur einer einzigen Variablen in der Datei `/etc/rc.config.d/security.rc.config` konfigurierbar:

```
REJECT_ALL_INCOMING_CONNECTIONS
```

Entsprechend der Kommentare in dieser Datei müssen Sie hier das oder die Netzwerkinterfaces, durch Leerzeichen getrennt, eintragen, deren ankommender

Datenverkehr beschränkt werden soll. Neben eben diesen Interfacenamen wie `eth0`, `eth1`, `ippp0` oder `tr0` sind folgende Schlüsselwörter erlaubt:

| | |
|---|---|
| `no` | Die Personal-firewall ist nicht aktiv. Gleiches gilt für eine leere Variable REJECT_ALL_INCOMING_CONNECTIONS. |
| `yes` | Die Personal-firewall wirkt auf alle Interfaces außer `lo`. `lo` ist das loopback interface, localhost. Verbindungen an localhost werden also zugelassen. |
| `modem` | Bezeichnet angeschlossene Modems und ist Kurzform für die Interfacenamen, die mit `ppp` beginnen. |
| `masq` | Pakete, die den Rechner erreichen, aber nicht für eines der Interfaces des Rechners bestimmt sind, sollten bei der Weiterleitung entsprechend maskiert werden (Masquerade). |

Tabelle 8.1: Mögliche Schlüsselwörter

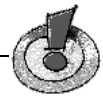
**Achtung**

Masquerading macht nur Sinn, wenn Sie auch IP-Forwarding (auch routing genannt) eingeschalten, indem Sie die Variable **`IP_FORWARD`** in `/etc/rc.config` auf **`IP_FORWARD=yes`** setzen. Beim nächsten Reboot oder nach dem Kommando **`echo 1 > /proc/sys/net/ipv4/ip_forward`** ist IP-Forwarding eingeschaltet.

**Beispiel: ISDN und Modem akzeptieren keine Verbindungen**

```
REJECT_ALL_INCOMING_CONNECTIONS="ippp0 modem"
```

dann werden sämtliche Verbindungsanfragen, die auf Ihrem ISDN-Interface oder Ihrem Modem auftreffen, verworfen. Eine eventuell vorhandene und konfigurierte Netzwerkkarte, die im Linux-System `eth0` benannt wäre, bleibt davon unberührt. Ebenfalls nicht aktiviert ist die Masquerade Funktion.



**Hinweis**

ADSL (z. B. T-DSL) und andere DSL-Varianten zählen hier als Modem, weil ein Interface-Name wie `ppp0` oder `ppp1` verwendet wird.

**Beispiel: Keine Verbindungen zu einem Modem, Masquerading**

```
REJECT_ALL_INCOMING_CONNECTIONS="modem masq"
```

Alle Interfaces, deren Namen mit `ppp` beginnen, können keine Verbindungen mehr annehmen. Sofern IP-Forwarding eingeschaltet ist, werden Pakete von anderen Netzwerkinterfaces als denen, deren Namen mit `ppp` beginnen, hinter der IP-Adresse Ihres Interface nach draußen „versteckt“. Welche Adresse zum Masquerade verwendet wird, ist hier nicht festgelegt! Sie könnten also theoretisch sowohl eine ISDN-Karte als auch ein Modem online haben; die Verbindungen von außen an das Modem würden abgelehnt, die an die ISDN-Karte nicht. Für Masquerading wird als Absenderadresse die Adresse des Interfaces eingetragen, über welches das Paket Ihren Router verlässt.

#### Beispiel: Keine ISDN-Verbindungen, Masquerading

Sie haben eine konfigurierte Netzkarte als `eth0` und eine konfigurierte ISDN-Karte als `ippp0` in Ihrem System installiert und tragen ein:

```
REJECT_ALL_INCOMING_CONNECTIONS="ippp0 masq"
```

Des weiteren haben Sie IP-Forwarding eingeschaltet. Wenn Sie auf Rechnern Ihres lokalen Netzwerks die IP-Adresse Ihrer Netzkarte als Default-Router konfiguriert haben, dann werden Verbindungen aus Ihrem lokalen Netz von Ihrem Router nach draußen maskiert (Masquerade). Verbindungsanfragen von außen an Ihr ISDN-Interface werden ignoriert.

#### Beispiel: Masquerading

```
REJECT_ALL_INCOMING_CONNECTIONS="masq"
```

Es werden keine Verbindungsanfragen an installierte Netzwerkinterfaces ignoriert, aber Verbindungen, die von anderen Rechnern über den Router weitergeleitet werden, werden maskiert (Masquerade). Voraussetzung ist dabei, dass diese Rechner die Adresse eines erreichbaren Netzwerkinterfaces Ihres Routers als default-gateway verwenden. Vorsicht: Damit könnte sich ein beliebiger Rechner, der Ihnen Pakete schicken kann, hinter Ihrem Router verstecken.

### 8.1.4 SuSEfirewall

Die Konfiguration der SuSEfirewall ist kompizierter und erfordert mehr Wissen. Unter `/usr/share/doc/packages/SuSEfirewall` finden Sie Dokumentation zur SuSE firewall, theoretische Überlegungen dazu in Kapitel 8.3.1 auf Seite 168 ff.

Die Konfiguration erfolgt in der Datei `/etc/rc.config.d/firewall.rc.config`, die auch englische Kommentare enthält. Wir werden Ihenen nun Schritt für Schritt eine erfolgreiche Konfiguration vorführen. Es ist bei jedem Punkt angeführt, ob er für Masquerading oder Firewall gilt. In der Konfigurationsdatei ist auch von einer DMZ („Demilitarisierte Zone“) die Rede, auf die an dieser Stelle nicht näher eingegangen wird.

Falls Sie wirklich nicht mehr als Masquerading brauchen, füllen Sie nur die mit *Masquerading* bezeichneten Zeilen aus.

- **START_FW** (Firewall, Masquerading): In `/etc/rc.config` auf `yes` setzen, damit das Skript gestartet wird; so wird Firewall oder Masquerading ermöglicht.
- **FW_DEV_WORLD** (Firewall, Masquerading): Zum Beispiel `eth0`, als Device, das ins Internet führt. Bei ISDN ist es z. B. `ippp0`.
- **FW_DEV_INT** (Firewall, Masquerading): Gegen Sie hier das Device an, das ins innere, „private“ Netz zeigt. Falls kein inneres Netz vorhanden ist, einfach leer lassen.
- **FW_ROUTE** (Firewall, Masquerading): Wenn Sie Masquerading brauchen, müssen Sie hier auf jeden Fall `yes` eintragen. Ihre internen Rechner sind nicht von außen sichtbar, da diese private Netzwerkadressen (z. B. `192.168.x.x`) haben, die im Internet gar nicht geroutet werden.

  Bei einer Firewall ohne Masquerading wählen Sie hier nur dann `yes`, wenn Sie Zugang zum internen Netz erlauben wollen. Dazu müssen die internen Rechner offiziell zugewiesene IP-Adressen haben. Im Normalfall sollten Sie allerdings keinen Zugang von außen auf die internen Rechner *nicht* erlauben!
- **FW_MASQUERADE** (Masquerading): Wenn Sie Masquerading brauchen, müssen Sie hier `yes` eintragen. Beachten Sie, dass es sicherer ist, wenn die Rechner des internen Netzes über Proxy-Server auf das Internet zugreifen.
- **FW_MASQ_NETS** (Masquerading): Tragen Sie hier die Rechner oder Netzwerke ein, für die Masquerading vorgenommen werden soll. Trennen Sie die einzelnen Einträge durch Leerzeichen. Zum Beispiel:

  ```
  FW_MASQ_NETS="192.168.0.0/24 192.168.10.1"
  ```

- **FW_PROTECT_FROM_INTERNAL** (Firewall): Tragen Sie hier `yes` ein, wenn Sie den Firewall-Rechner auch durch Angriffe vom inneren Netz schützen wollen. Dann müssen Sie die Services, die für das innere Netz verfügbar sind, explizit freigeben. Siehe auch **FW_SERVICES_INTERNAL_TCP** und **FW_SERVICES_INTERNAL_UDP**.
- **FW_AUTOPROTECT_GLOBAL_SERVICES** (Firewall): Im Normalfall auf `yes` lassen.
- **FW_SERVICES_EXTERNAL_TCP** (Firewall): Tragen Sie hier die Services ein, auf die zugegriffen werden soll; z. B. `"www smtp ftp domain 443"` – für den Rechner zu Hause, der keine Dienste anbieten soll, tragen Sie meist nichts ein.
- **FW_SERVICES_EXTERNAL_UDP** (Firewall): Wenn Sie nicht gerade einen Nameserver betreiben, auf den von außen zugegriffen werden soll, lassen Sie dieses Feld leer. Ansonsten fügen Sie hier die benötigten Ports ein.
- **FW_SERVICES_INTERNAL_TCP** (Firewall): Hier werden die für das innere Netz zur Verfügung stehenden Dienste deklariert. Die Angaben sind analog zu denen unter **FW_SERVICES_EXTERNAL_TCP**, beziehen sich hier aber auf das *interne* Netz.

- **FW_SERVICES_INTERNAL_UDP** (Firewall): Siehe oben.
- **FW_TRUSTED_NETS** (Firewall): Hier tragen Sie die Rechner ein, denen Sie *wirklich* vertrauen können („Trusted Hosts“). Beachten Sie zudem, dass auch diese Rechner vor Eindringlingen geschützt sein müssen. `"172.20.0.0/16 172.30.4.2"` bedeutet, dass alle Rechner, deren IP-Adresse mit `172.20.x.x` beginnt, sowie der Rechner mit der IP-Adresse `172.30.4.2` durch die Firewall hindurch können.
- **FW_SERVICES_TRUSTED_TCP** (Firewall): Hier legen Sie die TCP-Portadressen fest, die von den „Trusted Hosts“ benutzt werden können. Geben Sie z. B. `1:65535` ein, wenn die vertrauenswürdigen Rechner auf alle Services zugreifen dürfen. Normalerweise sollte es reichen, wenn man hier als Service `ssh` eingibt.
- **FW_SERVICES_TRUSTED_UDP** (Firewall): Wie oben, nur auf UDP bezogen.
- **FW_ALLOW_INCOMING_HIGHPORTS_TCP** (Firewall): Wenn Sie mit normalem (aktivem) FTP arbeiten wollen, so tragen Sie hier `ftp-data` ein.
- **FW_ALLOW_INCOMING_HIGHPORTS_UDP** (Firewall): Tragen Sie hier `dns` ein, damit Sie die in `/etc/resolv.conf` eingetragenen Nameserver verwenden können. Mit `yes` geben Sie alle hohen Portnummern frei.
- **FW_SERVICE_DNS** (Firewall): Falls bei Ihnen ein Nameserver läuft, auf den von außen zugegriffen werden soll, tragen Sie hier `yes` ein; in **FW_TCP_SERVICES_*** muss zugleich der Port 53 freigeschaltet sein.
- **FW_SERVICE_DHCLIENT** (Firewall): Wenn Sie dhclient benützen, um Ihre IP-Adresse zu beziehen, so müssen Sie hier `yes` eintragen.
- **FW_LOG_***: Stellen Sie hier ein, was Sie mitloggen wollen. Für den laufenden Betrieb reicht `yes` bei **FW_LOG_DENY_CRIT**.
- **FW_STOP_KEEP_ROUTING_STATE** (Firewall): Falls Sie automatisch per diald oder über ISDN (dial on demand) ins Internet gehen, so tragen Sie hier `yes` ein.

Damit ist die Konfiguration abgeschlossen. Vergessen Sie nicht, die Firewall zu testen (z. B. **telnet** von außen); Sie sollten dann in `/var/log/messages` in etwa folgende Einträge sehen:

```
Feb  7 01:54:14 www kernel: Packet log: input DENY eth0
PROTO=6 129.27.43.9:1427 195.58.178.210:23 L=60 S=0x00
I=36981 F=0x4000 T=59 SYN (#119)
```

## 8.2 SSH – secure shell, die sichere Alternative

In unserer Zeit der immer stärkeren Vernetzung werden auch Zugriffe auf entfernte Systeme immer häufiger. Ob elektronische Post abgeholt, ein Server gewartet oder einer Webseite eines Redaktionssystems einen Artikel hinzugefügt wird, immer muss eine Authentifikation der Person erfolgen.

In der Regel sollten Nutzer heutzutage verinnerlicht haben, dass ihr Benutzername und ihr Kennwort lediglich für sie allein gedacht sind. Eine entsprechende Vereinbarung zwischen Arbeitgeber, Rechenzentrum oder Serviceanbieter über die Personengebundenheit dieser Daten ist Standard.

Erschreckend ist demgegenüber die weitgehende Praxis, dass Authentifizierung und Datenübertragung weiterhin in Form von Klartextdaten erfolgt. Die ist beispielsweise der Fall, wenn mit **`Post Office Protocol (POP)`** E-Mail abgeholt wird oder man sich mit **`telnet`** auf einem entfernten System anmeldet. Hierbei gehen die in den Nutzungsbedingungen als sensibel eingestuften Nutzerinformationen und Daten, z. B. der Inhalt eines Briefes, oder ein per talk-Kommando geführtes Gespräch, ohne jeden Schutz offen über das Netzwerk. Dies beeinträchtigt einerseits die Privatsphäre des Nutzers und eröffnet andererseits die Möglichkeit zum Missbrauch eines Zugangs. Insbesondere werden solche Zugänge gern benutzt, um von dort aus andere Systeme anzugreifen, oder Administrator- bzw. Rootrechte auf diesem System zu erlangen.

Jedes an der Weiterleitung der Daten beteiligte oder im gleichen lokalen Netz betriebene Gerät wie Firewall, Router, Switch, Mailserver, Arbeitsplatzrechner, etc., kann die Daten zusätzlich einsehen. Grundsätzlich untersagen zwar die geltenden rechtlichen Regelungen ein solches Vorgehen, stellen es sogar unter Strafe, jedoch sind derartige Angriffe oder unberechtigte Einsichtnahmen nur schwer festzustellen und nachzuweisen.

Die SSH-Software liefert hier den gewünschten Schutz. Die komplette Authentifizierung, in der Regel Benutzername und Passwort, und die Kommunikation erfolgen hier verschlüsselt. Zwar ist auch hier weiterhin das Mitschneiden der übertragenen Daten möglich, doch kann der Inhalt mangels fehlendem Schlüssel durch einen Unwissenden nicht wieder entschlüsselt werden. So wird sichere Kommunikation über unsichere Netze wie das Internet möglich. SuSE Linux bietet in der Serie sec das Paket OpenSSH an.

### 8.2.1 Das OpenSSH-Paket

Sobald Sie das Paket OpenSSH installiert haben, stehen Ihnen die Programme ssh, scp und sftp als Alternative für telnet, rlogin, rsh, rcp und ftp zur Verfügung.

### 8.2.2 Das ssh-Programm

Mit dem ssh-Programm können Sie Verbindung zu einem entfernten System aufnehmen und dort interaktiv arbeiten. Es ist somit gleichermaßen ein Ersatz für telnet und rlogin. Aufgrund der Verwandtschaft zu rlogin zeigt der zusätzliche symbolische Name slogin ebenfalls auf ssh. Zum Beispiel kann man sich mittels

```
hannes@erde:~> ssh sonne
```

auf dem Rechner sonne anmelden. Anschließend wird man nach seinem Passwort auf dem System sonne gefragt:

```
hannes@sonne's password:
```

Nach erfolgreicher Authentifizierung kann dort auf der Kommandozeile oder interaktiv, z. B. mit dem SuSE- Administrationsprogramm YaST, gearbeitet werden. Sollten sich der lokale Benutzername und der auf dem entfernten System unterscheiden, kann ein abweichender Name angegeben werden:

```
hannes@erde:~> ssh -l august sonne
```

oder

```
hannes@erde:~> ssh august@sonne
```

Darüber hinaus bietet ssh die von rsh bekannte Möglichkeit, Kommandos auf einem anderen System auszuführen. Im nachfolgenden Beispiel wird das Kommando **`uptime`** auf dem Rechner`sonne` ausgeführt und ein Verzeichnis mit dem Namen `tmp` angelegt. Die Programmausgabe erfolgt auf dem lokalen Terminal des Rechners `erde`.

```
hannes@erde:~> ssh sonne 'uptime; mkdir tmp'
hannes@sonne's password:
  1:21am  up  2:17,  9 users,  load average: 0.15, 0.04, 0.02
```

Hochkommata sind hier zum Zusammenfassen der beiden Anweisungen in einem Kommando erforderlich. Nur so wird auch der zweite Befehl auf dem Rechner `sonne` ausgeführt.

### 8.2.3 scp – sicheres Kopieren

Mittels scp kopieren Sie Dateien auf einen entfernten Rechner. scp ist der sichere, verschlüsselte Ersatz für rcp. Zum Beispiel kopiert

```
hannes@erde:~> scp MeinBrief.tex sonne:
```

die Datei `MeinBrief.tex` vom Rechner `erde` auf den Rechner `sonne`. Insoweit sich die beteiligten Nutzernamen auf `erde` und `sonne` unterscheiden, muss bei scp auf die bereits zum ssh-Kommando beschriebene Schreibweise **`Nutzername@Rechnername`** zurückgegriffen werden. Eine Option **`-l`** existiert nicht.

Nachdem das Passwort eingegeben wurde, beginnt scp mit der Datenübertragung und zeigt dabei den Fortschritt anhand eines von links nach rechts anwachsenden Balkens aus Sternen an. Zudem wird am rechten Rand die geschätzte Restübertragungszeit (engl. *estimated time of arrival*) angezeigt. Jegliche Ausgabe kann durch die Option **`-q`** unterdrückt werden.

scp bietet neben dem Kopieren einzelner Dateien ein rekursives Verfahren zum Übertragen ganzer Verzeichnisse.

```
hannes@erde:~> scp -r src/ sonne:backup/
```

kopiert den kompletten Inhalt des Verzeichnisses `src/` inklusive aller Unterverzeichnisse auf den Rechner `sonne` und dort in das Unterverzeichnis `backup/`. Dieses wird automatisch angelegt wenn es fehlt.

Mittels der Option **`-p`** kann scp die Zeitstempel der Dateien erhalten. **`-C`** sorgt für eine komprimierte Übertragung. Dadurch wird einerseits das zu übertragende Datenvolumen minimiert, andererseits aber ein höherer Rechenaufwand erforderlich.

### 8.2.4 sftp - sicherere Dateiübertragung

Alternativ kann man zur sicheren Datenübertragung sftp verwenden. sftp bietet innerhalb der Sitzung viele der von ftp bekannten Kommandos. Gegenüber scp mag es vor allem beim Übertragen von Daten, deren Dateinamen unbekannt sind, von Vorteil sein.

### 8.2.5 Der SSH Daemon (sshd) – die Serverseite

Damit ssh und scp, die Clientprogramme des SSH-Paketes, eingesetzt werden können, muss im Hintergrund der SSH-Daemon, ein Server, laufen. Dieser erwartet seine Verbindungen auf **`TCP/IP Port 22`**.

Der SSH-Daemon ist Bestandteil des SSH-Paketes und wird in einem SuSE Linux System automatisch in `Runlevel 3` und `5` gestartet. Die Variable `START_SSHD` ist dazu in `/etc/rc.config` auf `yes` voreingestellt.

Während des ersten Starts generiert der Daemon zwei Schlüsselpaare. Die Schlüsselpaare bestehen aus einem privaten und einem öffentlichen (engl. *public*) Teil. Deshalb bezeichnet man dies als ein public-key basiertes Verfahren. Um die Sicherheit der Kommunikation mittels SSH zu gewährleisten, darf ausschließlich der Systemadministrator die Dateien der privaten Schlüssel einsehen können. Die Dateirechte werden per Voreinstellung entsprechend restriktiv gesetzt. Die privaten Schlüssel werden lediglich lokal vom SSH-Daemon benötigt und dürfen an niemanden weitergegeben werden. Demgegenüber werden die öffentlichen Schlüsselbestandteile (an der Namensendung `.pub` erkennbar) an Kommunikationspartner weitergegeben und sind entsprechend für alle Benutzer lesbar.

Eine Verbindung wird vom SSH-Client eingeleitet. Der wartende SSH-Daemon und der anfragende SSH-Client tauschen Identifikationsdaten aus, um die Protokoll- und Softwareversion abzugleichen und die Verbindung zu einem falschen Port auszuschließen. Da ein Kindprozess des ursprünglichen SSH-Daemons antwortet, sind gleichzeitig viele SSH-Verbindungen möglich.

Der Server sendet sodann seinen öffentlichen **`host key`** und einen stündlich vom SSH-Daemon neu generierten **`server key`**. Mittels beider verschlüsselt (engl. *encrypt*) der SSH-Client einen von ihm frei gewählten Sitzungsschlüssel (engl. *session key*) und sendet ihn an den SSH-Server. Er teilt dem Server zudem die gewählte Verschlüsselungsmethode (engl. *cipher*) mit.

Die zur Entschlüsselung des Sitzungsschlüssels zwingend erforderlichen privaten host und server keys, können nicht aus den öffentlichen Teilen abgeleitet werden. Somit kann allein der kontaktierte SSH-Daemon mit seinen privaten Schlüsseln den Sitzungsschlüssel entziffern (vgl. `/usr/share/doc/packages/openssh/RFC.nroff`). Diese einleitende Phase der Verbindung kann man mittels der Fehlersuchoption **`-v`** des SSH-Clientprogramms gut nachvollziehen. Sie endet mit der Bestätigung „Received encrypted confirmation.“ des SSH-Daemons. Indem der Client alle öffentlichen host keys nach der ersten Kontaktaufnahme in `~/.ssh/known_hosts` ablegt, können so genannte „man-in-the-middle“-Angriffe unterbunden werden. SSH-Server, die versuchen, Name und IP-Adresse eines anderen vorzutäuschen, werden durch einen deutlichen

Hinweis enttarnt. Sie fallen entweder durch einen gegenüber `~/.ssh/known_hosts` abweichenden host-Schlüssel auf, oder können mangels passendem privaten Gegenstück den vereinbarten Sitzungschlüssel nicht entschlüsseln.

Es empfiehlt sich, die in `/etc/ssh/` abgelegten privaten und öffentlichen Schlüssel extern und gut geschützt zu archivieren. So können Änderungen der Schlüssel festgestellt und nach einer Neuinstallation die alten wieder eingespielt werden. Dies erspart den Benutzern die beunruhigende Warnung. Ist es sichergestellt, dass es sich trotz der Warnung um den korrekten SSH-Server handelt, muss der vorhandene Eintrag zu diesem System aus `~/.ssh/known_hosts` entfernt werden.

### 8.2.6 SSH-Authentifizierungsmechanismen

Jetzt erfolgt die eigentliche Authentifizierung, die in ihrer einfachsten Weise aus der Eingabe eines Passwortes besteht, wie es in den oben aufgezeigten Beispielen erfolgte. Ziel von SSH war die Einführung einer sicheren, aber zugleich einfach zu nutzenden Software. Wie bei den abzulösenden Programmen rsh und rlogin muss deshalb auch SSH eine im Alltag einfach zu nutzende Authentifizierungsmethode bieten. SSH realisiert dies mittels eines weiteren hier vom Nutzer erzeugten Schlüsselpaares. Dazu liefert das SSH-Paket das Hilfsprogramm ssh-keygen mit. Nach der Eingabe von

```
hannes@sonne:~> ssh-keygen
Generating RSA keys:
```

wird das Schlüsselpaar generiert und der Basisdateiname zur Ablage der Schlüssel erfragt:

```
Enter file in which to save the key (/home/hannes/.ssh/identity):
```

Bestätigen Sie die Voreinstellung und beantworten Sie die Frage nach einer Passphrase:

```
Enter passphrase (empty for no passphrase):
```

Auch wenn die Software eine leere Passphrase nahelegt, sollte bei der hier vorgeschlagenen Vorgehensweise ein Text von zehn bis 30 Zeichen Länge gewählt werden. Verwenden Sie möglichst keine kurzen und einfachen Worte oder Sätze. Nach erfolgter Eingabe wird zur Bestätigung eine Wiederholung der Eingabe verlangt. Anschließend wird der Ablageort des privaten und öffentlichen Schlüssels, in unserem Beispiel der Dateien `identity` und `identity.pub`, ausgegeben.

```
Enter same passphrase again:
Your identification has been saved in /home/hannes/.ssh/identity.
Your public key has been saved in /home/hannes/.ssh/identity.pub.
The key fingerprint is:
79:c1:79:b2:e1:c8:20:c1:89:0f:99:94:a8:4e:da:e8 hannes@sonne
```

Insbesondere, wenn der private Schlüssel (identity) auf einem nicht von Ihnen selbst administrierten System erzeugt wird und abgelegt ist, oder Sie Ihr Benutzerverzeichnis per NFS beziehen, sollten Sie eine Passphrase benutzen. Verwenden Sie **`ssh-keygen -p`** , um Ihre alte Passphrase zu ändern.

Kopieren Sie den öffentlichen Teil des Schlüssels (identity.pub) auf den entfernten Rechner und legen Sie ihn dort als `~/.ssh/authorized_keys` ab. Zur Authentifizierung werden Sie beim nächsten Verbindungsaufbau nach Ihrer Passphrase gefragt. Sollte dies nicht der Fall sein, überprüfen Sie bitte Ort und Inhalt der zuvor erwähnten Dateien.

Auf Dauer ist diese Vorgehensweise mühsamer, als die Eingabe eines Passwortes. Entsprechend liefert das SSH-Paket ein weiteres Hilfsprogramm, den ssh-agent, der für die Dauer einer „X-session“ private Schlüssel bereit hält. Dazu wird das gesamte X als Kindprozess des ssh-agents gestartet. Sie erreichen dies am einfachsten, indem Sie am Anfang der Datei `.xsession` die Variable `usessh` auf `yes` setzen und sich über einen Displaymanager, z. B. KDM oder XDM, anmelden. Alternativ können Sie ssh-agent startx verwenden.

Sobald Ihre X-session gestartet ist, schalten Sie Ihren privaten Schlüssel mittels ssh-add frei. Insoweit ssh-add nicht auf ein Terminal zugreifen kann, z. B. über ein Menü aufgerufen wird, oder eine Eingabeumleitung von **`</dev/null`** erfolgt, erscheint eine grafische Eingabeaufforderung `x11-ssh-askpass`. Nun können Sie wie gewohnt ssh oder scp nutzen. Insoweit Sie Ihren privaten Schlüssel wie zuvor verteilt haben, sollten Sie jetzt nicht mehr nach dem Passwort gefragt werden. Achten Sie beim Verlassen Ihres Rechner darauf, dass Sie Ihre X-session beenden oder mittels einer passwortgeschützten Bildschirmsperre, z. B. xlock, verriegeln.

### 8.2.7 X-, Authentifizierungs- und sonstige Weiterleitung

Über die bisher beschriebenen sicherheitsrelevanten Verbesserungen hinaus erleichtert ssh auch die Verwendung von entfernten X-Anwendungen. Insoweit Sie **`ssh`** mit der Option **`-X`** aufrufen, wird auf dem entfernten System automatisch die DISPLAY-Variable gesetzt und alle X-Ausgaben werden durch die bestehende ssh-Verbindung auf den Ausgangsrechner weitergeleitet. Diese bequeme Funktion unterbindet gleichzeitig die bisher bestehenden Abhörmöglichkeiten bei entfernt aufgerufenen und lokal betrachteten X-Anwendungen.

Durch die gesetzte Option **`-A`** wird der ssh-agent-Authentifizierungsmechanismus auf den nächsten Rechner mit übernommen. Man kann so von einem Rechner zum anderen gehen, ohne ein Passwort eingeben zu müssen. Allerdings nur, wenn man zuvor seinen öffentlichen Schlüssel auf die beteiligten Zielrechner verteilt und korrekt abgelegt hat.

Beide Mechanismen sind vorsichtshalber in der Voreinstellung deaktiviert, können jedoch in der systemweiten Konfigurationsdatei `/etc/ssh/sshd_config` oder der nutzereigenen `~/.ssh/config` permanent eingeschaltet werden.

Analog zur X-Weiterleitung kann man ssh zur beliebigen Umleitung von TCP/IP-Verbindungen benutzen. Als Beispiel sei hier die Weiterleitung des SMTP- und POP3-Ports aufgeführt:

```
root@erde:~ # ssh -L 25:sonne:25 sonne
```

Hier wird jede Verbindung zu „`erde` Port 25“, SMTP auf den SMTP-Port von `sonne` über den verschlüsselten Kanal weitergeleitet. Dies ist insbesondere für Nutzer von SMTP-Servern ohne SMTP-AUTH oder POP-before-SMTP-

Fähigkeiten von Nutzen. Mail kann so von jedem beliebigen Ort mit Netzanschluss zur Auslieferung durch den „heimischen“ Mailserver übertragen werden.

Analog leitet

```
root@erde:~ # ssh -L 110:sonne:110 sonne
```

alle Port 110, POP3-Anfragen an `erde` auf den POP3-Port von `sonne` weiter.

Beide Beispiele müssen Sie als Nutzer `'root'` ausführen, da auf privilegierte lokale Ports verbunden wird. Bei bestehender SSH-Verbindung wird die Post wie gewohnt als normaler Nutzer versandt und abgeholt. Der SMTP- und POP3-Host muss dabei auf localhost konfiguriert werden.

Zusätzliche Informationen können den Manualpages der einzelnen Programme und den Dateien unter `/usr/share/doc/packages/openssh` entnehmen werden.

## 8.3 Sicherheit ist Vertrauenssache

### 8.3.1 Grundlagen

Eines der grundlegendsten Leistungsmerkmale eines Linux/Unix-Systems ist, dass mehrere Benutzer (multiuser) mehrere Aufgaben zur gleichen Zeit auf demselben Rechner (multi-tasking) ausführen können. Das Betriebssystem ist darüber hinaus netzwerktransparent, sodass Benutzer oftmals gar nicht wissen, ob sich die Daten oder Applikationen, mit denen sie arbeiten, lokal auf dem Rechner befinden oder über das Netzwerk bezogen werden.

Damit mehrere Benutzer auf einem System arbeiten können, müssen ihre jeweiligen Daten auch voneinander getrennt verwaltet werden können. Hier geht es unter anderem auch um Sicherheit und den Schutz der Privatsphäre. Datensicherheit war auch schon relevant, als Computer noch nicht miteinander vernetzt waren. Bei Verlust oder Defekt der Datenträger (im Allgemeinen Festplatten) mussten wichtige Daten weiterhin verfügbar sein, auch wenn solche Defekte womöglich den vorübergehenden Ausfall einer größeren Infrastruktur zur Folge hatten.

Auch wenn sich dieses Kapitel des SuSE-Handbuchs in der Hauptsache mit der Vertraulichkeit der Daten und dem Schutz der Privatsphäre der Benutzer beschäftigt, sei betont, dass ein umfassendes Sicherheitskonzept als integralen Bestandteil immer ein regelmäßiges, funktionierendes und überprüftes Backup beinhaltet. Ohne dieses Backup der Daten wird es nicht nur im Fall eines Hardware-Defekts schwierig sein, weiterhin auf die Daten zuzugreifen, sondern insbesondere auch dann, wenn nur der Verdacht besteht, dass jemand sich unbefugterweise an den Daten zu schaffen gemacht hat.

### 8.3.2 Lokale Sicherheit und Netzwerksicherheit

Es gibt verschiedene Möglichkeiten, auf Daten zuzugreifen:

- Persönliche Kommunikation mit jemand, der über die gewünschten Informationen verfügt bzw. Zugang zu bestimmten Daten auf einem Computer hat,

- direkt an der Console eines Rechners (physikalischer Zugriff),
- über eine serielle Schnittstelle oder
- über ein Netzwerk.

Alle diese Fälle sollten eine Gemeinsamkeit haben: Sie sollten sich als Benutzer authentifizieren müssen, bevor Sie Zugriff auf die Ressourcen oder Daten bekommen. Ein Webserver mag da anders geartet sein, aber Sie wollen sicherlich nicht, dass der Webserver Ihre persönlichen Daten an Surfer preisgibt. Eine SuSE-Linux Installation ließe sich mit wenigen Handgriffen dazu bringen, Sie nach dem Systemstart direkt und ohne Passwort mit Ihrer Arbeitsoberfläche zu konfrontieren, aber dieses Vorgehen ist meistens unangemessen. Damit könnte jemand in Ihrem Namen Daten manipulieren und Programme ausführen.

Der erste Fall der oben genannten ist der menschlichste von allen: Etwa bei einer Bank müssen Sie einem Angestellten beweisen, dass Ihnen der Zugriff auf Ihre Konten gestattet ist, indem Sie mit Ihrer Unterschrift, einer PIN oder mit einem Passwort beweisen, dass Sie derjenige sind, für den Sie sich ausgeben. In manchen Fällen kann es gelingen, durch das Erwähnen von Kenntnissen oder durch geschickte Rhetorik das Vertrauen eines Wissensträgers zu erschleichen, so dass dieser weitere Information preisgibt, womöglich ohne dass das Opfer dies bemerkt.

Manche Menschen sind so unvorsichtig mit ihren Äußerungen und unbewusst mit ihren Antworten, dass auch die Antworten, die sie für nicht beantwortet halten, genug Information enthalten, um Fragen immer präziser zu stellen, weil wie in einem Mosaik immer mehr Details bekannt werden. („Nein, der Herr Meier ist im Urlaub und kommt erst in drei Wochen wieder. Und im übrigen ist er nicht mein Chef, zumal er im vierten Stock sitzt und ich im dritten!“) Man nennt dies in Hackerkreisen „Social Engineering“. Gegen diese Art von Angriff hilft nur Aufklärung und ein bewusster Umgang mit Information und Sprache. Einbrüchen auf Rechnersystemem geht oft eine Art Social-Engineering-Angriff, etwa auf das Empfangspersonal, Dienstleister in der Firma oder auch Familienmitglieder, voraus, der erst viel später bemerkt wird.

Jemand, der (unbefugt) Zugriff auf Daten erlangen will, könnte auch die traditionellste Methode benutzen, denn die Hardware selbst ist ein Angriffspunkt. Der Rechner muss gegen Entnahme, Austausch und Sabotage von Teilen und Gesamteinheit (und dem Backup der Daten!) sicher verstaut sein - dazu kann auch eine eventuell vorhandene Netzwerkleitung oder ein Stromkabel gehören. Außerdem muss der Startvorgang muss abgesichert sein, denn allgemein bekannte Tastenkombinationen können den Rechner zu speziellen Reaktionen veranlassen. Dagegen hilft das Setzen von BIOS- und Bootloaderpasswörtern.

Serielle Schnittstellen mit seriellen Terminals sind heute zwar immer noch gebräuchlich, werden aber kaum noch an neuen Arbeitsplätzen installiert. Ein serielles Terminal stellt eine besondere Art des Zugriffs dar: Es ist keine Netzwerkschnittstelle, da kein Netzwerkprotokoll zur Kommunikation zwischen den Systemeinheiten verwendet wird. Ein simples Kabel (oder eine Infrarotschnittstelle) wird als Übertragungsmedium für einfache Zeichen verwendet. Das Kabel selbst ist dabei der einfachste Angriffspunkt: Man muss nur einen alten Drucker daran

anschließen und kann die Kommunikation aufzeichnen. Was mit einem Drucker möglich ist, geht selbstverständlich mit beliebigem Aufwand auch anders.

Netzwerke vereinfachen uns den Zugriff auf Daten mit zum Teil komplexen Kommunikationsprotokollen. Das mag paradox klingen, muss aber so sein. Wenn Sie völlig vom Ort unabhängig sein wollen und einen Rechner fernsteuern oder Daten von ihm beziehen wollen, dann brauchen Sie abstrakte, modulare Modelle, deren Ebenen weitgehend voneinander unabhängig sind. Im täglichen Umgang mit Computern begegnen Sie ständig solchen Modellen: Modularität ist, wenn Ihr Textverarbeitungsprogramm nicht wissen muss, welche Art von Festplatte Sie haben, und Ihr E-Mail-Programm sollte sich nicht darum kümmern müssen, ob Sie nun ein Modem oder eine Ethernet-Karte haben. Teile Ihres Betriebssystems (in unserem Fall Linux) stellen Ihnen die Funktionalität mittels einer definierten Schnittstelle zur Verfügung und kümmern sich um die Details. So kann einerseits ein Textverarbeitungsprogramm oder ein Mail-User-Agent (MUA) auch auf Rechnern mit gänzlich unterschiedlicher Hardware funktionieren, und andererseits können sie von einem beliebigen Ort aus betrieben werden, die nötige technische Ausstattung vorausgesetzt.

In Hinblick auf die Daten bedeutet dies, dass es keinen Unterschied macht, ob eine Datei in der Kommandozeile geöffnet oder mit einem Webbrowser betrachtet wird. Genauso kann man sich über ein Netzwerk (etwa mit einem telnet-Programm oder, viel besser, mit einem secure shell Programm (ssh), das den Netzverkehr vollständig verschlüsselt) einloggen und die Datei lesen. Dennoch müssten dabei mehrere Hürden übersprungen werden. Zunächst müssen Netzwerk und Rechner verbunden werden, dann muss sich der Benutzer authentifizieren (die Identität nachweisen). Dabei schränken schließlich noch die Zugriffsrechte der Datei die Handlungsmöglichkeiten ein.

Da das Öffnen einer Datei auf einem Rechner anderen Zugriffsbeschränkungen unterliegt als das Öffnen einer Netzwerkverbindung zu einem Dienst auf einem Rechner, ist es nötig, zwischen lokaler Sicherheit und Netzwerksicherheit zu unterscheiden. Die Trennlinie ist da markiert, wo Daten in Pakete verschnürt werden müssen, um verschickt zu werden und zur Anwendung zu gelangen.

### Lokale Sicherheit

Lokale Sicherheit mit den physikalischen Gegebenheiten, in denen der Rechner aufgestellt ist. Wir gehen davon aus, dass Sie Ihren Rechner so aufgebaut haben, dass das Maß an Sicherheit Ihrem Anspruch und Ihren Anforderungen genügt. In Bezug auf „Lokale Sicherheit“ besteht die Aufgabe darin, die einzelnen Benutzer voneinander zu trennen, sodass kein Benutzer die Rechte oder oder die Identität eines anderen Benutzen annehmen kann. Dies gilt im Allgemeinen, im Speziellen sind natürlich besonders `'root'`-Rechte gemeint, da der Benutzer `'root'` im System Allmacht hat; er kann unter anderem ohne Passwort zu jedem lokalen Benutzer werden und jede lokale Datei lesen.

Die Liste der Möglichkeiten, ein System anzugreifen, wenn man bereits Zugriff auf lokale Ressourcen über die Kommandozeile hat, ist recht lang.

### Passwörter

Ihr Linux-System speichert Passwörter nicht etwa im Klartext ab und vergleicht ein eingegebenes Passwort mit dem, was gespeichert ist. Bei einem Diebstahl der Datei, in der die Passwörter stehen, wären dann ja alle Accounts auf Ihrem System kompromittiert. Stattdessen wird Ihr Passwort verschlüsselt abgelegt und jedes Mal, wenn Sie das Passwort eingegeben haben, wird dieses wieder verschlüsselt und das Ergebnis verglichen mit dem, was als verschlüsseltes Passwort abgespeichert ist. Dies macht natürlich nur dann Sinn, wenn man aus dem verschlüsselten Passwort nicht das Klartextpasswort errechnen kann. Dies erreicht man durch so genannte „Falltüralgorithmen“, die nur in eine Richtung funktionieren. Ein Angreifer, der das verschlüsselte Passwort in seinen Besitz gebracht hat, kann nicht einfach zurückrechnen und das Passwort sehen, sondern er muss alle möglichen Buchstabenkombinationen für ein Passwort durchprobieren, bis er dasjenige findet, welches verschlüsselt so aussieht wie Ihres. Bei acht Buchstaben pro Passwort gibt es beträchtlich viele Kombinationen.

Mit ein Argument für die Sicherheit dieser Methode in den 70er Jahren war, dass der verwendete Algorithmus recht langsam ist und Zeit im Sekundenbereich für das verschlüsseln von einem Passwort brauchte. Heutige PCs schaffen ohne weiteres mehrere hunderttausend bis Millionen Verschlüsselungen pro Sekunde. Aus diesem Grund dürfen verschlüsselte Passwörter nicht für jeden Benutzer sichtbar sein (`/etc/shadow` ist für einen normalen Benutzer nicht lesbar), und die Passwörter dürfen nicht leicht zu erraten sein, für den Fall, dass die verschlüsselten Passwörter wegen eines Fehlers eben doch sichtbar werden. Ein Passwort wie „Phantasie“ umzuschreiben in „Ph@nt@s13“ hilft nicht viel: Solche Vertauschungsregeln können von Knackprogrammen, die Wörterbücher zum Raten benutzen, sehr leicht aufgelöst werden. Besser sind Kombinationen von Buchstaben, die kein bekanntes Wort bilden und nur für den Benutzer eine persönliche Bedeutung haben, etwa die Anfangsbuchstaben der Wörter eines Satzes, oder nehmen Sie zum Beispiel einen Buchtitel wie „Der Name der Rose“ von Umberto Eco. Daraus gewinnen Sie ein gutes Passwort: „DNdRvUE9“. Ein Passwort wie „Bierjunge“ oder „Jasmin76“ würde schon jemand erraten können, der Sie oberflächlich gut kennt.

### Der Bootvorgang

Verhindern Sie, dass mit einer Diskette oder einer CD-ROM gebootet werden kann, indem Sie die Laufwerke ausbauen oder indem Sie ein BIOS-Passwort setzen und im BIOS ausschließlich das Booten von Festplatte erlauben.

Linux Systeme starten gewöhnlicherweise mit einem Boot-loader, der es erlaubt, zusätzliche Optionen an den zu startenden Kernel weiterzugeben. Solche Optionen sind im hohem Maße sicherheitskritisch, weil der Kernel ja nicht nur mit `'root'`-Rechten läuft, sondern die `'root'`-Rechte von Anfang an vergibt. Verhindern Sie, dass jemand solche Optionen verwendet, während Ihr Rechner startet, indem Sie die Optionen „restricted“ und „password=irgendein_passwort“ in `/etc/lilo.conf` verwenden. Vergessen Sie nicht, das Kommando **`lilo`** auszuführen, wenn Sie die Datei `/etc/lilo.conf` verändert haben, und achten Sie auf die Ausgaben des Programms! Wenn Sie das Passwort vergessen,

müssen Sie das BIOS-Passwort kennen und von CD booten, um den Eintrag in `/etc/lilo.conf` aus einem Rettungssystem heraus zu lesen.

### Zugriffsrechte

Es gilt das Prinzip, immer mit den niedrigst möglichen Privilegien für eine jeweilige Aufgabe zu arbeiten. Es ist definitiv nicht nötig, seine E-Mails als root zu lesen und zu schreiben. Wenn das Mailprogramm (MUA = Mail User Agent), mit dem Sie arbeiten, einen Fehler hat, dann wirkt sich dieser Fehler mit genau den Rechten aus, die Sie zum Zeitpunkt des des Angriffs hatten. Hier geht es also auch um Schadensminimierung.

Die einzelnen Rechte der weit über 200.000 Dateien einer SuSE-Distribution sind sorgfältig vergeben. Der Administrator eines Systems sollte zusätzliche Software oder andere Dateien nur unter größtmöglicher Sorgfalt installieren und besonders gut auf die vergebenen Rechte der Dateien achten. Erfahrene und sicherheitsbewusste Admins verwenden bei dem Kommando **`ls`** stets die Option **`-l`** für eine ausführliche Liste der Dateien mitsamt den Zugriffsrechten, so dass sie eventuell falsch gesetzte Dateirechte gleich erkennen können. Ein falsch gesetztes Attribut bedeutet nicht nur, dass Dateien überschrieben oder gelöscht werden können, sondern auch dass die veränderten Dateien von `'root'` ausgeführt oder im Fall von Konfigurationsdateien von Programmen als `'root'` benutzt werden können. Damit könnte ein Angreifer seine Rechte beträchtlich ausweiten. Man nennt solche Angriffe dann Kukukseier, weil das Programm (das Ei) von einem fremden Benutzer (Vogel) ausgeführt (ausgebrütet) wird, ähnlich wie der Kukuk seine Eier von fremden Vögeln ausbrüten lässt.

SuSE-Systeme verfügen über die Dateien `permissions`, `permissions.easy`, `permissions.secure` und `permissions.paranoid` im Verzeichnis `/etc`. In diesen Dateien werden besondere Rechte wie etwa welt-schreibbare Verzeichnisse oder für Dateien setuser-ID-bits festgelegt, d.h. das Programm läuft dann nicht mit der Berechtigung des Eigentümers des Prozesses, der es gestartet hat, sondern mit der Berechtigung des Eigentümers der Datei, und das ist in der Regel `'root'`. Für den Administrator steht die Datei `/etc/permissions.local` zur Verfügung, in der er seine eigenen Änderungen festhalten kann. Die Variable **`PERMISSION_SECURITY`** aus der Datei `/etc/rc.config` legt fest, welche der Dateien für Konfigurationsprogrammen von SuSE zur Vergabe der Rechte benutzt werden sollen. Diese Auswahl können Sie auch komfortabel unter dem Menüpunkt '`Sicherheit`' von YaST1 und YaST2 treffen. Mehr zu diesem Thema erfahren Sie direkt aus der Datei `/etc/permissions` und der Manpage des Kommandos **`chmod`** (**`man chmod`**).

### file race conditions

Ein Programm will eine Datei in einem Verzeichnis anlegen, welches für jedermann schreibbar ist (wie `/tmp`). Es überprüft, ob die Datei bereits existiert und erzeugt die Datei, wenn sie noch nicht vorhanden war. Zwischen dem Überprüfen der Existenz und dem Anlegen der Datei vergeht aber eine kurze Zeit, in der ein Angreifer einen symbolischen Link anlegen kann, einen Zeiger auf eine

andere Datei. Das Programm verfolgt dann diesen symbolischen Link und überschreibt dabei die Zieldatei mit seinen Privilegien. Dies ist ein Rennen ((engl. *race*)), weil für den Angreifer nur eine kurze Zeit bleibt, in der er den „symlink“ anlegen kann. Dieses Rennen ist nur dann möglich, wenn der Vorgang von Überprüfen und Anlegen einer Datei nicht atomisch, also unteilbar ist. Wenn das Rennen stattfinden kann, dann kann es von einem Angreifer auch gewonnen werden, das ist eine Frage der Wahrscheinlichkeit.

#### Buffer overflows, format string bugs

Wann immer ein Programm Daten verarbeitet, die in beliebiger Form unter Einfluss eines Benutzers stehen oder standen, ist besondere Vorsicht geboten. Diese Vorsicht gilt in der Hauptsache für den Programmierer der Anwendung: Er muss sicherstellen, dass die Daten durch das Programm richtig interpretiert werden, dass die Daten zu keinem Zeitpunkt in Speicherbereiche geschrieben werden, die eigentlich zu klein sind und dass er die Daten in konsistenter Art und Weise durch sein eigenes Programm und die dafür definierten Schnittstellen weiterreicht.

Ein „Buffer Overflow“ passiert dann, wenn beim Beschreiben eines Pufferspeicherbereichs nicht darauf geachtet wird, wie groß der Puffer eigentlich ist. Es könnte sein, dass die Daten (die vom Benutzer kamen) etwas mehr Platz verlangen, als im Puffer zur Verfügung steht. Durch dieses Überschreiben des Puffers über seine Grenze hinaus ist es unter Umständen möglich, dass ein Programm aufgrund der Daten, die er eigentlich nur verarbeiten soll, Programmsequenzen ausführt, die unter dem Einfluss des Users und nicht des Programmierers stehen. Dies ist ein schwerer Fehler, insbesondere wenn das Programm mit besonderen Rechten abläuft (siehe Abschnitt 8.3.2 auf der vorherigen Seite). „Format String Bugs“ funktionieren etwas anders, verwenden aber wieder user-input, um das Programm von seinem eigentlichen Weg abzubringen.

Diese Programmierfehler werden normalerweise bei Programmen ausgebeutet (engl. *exploit*), die mit gehobenen Privilegien ausgeführt werden, also setuid- und setgid-Programme. Sie können sich und Ihr System also vor solchen Fehlern schützen, indem Sie die besonderen Ausführungsrechte von den Programmen entfernen. Auch hier gilt wieder das Prinzip der geringstmöglichen Privilegien (siehe Abschnitt 8.3.2 auf der vorherigen Seite).

Da „Buffer Overflows“ und „Format String Bugs“ Fehler bei der Behandlung von Benutzerdaten sind, sind sie nicht notwendigerweise nur ausbeutbar, wenn man bereits Zugriff auf ein lokales „login“ hat. Viele der bekannt gewordenen Fehler können über eine Netzwerkverbindung ausgenutzt werden. Deswegen sind „Buffer Overflows“ und „Format String Bugs“ nicht direkt auf den lokalen Rechner oder das Netzwerk klassifizierbar.

#### Viren

Entgegen andersartiger Verlautbarungen gibt es tatsächlich Viren für Linux. Die bekannten Viren sind von ihren Autoren als „Proof-of-Concept“ geschrieben

worden, als Beweis, dass die Technik funktioniert. Allerdings ist noch keiner dieser Viren in „freier Wildbahn“ beobachtet worden.

Viren benötigen zur Ausbreitung einen Wirt, ohne den sie nicht überlebensfähig sind. Dieser Wirt ist ein Programm oder ein wichtiger Speicherbereich für das System, etwa der Master-Boot-Record, und er muss für den Programmcode des Virus beschreibbar sein. Linux hat aufgrund seiner Multi-User Fähigkeiten die Möglichkeit, den Schreibzugriff auf Dateien einzuschränken, also insbesondere Systemdateien. Wenn Sie als `'root'` arbeiten, erhöhen Sie also die Wahrscheinlichkeit, dass Ihr System von solch einem Virus infiziert wird. Berücksichtigen Sie aber die Regel der geringstmöglichen Privilegien, ist es Schwierigkeiten unter Linux einen Virus zu bekommen. Darüber hinaus sollten Sie nie leichtfertig ein Programm ausführen, das Sie aus dem Internet bezogen haben und dessen genaue Herkunft Sie nicht kennen. SuSE-rpm Pakete sind kryptographisch signiert und tragen mit dieser digitalen Unterschrift das Markenzeichen der Sorgfalt beim Bau der Pakete bei SuSE. Viren sind klassische Symptome dafür, dass auch ein hochsicheres System unsicher wird, wenn der Administrator oder auch der Benutzer ein mangelndes Sicherheitsbewusstsein hat.

Viren sind nicht mit Würmern zu verwechseln, die Phänomene auch der Netzwerksicherheit sind und keinen Wirt brauchen, um sich zu verbreiten.

### Netzwerksicherheit

Bei der lokalen Sicherheit war es die Aufgabe, die Benutzer, die an demselben Rechner arbeiten, voneinander zu trennen, insbesondere den Benutzer `'root'`. Im Gegensatz dazu soll bei der Netzwerksicherheit das ganze System gegen Angriffe über das Netzwerk geschützt werden. Benutzerauthentifizierung beim klassischen Einloggen durch Benutzerkennung und Passwort gehört zur lokalen Sicherheit. Beim Einloggen über eine Netzwerkverbindung muss man differenzieren zwischen beiden Sicherheitsaspekten: bis zur erfolgten Authentifizierung spricht man von Netzwerksicherheit, nach dem Login geht es um lokale Sicherheit.

### X-Windows (X11-Authentifizierung)

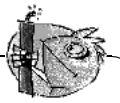
Wie bereits erwähnt ist Netzwerktransparenz eine grundlegende Eigenschaft eines Unix-Systems. Bei X11, dem Windowing-System von Unix, gilt dies in besonderem Maße! Sie können sich ohne Weiteres auf einem entfernten Rechner einloggen und dort ein Programm starten, welches dann über das Netzwerk auf Ihrem Rechner angezeigt wird. Das Protokoll, welches zwischen der X-Applikation und dem X-Server (der lokale Prozess, der die Fenster auf der Grafikkarte zur Anzeige bringt) zur Kommunikation verwendet wird, ist recht sparsam, was Netzwerkbandbreiten angeht. Das ist durch die in den 80er Jahren, als das System entworfen wurde, zur Verfügung stehenden Bandbreiten bedingt.

Wenn nun ein X-Client über das Netzwerk bei unserem X-Server angezeigt werden soll, dann muss der Server die Ressource, die er verwaltet (das Display), gegen unberechtigte Zugriffe schützen. Konkret heißt das hier, dass das Client-Programm Rechte bekommen muss. Bei X-Windows geschieht dies auf zwei

verschiedene Arten: Host-basierte und cookie-basierte Zugriffskontrolle. Erstere basiert auf der IP-Adresse des Rechners, auf dem das Client-Programm laufen soll und wird mit dem Programm `xhost` kontrolliert. Das Programm `xhost` trägt eine IP-Adresse eines legitimen Client in eine Mini-Datenbank im X-Server ein. Eine Authentifizierung einzig und allein auf einer IP-Adresse aufzubauen gilt jedoch nicht gerade als sicher. Es könnte noch ein zweiter Benutzer auf dem Rechner mit dem Client-Programm aktiv sein, und dieser hätte dann genau wie jemand, der die IP-Adresse stiehlt, Zugriff auf den X-Server. Deswegen soll hier auch nicht näher auf diese Methoden eingegangen werden. Die Manpage des **`xhost`**-Kommandos gibt mehr Aufschluss über die Funktionsweise (und enthält ebenfalls die Warnung!).

Bei „cookie“-basierter Zugriffskontrolle wird eine Zeichenkette, die nur der X-Server und der legitim eingeloggte Benutzer kennen, als einem Passwort ähnliches Ausweismittel verwendet. Dieses „cookie“ (das englische Wort cookie bedeutet Keks und meint hier die chinesischen fortune cookies, die einen Spruch enthalten) wird in der Datei `.Xauthority` im home-Verzeichnis des Benutzers beim login abgespeichert und steht somit jedem X-Windows-client, der ein Fenster beim X-Server zur Anzeige bringen will, zur Verfügung. Das Programm `xauth` gibt dem Benutzer das Werkzeug, die Datei `.Xauthority` zu untersuchen. Wenn Sie `.Xauthority` aus Ihrem home-Verzeichnis löschen oder umbenennen, dann können Sie keine weiteren Fenster von neuen X-Clients mehr öffnen. Näheres über Sicherheitsaspekte von X-Windows erfahren Sie in der manpage von `Xsecurity` (**`man Xsecurity`**).

`ssh` (secure shell) kann über eine vollständig verschlüsselte Netzverbindung für einen Benutzer transparent (also nicht direkt sichtbar) die Verbindung zu einem X-Server weiterleiten. Man spricht von „X11-forwarding“. Dabei wird auf der Server-Seite ein X-Server simuliert und bei der Shell auf der remote-Seite die DISPLAY-Variable gesetzt. Der Client öffnet zum Anzeigen dann eine Verbindung zum sshd (secure shell daemon, das serverseitige Programm), der dann die Verbindung an den richtigen, realen X-Server durchschleust. Wenn Sie X-Clients über das Netzwerk anzeigen lassen müssen, dann sollten Sie `ssh` einmal genauer unter die Lupe nehmen. Die manpage von `ssh` gibt weitere Auskünfte über diese Funktionalität.



**Achtung**

Wenn Sie den Rechner, auf dem Sie sich einloggen, nicht als sicher betrachten, dann sollten Sie auch keine X-Windows-Verbindungen weiterleiten lassen. Mit eingeschaltetem „X11-forwarding“ könnten sich auch Angreifer über Ihre ssh-Verbindung mit Ihrem X-Server authentifiziert verbinden und beispielsweise Ihre Tastatur belauschen.

Weiter Informationen zu `ssh` finden Sie im Abschnitt 8.2 auf Seite 162 dieses Buches.

### Buffer Overflows und Format String Bugs

Nicht direkt klassifizierbar in lokal und remote gilt das im Abschnitt „Lokale Sicherheit“ über „Buffer Overflows“ und „Format String Bugs“ Gesagte äquivalent für Netzwerksicherheit. Wie auch bei den lokalen Varianten dieser Programmierfehler führen Buffer Overflows bei Netzwerkdiensten meistens zu `'root'`-Rechten. Sollte dies nicht der Fall sein, dann könnte sich der Angreifer zumindest Zugang zu einem unpriviligierten lokalen Account verschaffen, mit dem er dann weitere (lokale) Sicherheitsprobleme ausnutzen kann, falls diese vorhanden sind.

Über das Netzwerk ausbeutbare Buffer Overflows und Format String Bugs sind wohl die häufigsten Varianten von remote-Angriffen überhaupt. Auf Sicherheitsmailinglisten werden so genannte „exploits“ herumgereicht, d.h. Programme, die die frisch gefundenen Lücken ausnutzen. Auch jemand, der nicht die genauen Details der Lücke kennt, kann damit die Lücke ausnutzen. Im Laufe der Jahre hat sich herausgestellt, dass die freie Verfügbarkeit von „exploitcodes“ generell die Sicherheit von Betriebssystemen erhöht hat, was sicherlich daran liegt, dass Betriebssystemhersteller dazu gezwungen waren, die Probleme in ihrer Software zu beseitigen. Da bei freier Software der Sourcecode für jedermann erhältlich ist (SuSE-Linux liefert alle verfügbaren Quellen mit), kann jemand, der eine Lücke mitsamt „exploitcode“ findet, auch gleichzeitig noch einen Reparaturvorschlag für das Problem anbieten.

### DoS — Denial of Service

Ziel dieser Art von Angriff ist das Einstellen des Dienstes (oder gleich des ganzen Systems). Dies kann auf verschiedenste Arten passieren: Durch Überlastung, durch Beschäftigung mit unsinnigen Paketen oder durch Ausnutzen von „Remote Buffer Overflows“, die nicht direkt zum Ausführen von Programmen auf der remote-Seite ausbeutbar sind.

Der Zweck eines DoS mag meistens darin begründet sein, dass der Dienst einfach nicht mehr verfügbar ist. Dass ein Dienst fehlt, kann aber weitere Konsequenzen haben. Siehe „man in the middle: sniffing, tcp connection hijacking, spoofing“ und „DNS poisoning“.

### man in the middle: sniffing, tcp connection hijacking, spoofing

Ganz allgemein gilt: Ein Angriff vom Netzwerk, bei der der Angreifer eine Position zwischen zwei Kommunikationspartnern einnimmt, nennt sich „man in the middle attack“. Sie haben in der Regel eines gemeinsam: Das Opfer merkt nichts davon. Viele Varianten sind denkbar: Der Angreifer nimmt die Verbindung entgegen und stellt, damit das Opfer nichts merkt, selbst eine Verbindung zum Ziel her. Das Opfer hat also, ohne es zu wissen, eine Netzwerkverbindung zum falschen Rechner geöffnet, weil dieser sich als das Ziel ausgibt. Der einfachste „man in the middle attack“ ist ein „sniffer“. Er belauscht einfach nur die Netzverbindungen, die an ihm vorüber geführt werden (sniffing = engl. schnüffeln). Komplexer wird es, wenn der Angreifer in der Mitte versucht, eine etablierte, bestehende Verbindung zu übernehmen (entführen = engl. hijacking). Dafür muss

der Angreifer die Pakete, die an ihm vorbeigeführt werden, eine Weile lang analysiert haben, damit er die richtigen TCP-Sequenznummern der TCP-Verbindung vorhersagen kann. Wenn er dann die Rolle des Ziels der Verbindung übernimmt, merkt das das Opfer, weil auf der Seite des Opfers die Verbindung als ungültig terminiert wird.

Der Angreifer profitiert dabei insbesondere bei Protokollen, die nicht kryptographisch gegen „hijacking“ gesichert sind und bei denen zu Beginn der Verbindung einen Authentifizierung stattfindet. „Spoofing“ nennt sich das Verschicken von Paketen mit modifizierten Absenderdaten, also hauptsächlich der IP Adresse. Die meisten aktiven Angriffsvarianten verlangen das Verschicken von gefälschten Paketen, was unter Unix/Linux übrigens nur der Superuser (`'root'`) darf.

Viele der Angriffsmöglichkeiten kommen in Kombination mit einem DoS vor. Gibt es eine Möglichkeit, einen Rechner schlagartig vom Netzwerk zu trennen (wenn auch nur für kurze Zeit), dann wirkt sich das förderlich auf einen aktiven Angriff aus, weil keine Störungen mehr erwartet werden müssen.

#### DNS poisoning

Der Angreifer versucht, mit gefälschten („gespooften“) DNS-Antwortpaketen den cache eines DNS-Servers zu vergiften (engl. *poisoning*), so dass dieser die gewünschte Information an ein Opfer weitergibt, das danach fragt. Um einem DNS-Server solche falschen Informationen glaubhaft zuschieben zu können, muss der Angreifer normalerweise einige Pakete des Servers bekommen und analysieren. Weil viele Server ein Vertrauensverhältnis zu anderen Rechnern aufgrund ihrer IP Adresse oder ihres Hostnamens konfiguriert haben, kann ein solcher Angriff trotz eines gehörigen Aufwands recht schnell Früchte tragen. Voraussetzung ist allerdings eine gute Kenntnis der Vertrauensstruktur zwischen diesen Rechnern. Ein zeitlich genau abgestimmter DoS gegen einen DNS-Server, dessen Daten gefälscht werden sollen, ist aus Sicht des Angreifers meistens nicht vermeidbar.

Abhilfe schafft wieder eine kryptographisch verschlüsselte Verbindung, die die Identität des Ziels der Verbindung verifizieren kann.

#### Würmer

Würmer werden häufig mit Viren gleichgesetzt. Es gibt aber einen markanten Unterschied: Ein Wurm muss keinerlei Wirtsprogramm infizieren, und er ist darauf spezialisiert, sich möglichst schnell im Netzwerk zu verbreiten. Bekannte Würmer wie Ramen, Lion oder Adore nutzen wohlbekannte Sicherheitslücken von Serverprogrammen wie `bind8` oder `lprNG`. Man kann sich relativ einfach gegen Würmer schützen, weil zwischen dem Zeitpunkt des Bekanntwerdens der ausgenutzten Lücken bis zum Auftauchen des Wurms normalerweise einige Tage vergehen, so dass update-Pakete vorhanden sind. Natürlich setzt dies voraus, dass der Administrator die Security-updates auch in seine Systeme einspielt.

### 8.3.3 Tipps und Tricks: Allgemeine Hinweise

**Information:** Für einen effizienten Umgang mit dem Bereich Sicherheit ist es nötig, mit den Entwicklungen Schritt zu halten und auf dem Laufenden zu sein, was die neuesten Sicherheitsprobleme angeht. Ein sehr guter Schutz gegen Fehler aller Art ist das schnellstmögliche Einspielen von update-Paketen, die von einem Security-Announcement angekündigt werden. Die SuSE-security-Announcements werden über eine Mailingliste verbreitet, in die Sie sich, den Links unter `http://www.suse.de/security` folgend, eintragen können. `suse-security-announce@suse.de` ist die erste Informationsquelle für update-Pakete, die vom Security-Team mit neuen Informationen beliefert wird.

Die Mailingliste `suse-security@suse.de` ist ein lehrreiches Diskussionsforum für den Bereich Sicherheit. Sie können sich auf der gleichen URL wie für `suse-security-announce@suse.de` für die Liste anmelden.

Eine der bekanntesten Sicherheitsmailinglisten der Welt ist die Liste `bugtraq@securityfocus.com`. Die Lektüre dieser Liste bei durchschnittlich 15-20 Postings am Tag kann mit gutem Gewissen empfohlen werden. Mehr Information finden Sie auf `http://www.securityfocus.com`.

Einige Grundregeln, die zu kennen nützlich sein kann, sind nachstehend aufgeführt:

- Vermeiden Sie es, als `'root'` zu arbeiten, entsprechend dem Prinzip, die geringstnötigen Privilegien für eine Aufgabe zu benutzen. Das verringert die Chancen für ein Kukuksei oder einen Virus, und überdies für Fehler Ihrerseits.
- Benutzen Sie nach Möglichkeit immer verschlüsselte Verbindungen, um Arbeiten remote auszuführen. „ssh“ (secure shell) ist Standard, vermeiden Sie `telnet`, `ftp`, `rsh` und `rlogin`.
- Benutzen Sie keine Authentifizierungsmethoden, die alleine auf der IP-Adresse aufgebaut sind.
- Halten Sie Ihre wichtigsten Pakete für den Netzwerkbereich immer auf dem neuesten Stand und abonnieren Sie die Mailinglisten für Announcements der jeweiligen Software (z. B. Beispiel `bind`, `sendmail`, `ssh`). Dasselbe gilt für Software, die nur lokale Sicherheitsrelevanz hat.
- Optimieren Sie die Zugriffsrechte auf sicherheitskritische Dateien im System, indem Sie die `/etc/permissions`-Datei Ihrer Wahl an Ihre Bedürfnisse anpassen. Ein setuid-Programm, welches kein setuid-bit mehr hat, mag zwar nicht mehr wirklich seine Aufgabe erledigen können, aber es ist in der Regel kein Sicherheitsproblem mehr. Mit einer ähnlichen Vorgehensweise können Sie auf welt-schreibbare Dateien und Verzeichnisse losgehen.
- Deaktivieren Sie jegliche Netzwerkdienste, die Sie auf Ihrem Server nicht zwingend brauchen. Das macht Ihr System sicherer, und es verhindert, dass Ihre Benutzer sich an einen Dienst gewöhnen, den Sie nie absichtlich freigegeben haben (legacy-Problem). Offene Ports (mit socket-Zustand LISTEN) finden Sie mit dem Programm `netstat`. Als Optionen bietet sich an,

**netstat -ap** oder **netstat -anp** zu verwenden. Mit der **-p**-Option können Sie gleich sehen, welcher Prozess mit welchem Namen den Port belegt.

Vergleichen Sie die Ergebnisse, die Sie haben, mit einem vollständigen Portscan Ihres Rechners von außen. Das Programm `nmap` ist dafür hervorragend geeignet. Es klopft jeden einzelnen Port ab und kann anhand der Antwort Ihres Rechners Schlüsse über einen hinter dem Port wartenden Dienst ziehen. Scannen Sie niemals einen Rechner ohne das direkte Einverständnis des Administrators, denn dies könnte als aggressiver Akt aufgefasst werden. Denken Sie daran, dass Sie nicht nur TCP-Ports scannen sollten, sondern auf jeden Fall auch UDP-Ports (Optionen **-sS** und **-sU**).

- Zur zuverlässigen Integritätsprüfung der Dateien in Ihrem System sollten Sie `tripwire` benutzen und die Datenbank verschlüsseln, um sie gegen manipulative Zugriffe zu schützen. Darüber hinaus brauchen Sie auf jeden Fall ein backup dieser Datenbank außerhalb der Maschine auf einem eigenen Datenträger, der nicht über einen Rechner mit einem Netzwerk verbunden ist.

- Seien Sie vorsichtig beim Installieren von Fremdsoftware. Es gab schon Fälle, wo ein Angreifer tar-Archive einer Sicherheitssoftware mit einem trojanischen Pferd versehen hat. Zum Glück wurde dies schnell bemerkt. Wenn Sie ein binäres Paket installieren, sollten Sie sicher sein, woher das Paket kommt.

  SuSE rpm-Pakete werden gpg-signiert ausgeliefert. Der Schlüssel, den wir zum Signieren verwenden, ist

  ID:9C800ACA 2000-10-19 SuSE Package Signing Key <build@suse.de>

  Key fingerprint = 79C1 79B2 E1C8 20C1 890F 9994 A84E DAE8 9C80 0ACA

  Das Kommando **rpm -checksig paket.rpm** zeigt an, ob die Prüfsumme und die Signatur des (nicht installierten!) Pakets stimmen. Sie finden den Schlüssel auf der ersten CD einer SuSE-Distribution ab SuSE-7.1 und auf den meisten Keyservern der Welt.

- Überprüfen Sie regelmäßig Ihr Backup der Daten und des Systems. Ohne eine zuverlässige Aussage über die Funktion des Backups ist das Backup unter Umständen wertlos.

- Überwachen Sie Ihre „Logfiles“. Nach Möglichkeit sollten Sie sich ein kleines Script schreiben, welches Ihre Logfiles nach ungewöhnlichen Einträgen absucht. Diese Aufgabe ist alles andere als trivial, denn nur Sie wissen, was ungewöhnlich ist und was nicht.

- Verwenden Sie `tcp_wrapper`, um den Zugriff auf die einzelnen Dienste Ihres Rechners auf die IP-Adressen einzuschränken, denen der Zugriff auf einen bestimmten Dienst explizit gestattet ist. Nähere Information zu den `tcp_wrappern` finden Sie in der manual page von `tcpd(8)` und `hosts\_access` (**man tcpd**, **man hosts_access**).

- Als zusätzlichen Schutz zu dem `tcpd` (tcp_wrapper) könnten Sie die SuSEfirewall verwenden. Wenn Sie gar keine Dienste auf Ihrem Rechner

zur Verfügung stellen wollen, dann verwenden Sie am besten die SuSE personal-firewall. Die Konfiguration beschränkt sich auf den Namen des Netzwerkinterface, auf welchem hereinkommende Verbindungen abgelehnt werden sollen. Nähere Informationen finden Sie in der Datei `/sbin/SuSEpersonal-firewall` und in `/etc/rc.config.d/security.rc.config` sowie im Abschnitt 8.1 auf Seite 155.

- Legen Sie Ihr Sicherheitsdenken redundant aus: Eine Meldung, die zweimal eintrifft, ist besser als eine, die Sie nie sehen. Dies gilt genauso für Gespräche mit Kollegen.

### 8.3.4 Zentrale Meldung von neuen Sicherheitsproblemen

Wenn Sie ein Sicherheitsproblem finden (bitte überprüfen Sie die zur Verfügung stehenden update-Pakete), dann wenden Sie sich bitte vertrauensvoll an die E-Mail-Adresse `security@suse.de`. Bitte fügen Sie eine genaue Beschreibung des Problems bei, zusammen mit den Versionsnummern der verwendeten Pakete. Wir werden uns bemühen, Ihnen so schnell wie möglich zu antworten. Eine pgp Verschlüsselung Ihrer E-Mail ist erwünscht. Unser pgp key ist:

ID:3D25D3D9 1999-03-06 SuSE Security Team <security@suse.de>

Key fingerprint = 73 5F 2E 99 DF DB 94 C4 8F 5A A3 AE AF 22 F2 D5

Der Schlüssel liegt auch unter `http://www.suse.de/security` zum Download bereit.

# Index