Michael Calmer, Rebecca Ellis, Uwe Gansert, Dennis Geider, Viviane Glanz,
Roland Haidl, Edith Parzefall, Peter Reinhart, Ulrich Schairer,
Thomas Schraitle, Marius Tomaschewski, Steve Tomlin

SuSE Linux Firewall on CD

SuSE Inc.
580 2nd Street, #210
Oakland, CA 94607
USA

Toll free phone number within the US and Canada:	1-888-UR-LINUX (1-888-875-4689)
Phone:	+1-510-628-3380
Fax:	+1-510-628-3381
e-mail:	info@suse.com
	orders@suse.com
	webmaster@suse.com
WWW:	http://www.suse.com

Europe:

SuSE Linux Ltd.
The Kinetic Centre
Theobald Street
Borehamwood, WD6 4PJ UK

Phone:	+44-20-8387-4088
Fax:	+44-20-8387-4010
WWW:	http://www.suse.co.uk

SuSE GmbH
Schanzäckerstr. 10
D-90443 Nürnberg
Germany

Tel.:	+49-911-740-5331
Fax:	+49-911-741-7755
e-mail:	suse@suse.de
WWW:	http://www.suse.de

Michael Calmer, Rebecca Ellis, Uwe Gansert, Dennis Geider, Viviane Glanz, Roland Haidl, Edith Parzefall, Peter Reinhart, Ulrich Schairer, Thomas Schraitle, Marius Tomaschewski, Steve Tomlin

SuSE Linux Firewall on CD
1st edition 2001
(c) SuSE GmbH

Contents

1 Preface 9

2 Introduction and General Overview 11
2.1 Introduction 11
2.2 An Overview of Typical Firewall Setups 14

3 Adminhost 17
3.1 Introduction 17
3.2 Update 17
3.2.1 Update with YaST2 18
3.2.2 Update with YaST1 18
3.2.3 Closing Update 19
3.3 New installation of the SuSE Linux Firewall Adminhost 20
3.3.1 Language Selection 20
3.3.2 Selecting the Mouse 21
3.3.3 Keyboard and Time Zone 21
3.3.4 Preparing the Hard Disk 22
3.3.5 Boot Manager Configuration 23
3.3.6 Setting the `‘root’` Password 23
3.3.7 Confirming Settings – Starting the Installation 24
3.3.8 Preparing the Graphical Interface 25
3.3.9 Configuring the Network with YaST2 25
3.3.10 Manual Network Configuration 25
3.3.11 The user `‘fwadmin’` for the FAS 33
3.4 Firewall Administration System (FAS) 34
3.4.1 Logging in as fwadmin 34
3.4.2 Starting the Firewall Administration System 34
3.4.3 Creating a new firewall configuration 35
3.4.4 Editing an existing firewall configuration 59
3.4.5 Testing the configuration 59
3.4.6 Documenting the configuration, the tests and the results 60
3.5 Monitoring the Firewall 60

4 SuSE Linux Live CD for Firewall 63
4.1 Introduction 63
4.2 Description of the SuSE Linux Live CD for Firewall 64
4.3 Services on the Firewall 64
4.3.1 IPCHAINS 64
4.3.2 DNS 77
4.3.3 MAIL 77
4.3.4 HTTP proxy 77
4.3.5 FTP proxy 78
4.3.6 ssh 79
4.3.7 chroot, secumod, compartment, kernel capabilities . . . 79
4.4 The Configuration Disk 80
4.4.1 Creating the configuration disk 80
4.4.2 The configuration files 80
4.5 Boot Parameters 83

5 Implementing the Firewall 85
5.1 Booting the Firewall Host 85
5.2 Testing the Firewall 86
5.3 Operation 86
5.3.1 Internal testing 86
5.3.2 External testing 87
5.3.3 “Really” going online 87

6 Help 89
6.1 Troubleshooting 89
6.1.1 Problems Installing the Adminhost 89
6.1.2 Problems Booting the Live CD 89
6.2 Security Policy and Communication Analysis 90
6.2.1 Security Policy 90
6.2.2 Communication Analysis 91
6.3 Services Offered by SuSE Linux Solutions AG 91
6.4 Updates 92
6.5 What to Do If 93
6.5.1 Intrusion Detection and Event Display 93
6.5.2 External Attacks 95
6.5.3 Advantage of the Live Filesystem of the “SuSE Linux Firewall on CD” 96
6.6 Professional Help and Support 97

6.6.1 Support Regulations . . . 97
6.6.2 Professional Services . . . 98
6.6.3 Training . . . 99
6.6.4 Feedback . . . 99
6.6.5 Further Services . . . 99
6.7 Bibliography . . . 101

A DNS – Domain Name System 103
A.1 Starting the Name Server BIND . . . 103
A.2 The Configuration File /etc/named.conf . . . 104
A.2.1 Important configuration options in “options” . . . 105
A.2.2 The Configuration Section “Logging” . . . 106
A.2.3 The Structure of Zone Entry . . . 106
A.2.4 Structure of Zone Files . . . 107
A.3 DNS Sample Configuration . . . 110
A.4 Further Information . . . 111

B DHCP 115
B.1 The DHCP Protocol . . . 115
B.2 DHCP Software Packages . . . 116
B.3 The DHCP Server dhcpd . . . 116
B.4 Assigning Fixed IP Addresses to Hosts . . . 118
B.5 The Finer Points . . . 119

C Proxy Server: Squid 121
C.1 What is a Proxy Cache? . . . 121
C.2 Some Facts About Cache Proxying . . . 122
C.2.1 Squid and Security . . . 122
C.2.2 Multiple Caches . . . 122
C.2.3 Caching Internet Objects . . . 123
C.3 System Requirements . . . 123
C.3.1 Hard Disk . . . 123
C.3.2 RAM . . . 124
C.3.3 CPU . . . 125
C.4 Starting Squid . . . 125
C.5 The Configuration File /etc/squid.conf . . . 126
C.6 Transparent Proxy Configuration . . . 130
C.6.1 Kernel Configuration . . . 131
C.6.2 Configuration Options in /etc/squid.conf . . . 131

C.6.3 Firewall Configuration with SuSEfirewall 131
C.7 Squid and Other Programs 133
C.7.1 cachemgr.cgi 133
C.7.2 SquidGuard 135
C.7.3 Cache Report Generation with Calamaris 136
C.8 More Information on Squid 137

D Network Security 139
D.1 Masquerading and Firewalls 139
D.1.1 Masquerading Basics 140
D.1.2 Firewalling Basics 141
D.1.3 Personal-firewall 142
D.1.4 SuSEfirewall 144
D.2 SSH – Secure Shell, the Safe Alternative 147
D.2.1 The OpenSSH Package 148
D.2.2 The ssh Program 148
D.2.3 scp – Secure Copy 148
D.2.4 sftp - Secure File Transfer 149
D.2.5 The SSH Daemon (sshd) – Server-Side 149
D.2.6 SSH Authentication Mechanisms 150
D.2.7 X, Authentication and other Forwarding Mechanisms . . 151
D.3 Security as a Matter of Confidence 152
D.3.1 Basic Considerations 152
D.3.2 Local Security and Network Security 153
D.3.3 Some General Security Tips and Tricks 161
D.3.4 Using the Central Security Reporting Address 164

E YaST and SuSE Linux License Terms 165

1 Preface

Many thanks to: Jürgen Scheiderer, Carsten Höger, Remo Behn, Thomas Biege, Roman Drahtmüller, Marc Heuse and Stephan Martin.

The SuSE Linux Firewall on CD

The *SuSE Linux Firewall on CD* is a tool package allowing you set up a firewall solution for your network, and helps you with the related configuration, monitoring and administration chores. The complete functionality of the *SuSE Linux Firewall on CD* is based on Open Source programs which have been specially selected and enhanced for this purpose.

The *SuSE Linux Firewall on CD* protects your local network from illegal access, manages authorized usage of your resources and provides a controlled channel through which users on your local network are enabled to communicate with the Internet.

To reduce the likelihood of the firewall being misconfigured, we recommend using the *Firewall Administration System* (FAS), which corrects most user mistakes by performing a number of sanity checks. FAS ensures a consistent configuration without limiting the available options.

Please remember that total security cannot be achieved by any firewall, and thus can't be achieved by the "SuSE Linux Firewall on CD" either. This package is not intended to serve as a substitute for a proper security policy, and doesn't eliminate the need to administer, maintain and monitor the firewall – although it provides you with a number of tools to make these tasks much easier to perform.

Note:

Use the "SuSE Linux Firewall on CD" products at your own risk. In particular, no claims can be made for damages caused by improperly configured services on the firewall or any other such damages.

2 Introduction and General Overview

2.1 Introduction

The importance of the Internet and the possibilities of communication it provides (e-mail, chat, etc.) and information gathering offered by it seem to grow every day, and the number of companies and private persons which have access to the worldwide network is on a steady increase.

But connecting to the Internet often introduces some security concerns which should not be underestimated. Most companies rely on their own networks to exchange and process mission-critical information for in-house purposes (Intranet, databases, email etc.). Without the proper protection mechanisms in place, all this data would be widely available to the outside world as soon as the local network is hooked on the Internet – something which could obviously cause a lot of damage especially for companies.

> It's easy to run a secure computer system. You just have to disconnect all dial-up connections and permit only direct-wired terminals, put the machine and its terminals in a shielded room, and post a guard at the door.
>
> F.T. Grampp and R.H. Morris

We all know that in the real world it's impossible to run a computer system in this way. If you are an administrator, you'll know all too well about the problems arising from the increasing number of networked machines. After all, you are the person who has to deal with the situation on a daily basis. Up to now, your network may have been quite manageable. You knew the users on it, you've always been paying a lot of attention to security aspects, though the network's functionality as provided to your users was mainly based on trust relationships. And in fact, it had to be like that just to keep the administrative overhead at a reasonable level. But now after connecting to the Internet, things have changed dramatically, and so have the duties of the administrator. All at once, there are users who are completely unknown who can use resources on your network (e. g. the web server).

So you are aware that these users need to be handled in a totally different manner than your company's internal employees that you've been working with (though on the other hand it has been proven that 80% of all attacks on corporate networks are not carried out from the Internet). And there are other reasons to protect the corporate network and to restrict access to it: Hackers, or intruders, are always on the lookout for memory to store hacked software, or – worse even – contents which are prohibited by law. Not taking any measures against this could not only mean that the company could open itself up to legal prosecution, but could also put your company's good reputation at stake.

Today there are a number of different mechanisms available to prevent unauthorized access to corporate networks and resources (i. e. disk space, CPU capacity, etc.), from IP packet filters on routers to multiple-level firewall solutions complete with a demilitarized zone (DMZ).

In a general sense, of course, the word "firewall" is used for some kind of contrivance to prevent the spreading of fires. Firewalls made of bricks are found in buildings where they are used to isolate whole sections from each other, and in cars a special metal plate shields the cockpit from the engine compartment. Similarly, the purpose of Internet firewalls is to fend off attacks directed at your Intranet, as well as to regulate and protect clients on your LAN by imposing an access policy.

The first ever firewall was a non-routing UNIX host connected to two different networks: One network interface was connected to the Internet, and the other one to a private LAN. To reach the Internet from within the private network, users had to log on to the UNIX firewall server first before they could access any outside host. To do so, they would start, for example, an X Window based browser on the firewall host and then export the window to the display of their workstation. With the browser being hosted on the firewall, users had access to both systems at the same time. However, you should not consider this kind of setup (called "dual homed system") for your own network unless you really trust all the users on it. To understand why this is so, remember that 99% of all break-ins on computer systems start with an attempt to obtain a user account on the targeted system.

The scope of your firewall solution will depend on the required degree of protection, which may need to be in line with legal regulations in some cases, and which should be determined through a communication analysis. SuSE Linux Solutions offers a range of services which can help your company to carry out such a communication analysis.

Another factor affecting the operation of a firewall is the state of its documentation, meaning that there should be a way to determine "who has changed what, when and how", in order to be able to trace back whether changes were made by an authorized party (or not). This will also be quite helpful when it comes to dealing with things like certifications and audits.

The "SuSE Linux Firewall CD" is a product covering the whole range of these issues in all its details, from packet filtering to setting up a multiple-level firewall. And given that the package is based on Open Source programs throughout, it is also possible to audit the source code without too much difficulty.

SuSE Linux Firewall on CD

The “SuSE Linux Firewall on CD” consists of these two basic product components:

1. the SuSE Linux Firewall Live-CD and
2. the SuSE Linux Firewall Admin-CD.

For the sake of simplicity, in this manual we will mostly refer to these as “Live CD” and “Admin CD”. The Live-CD contains the firewall package proper, which in our case is based on the concept of an application level gateway combined with IP packet filtering. The firewall’s routing and gateway capabilities are turned off by default, but can be enabled as required. All requests accepted and processed by the firewall are handled at the application level. The package supports the most important Internet protocols: SMTP, FTP, HTTP, HTTPS and DNS.

With the Admin CD, you are able to install the SuSE Linux Adminhost, which takes care of the configuration, monitoring and administration of the “SuSE Linux Firewall”. To properly manage these tasks, the CD is included with a special selection of software packages and the necessary installation routines.

Connecting an internal company network to the Internet will certainly require a good deal of planning, including a security concept based on a communication analysis and a demand analysis. Furthermore, there should be a concept describing how to deal with emergencies (break-in, data loss etc.). You’ll also need to make sure that the firewall machine is protected from unauthorized physical access. The best way to achieve that is to lock the machine away in a separate server room.

SuSE Linux Solutions AG provides a number of support and consulting services to help you with these planning tasks. In cooperation with your staff, SuSE’ experts can:

- create a security concept,
- perform the installation on your premises,
- training,
- provide documentation and
- maintenace.

SuSE Linux AG can also provide you and your staff with training in its own training center, which offers several training courses specifically on the topic of networks and network security.

2.2 An Overview of Typical Firewall Setups

In this section, we will give a brief overview of the most typical firewall setups. All the configurations presented below can be implemented with the “SuSE Linux Firewall on CD”.

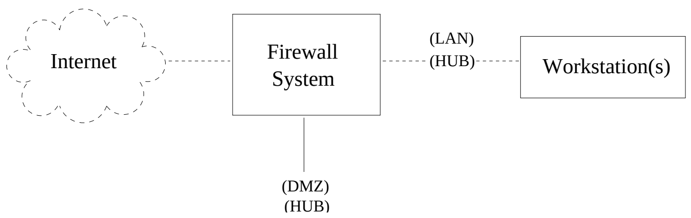


Figure 2.1: Very basic setup

Figure 2.1 shows a firewall with three network interfaces: an external one to connect with the Internet, and two internal ones connecting with the corporate network via LAN/HUB and with the DMZ (demilitarized zone) using another HUB.

With this setup, the firewall has to perform all the functions of the default gateway (router), the packet filter etc. The internal network would be left completely open if the firewall were infiltrated.

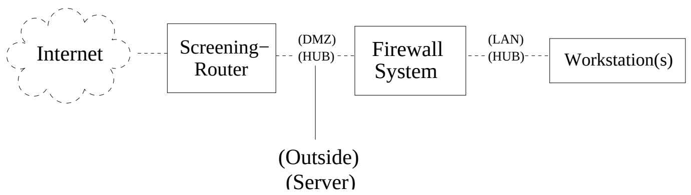


Figure 2.2: Simple setup

The setup shown in figure 2.2 is still a relatively simple one. The DMZ is only protected by a packet filter on the router (screening router).

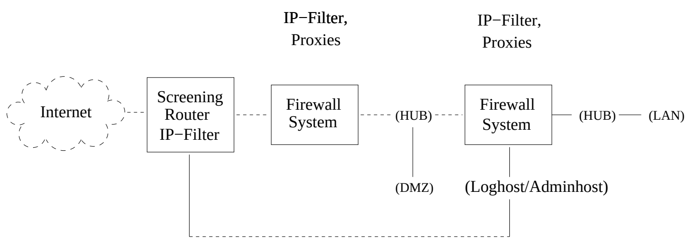


Figure 2.3: Effective but manageable setup

The setup shown in figure 2.3 is still not overly complicated but provides much better protection than the previous ones.

A “screening router” stops any illegal requests at the packet level. Packets which are allowed through are passed on to the first firewall host, which operates at the application level and uses packet filtering rules to control access to the DMZ and to the second firewall host (and thus to the internal network beyond). This second firewall is a packet filtering proxy machine protecting the internal network; in addition to that it controls access from the internal network to the Internet and the DMZ.

3 Adminhost

It is no easy task to administer, maintain and monitor a firewall; above all, monitoring the firewall should not be underestimated. This is why the "SuSE Linux Firewall on CD" includes the "SuSE Linux Firewall Adminhost", which should help you in configuring, administrating and maintaining the SuSE Linux Firewall.

3.1 Introduction

After installing the SuSE Linux Firewall Adminhost, the Firewall Administration System (FAS), a tool with a graphical administration interface, will be available to you, allowing you to carry out a menu-driven configuration of the SuSE Firewall on CD.

Configurable tools are available for monitoring the firewall, which can perform tests of the firewall, evaluation of the log files and monitoring of network traffic. In the case of an attack, there is the possibility of informing system administration by e-mail, pager, etc., so that suitable counter measures may be taken as quickly as possible (e. g. cutting off the network connection to the Internet/Intranet, etc.).

It can already be seen here that a firewall without any monitoring running cannot provide any real protection. Apart from this, all changes that were made to the firewall configuration should always be thoroughly documented. By means of this documentation, errors can be found more easily, or modifications made by unauthorized parties can be changed back again.

After installing the "SuSE Linux Adminhost for Firewall", you will have the choice between a new installation of the host reserved for this purpose or an update provided that SuSE Linux 7.2 is already installed on this host. In the following section, we will describe updating, followed by instructions for the new installation of the "SuSE Linux Adminhost for Firewall".

3.2 Update

If you have already installed SuSE Linux 7.2 on the computer which is to serve as the adminhost for the firewall, you will not need to carry out a new installation, but rather, only install the required packages at a later point. While doing this, you can choose between the graphical installation program YaST2 or the text-based YaST. Proceed in this as follows:

3.2.1 Update with YaST2

1. Log in to the graphical console as user `'root'`.
2. Start the YaST2 control center and select the menu '`Software`' under YaST2 Modules.
3. Insert the CD with the label *SuSE Linux Adminhost for Firewall*.
4. Under '`Change Source Media`', select '`Install from CD`' for the source medium.
5. Save and close the dialog.
6. Select the menu item '`Software`', activate the checkbox '`Show package series`'.
7. Select the series `zfw` and then the packages:
   - fas
   - fas-flc-base
   - fas-flc-dns
   - fas-flc-ftp
   - fas-flc-http
   - fas-flc-ipchains
   - fas-flc-mail
   - fasd
   - yast2-config-fwcdadmin
   - yast2-trans-fwcdadmin

   The package dependencies will be automatically resolved.
8. If you want to utilize this host as a loghost as well, you must install the syslog-ng from the series `n`.

3.2.2 Update with YaST1

1. Log in to a console as user `'root'`.
2. Insert the CD with the labeled *SuSE Linux Adminhost for Firewall*.
3. Start YaST.
4. In the YaST menu, select the item '`Package management`'.
5. Select '`Change/create configuration`'.
6. Switch to series `zfw1`

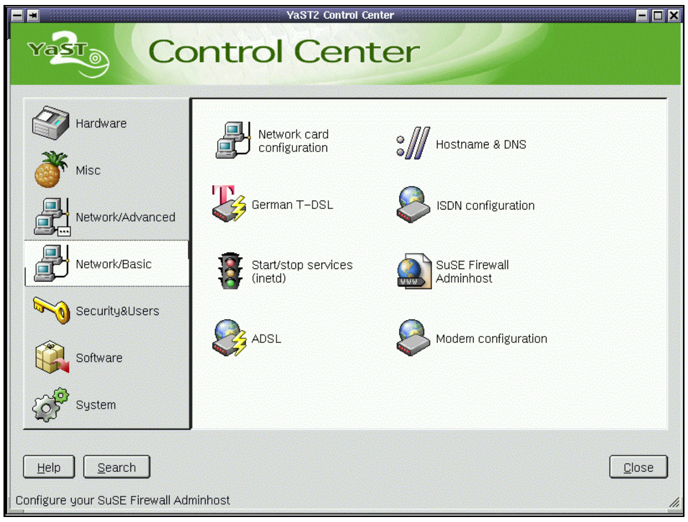
7. Here, select installation of all packages except for package `fas-devel`:
   - fas
   - fas-flc-base
   - fas-flc-dns
   - fas-flc-ftp
   - fas-flc-http
   - fas-flc-ipchains
   - fas-flc-mail
   - fasd
   - yast2-config-fwcdadmin
   - yast2-trans-fwcdadmin
8. The package dependencies will be shown. Resolve them by pressing `F10` inside the '`Unresolved dependencies`' dialog.

3.2.3 Closing Update

Regardless of whether you have installed the packages with YaST1 or YaST2, you will still need to start the YaST2 module for the SuSE Linux Adminhost for Firewall by using the following command:

```
root@adminhost: # /sbin/yast2 fwcdadmin
```

Alternatively, you can start the YaST2 Control Center in KDE2 and under '`Network/Base`', start the "SuSE Firewall Adminhost" module (see Figure 3.1).



Configure your SuSE Firewall Adminhost

Figure 3.1: YaST2 Control Center

Please note that the harden_suse script will be automatically started on the adminhost and that the packet filter will be set up. The file permissions will also be set to `/etc/permissions.secure`. That means that your system will behave completely differently than you are accustomed to following the update.

3.3 New installation of the SuSE Linux Firewall Adminhost

To install the "SuSE Linux Firewall Adminhost" for the first time, please insert the CD with the respective label into the CD drive of the computer and then reboot the machine. If the computer does not boot from the CD, you must change the boot sequence in the BIOS accordingly.

The linuxrc welcome screen will appear - simply press ↵ and the automatic hardware detection will start.

YaST2 will now lead you through the rest of the installation. All the necessary programs for administration, configuration and maintenance will be installed.

3.3.1 Language Selection

Now you will have to make your first decisions in the installation process beginning with the language selection. You can either use the mouse or the keyboard. All entry fields, selection lists and "buttons" or "switches" can be selected by clicking with the mouse. If you want to use the keyboard, the following rules apply:

- `Tab` moves the focus to a field, an entry/selection field or a button; `⇧ Shift` + `Tab` allows you to choose other selection groups. With ↑ and ↓ you can – depending on which area is activated – make a selection or cycle through a list.
- With ↵ the selected command is carried out: as a rule this is the action which is shown on the active button in question.
- With `Space` entries can be marked.
- In addition, most actions can be started with the key combination of `Alt` + *the underlined letter*.

Tip

Here and in the following dialogs, YaST2 is just collecting information. Later YaST2 will display the information it has collected; in Section 3.3.7 on page 24 you still have the chance, by means of the '`Back`' button, to return to the previous dialogs, to correct details.

YaST2 would now like to know what language you prefer. When you have chosen a language, select '`Apply`' to switch all texts to the selected language.

3.3.2 Selecting the Mouse

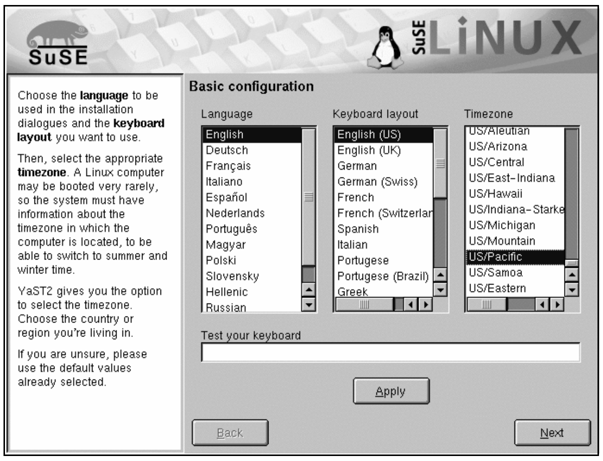
This dialog will only appear if YaST2 was not able to detect the mouse automatically. A dialog window with a long list of mouse types appears, and you will be asked to select the appropriate mouse type.

Once you have found the right mouse type, move with Tab to the 'Test' button and press ⏎. Now move the mouse. If the mouse cursor moves normally, everything is working properly and you can click with the mouse on 'Next'. In case the mouse doesn't work, you can just return to the selection list using the Tab key and change the settings.

If no mouse type functions, or if you don't want to use a mouse, please activate the entry 'No mouse'. The rest of the installation will be carried out using only the keyboard.

3.3.3 Keyboard and Time Zone

Now select the keyboard layout and time zone.



Choose the **language** to be used in the installation dialogues and the **keyboard layout** you want to use.

Then, select the appropriate **timezone**. A Linux computer may be booted very rarely, so the system must have information about the timezone in which the computer is located, to be able to switch to summer and winter time.

YaST2 gives you the option to select the timezone. Choose the country or region you're living in.

If you are unsure, please use the default values already selected.

Basic configuration

Language: English

Keyboard layout: English (US)

Timezone: US/Pacific

Figure 3.2: YaST2: Keyboard layout and time zone

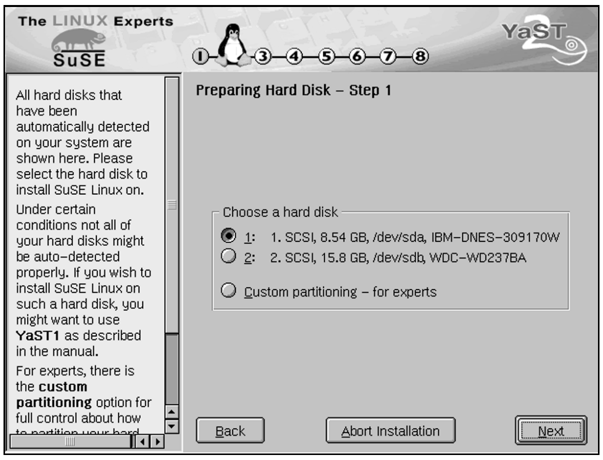
- Now test your keyboard. By clicking with the mouse or using Tab you can activate the entry line and type in letters there. You should especially test 'y'/'z' and special characters.
- The second item is a list of countries in a tree structure (continent/country/region). Select your country or region from these; YaST2 will find the appropriate time zone.

The 'Next' button takes you to the next dialog window.

3.3.4 Preparing the Hard Disk

In the following steps, you will select the hard disk or disks and prepare them for the SuSE Linux installation. – Depending on your computer's hardware, there may be slight differences from the dialogs which appear here.

Step 1



All hard disks that have been automatically detected on your system are shown here. Please select the hard disk to install SuSE Linux on.

Under certain conditions not all of your hard disks might be auto-detected properly. If you wish to install SuSE Linux on such a hard disk, you might want to use **YaST1** as described in the manual.

For experts, there is the **custom partitioning** option for full control about how

Preparing Hard Disk – Step 1

Choose a hard disk

[✓ 1: 1. SCSI, 8.54 GB, /dev/sda, IBM-DNES-309170W]
2: 2. SCSI, 15.8 GB, /dev/sdb, WDC-WD237BA
Custom partitioning – for experts

Figure 3.3: YaST2: Preparing the hard disk (I)

If more than one hard disk exists, you must first decide which one you want to use for the installation. The disks which are found will be listed in turn; see Figure 3.3. Or you can select the last option (‘`Custom Partitioning`’) to “partition” them by hand. In this dialog it is also possible to continue using existing partitions, by specifying formatting and allocating mount points directly.

Normally you will choose *one* hard disk and then click on ‘`Next`’.

Step 2

One of the following situations could occur:

- If the hard disk is *not* empty, YaST2 will show all existing partitions on the hard disk, as well as the item ‘`Use whole hard disk`’. *Free, non-partitioned* storage space at the “end” of the hard disk is also displayed and is automatically pre-selected. YaST2 can use further space for SuSE Linux, but only if it is contiguous, i.e. partitions can only be released for further use “from behind”. For example, if three partitions exist, partitions 1 and 2 will remain and you must check partition 3 to be made available. If you want to make the entire hard disk available for SuSE Linux, select ‘`Entire Hard Disk`’.

- For an *empty* hard disk, the entire hard disk will be used for SuSE Linux.

If you have other requirements, press '`Back`' to return to the previous dialog, as mentioned on on the facing page, for manual partitioning with the help of '`Extended Settings`'.

Note

Since the partitions you have made available for SuSE Linux will be formatted, all existing data there will be irretrievably lost!

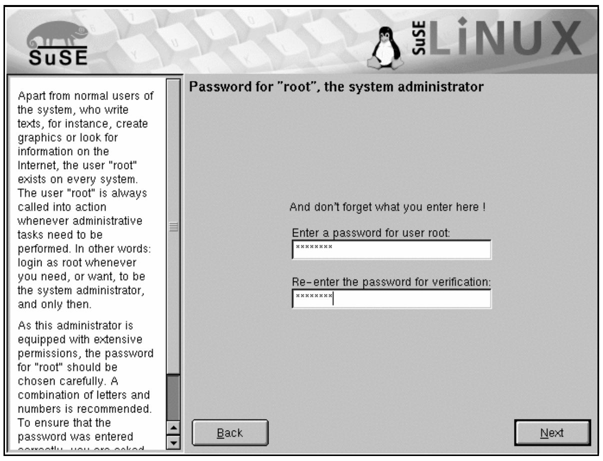
Once the installation starts and all requirements have been fulfilled, YaST2 will partition and format the necessary hard disk space on its own. The entire hard disk, or the available partitions, will then be split up for SuSE Linux into the 3 standard partitions (that is, a small partition for `/boot` (about 16 MB), as close as possible to the beginning of the hard disk, a partition for swap (128 MB) and all the rest for `/` (root partition)).

3.3.5 Boot Manager Configuration

In order for Linux to be bootable after the installation, a boot mechanism must be installed. You will need to specify how the system the boot manager LILO "LInux LOader" is to be installed, or if another boot procedure should be used.

3.3.6 Setting the '`root`' Password

The user '`root`' has special privileges in Linux. He can, for instance, start and stop system processes, create and remove users, manipulate important system files, etc., in other words, perform the duties of the system administrator.



Apart from normal users of the system, who write texts, for instance, create graphics or look for information on the Internet, the user "root" exists on every system. The user "root" is always called into action whenever administrative tasks need to be performed. In other words: login as root whenever you need, or want, to be the system administrator, and only then.

As this administrator is equipped with extensive permissions, the password for "root" should be chosen carefully. A combination of letters and numbers is recommended. To ensure that the password was entered correctly, you are asked

Password for "root", the system administrator

And don't forget what you enter here !

Enter a password for user root: ********

Re-enter the password for verification: ********

Figure 3.4: YaST2: Entering the '`root`' password

To do this, you are requested to provide a password for the user `'root'`; the same rules apply as for the normal user password.

Note

You must remember the `'root'` password very carefully, as you cannot call it up later to have a look at it. You will always need this password whenever you have to perform administrative tasks on the system.

If you press '`Next`', the installation will start.

3.3.7 Confirming Settings – Starting the Installation

In order to give you a chance to check your configuration, you can review all the settings made until now. In case you want to make changes, you can cycle through the windows with the '`Back`' button, all the way back to the first window.

If you press '`Next`', you are asked again for confirmation (in green) to start the installation with the settings as shown:

- After confirming, with '`Yes - install`', YaST2 will begin setting up the system.
- With '`No`', you have the option of checking data again, and changing items where necessary, by pressing '`Back`' to reach the relevant window.

You now still have the option of aborting the installation completely. All settings made and details supplied will be lost. If you select '`Abort installation`', your computer, after asking for confirmation, will shut down, and you can switch off the computer or re-boot without any problem. Up to this point no changes have been made to your computer.

With the option '`Save Settings to Floppy Disk`' you can save all details to floppy disk and use them again for other installations. This can only be selected if it is supported by your hardware, however.

As a rule, you will decide on '`Yes - install`'. Partitions will now be created and formatted. Depending on your system's capacity and size of hard disk, this could take some time. You should avoid aborting here, as this would put the hard disk into an undefined condition.

Afterwards the packages from the CD are read in, and the SuSE Linux base system is installed; after you have confirmed this with '`OK`', the text-oriented base system is started. YaST2 continues the installation of software and – if necessary – requests additional CDs; if you '`Abort`' the installation at this stage, the system will be in an unusable condition!

SuSE Linux is now successfully installed on your computer!

All that is missing is the configuration of the graphical interface; then you can try out SuSE Linux for the first time.

3.3.8 Preparing the Graphical Interface

In order to be able to provide you, even at the first *Login*, with a graphical user interface, YaST2 will now try to find out all the information it needs for the monitor and graphics card.

If this is successful, a sensible screen resolution, color resolution and repetition rate frequency will be selected for the monitor, and a test screen is displayed.

Note
Please check the settings before you give your "Ok". If you are not sure, consult the documents for your graphics card and monitor.

If the monitor is not detected, please select your model from the list provided. If you have an unknown model, you must enter the settings by hand or have the data read from a '`driver disk`', which might have been provided with your monitor; in this case you should consult the documentation for your monitor.

Set up the required screen resolution and a color depth of `16 bpp`. Check the settings by choosing '`Test`' and make fine adjustments if necessary.

Tip
In rare cases it may be necessary for you to have to configure the X server "by hand"; to do this you should call up the program SaX at a later time.

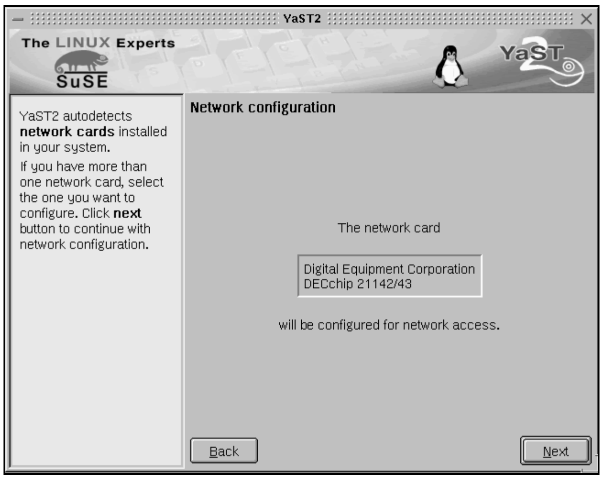
3.3.9 Configuring the Network with YaST2

Have the following data on hand when configuring the network: IP address, network mask, default gateway.

If you are managing a DHCP server, you can also configure the SuSE Linux Firewall Adminhost as a DHCP client. Here, it is to be urgently stressed that you assign a static IP address to the SuSE Linux Firewall Adminhost. You will be provided with a selection of the network cards detected by the hardware recognition. Activate the network card and enter the IP address and network mask. Enter all the required information if a name server already exists, as shown in Figure 3.7 on page 31.

3.3.10 Manual Network Configuration

Manual configuration of the network software should always be the last resort. We recommend that you use YaST, although it cannot cover all aspects of network configuration so that a bit of manual fine-tuning, in some cases, may be necessary.



YaST2 autodetects **network cards** installed in your system.
If you have more than one network card, select the one you want to configure. Click **next** button to continue with network configuration.

The network card

Digital Equipment Corporation
DECchip 21142/43

will be configured for network access.

Figure 3.5: YaST2: Network configuration

Configuration Files

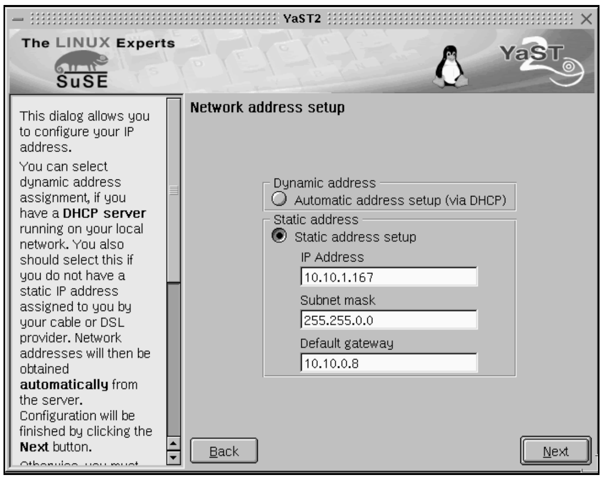
This section provides an overview of the network configuration files and explains their purpose as well as the format used.

`/etc/rc.config`
The majority of the network configuration takes place in this central configuration file. When making changes via YaST or when calling up SuSEconfig after the file has been modified manually, most of the following files will be automatically generated from these entries. Even the boot scripts are configured via these file entries.

Tip
If you modify this file by hand, you will always have to call up SuSEconfig separately afterwards so that the modified configurations are automatically written into the correct files.

`/etc/hosts`
In this file (see file contents 3.3.1 on the facing page), IP addresses are assigned to host names. If no name server is being used, then all hosts which are to receive an IP connection will have to be listed here. A line consisting of the IP address, the fully qualified host name and the host name (e. g. `earth`) is entered into the file for each host. The IP address has to be at the beginning of the line, and entries are separated by blanks or tabs. Comments are always preceeded by the '#' sign.

`/etc/networks`
Here, network names are converted to network addresses. The format is



This dialog allows you to configure your IP address.
You can select dynamic address assignment, if you have a **DHCP server** running on your local network. You also should select this if you do not have a static IP address assigned to you by your cable or DSL provider. Network addresses will then be obtained **automatically** from the server.
Configuration will be finished by clicking the **Next** button.

Network address setup

Dynamic address
- ○ Automatic address setup (via DHCP)

Static address
- ● Static address setup
- IP Address: 10.10.1.167
- Subnet mask: 255.255.0.0
- Default gateway: 10.10.0.8

Figure 3.6: YaST2: Configuring the network address

```
#
# hosts         This file describes a number of host name-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 helios.cosmos.com helios
192.168.0.20 earth.cosmos.com earth
# End of hosts
```

File contents 3.3.1: `/etc/hosts`

similar to that of the `hosts` file, except for the network names preceed the addresses (see the file contents of 3.3.2 on page 30).

`/etc/host.conf`

Name resolution – i. e. the translation of host and network names via the *resolver* library – is controlled by this file. It is only used for programs linked to the libc4 or the libc5; for current glibc programs, refer to the settings in `/etc/nsswitch.conf`! A parameter must always stand alone in its own line and any comments are preceeded by a `'#'` sign. Table 3.1 on the following page shows the parameters available.

order *hosts*, *bind*	Specifies in which order the services are to be accessed for the name resolution. Available arguments are (separated by blank spaces or commas):

Table 3.1: continued overleaf...

	hosts: Searches the `/etc/hosts` file *bind*: Accesses a name server *nis*: Via NIS
multi *on*/*off*	Defines if a host entered in `/etc/hosts` can have multiple IP addresses.
nospoof *on* alert *on*/*off*	These parameters influence the name server *spoofing* but apart from that, do not exert any influence on the network configuration.
trim <domainname>	The specified domain name is separated from the host name following the host name resolution (as long as the host name includes the domain named). This option is useful if only names from the local domain are in the `/etc/hosts` file but are still to be recognized with the attached domain names.

Table 3.1: Parameters for `/etc/host.conf`

An example for `/etc/host.conf` is shown in the file 3.3.3 on page 30.

`/etc/nsswitch.conf`

With the advent of the GNU C Library 2.0, the “Name Service Switch” (NSS) has stepped to the forefront (see manpage manpage for **`nsswitch.conf`** (**`man 5 nsswitch.conf`**), and for more detailed information, see *The GNU C Library Reference Manual*, Chap. "System Databases and Name Service Switch", refer to package `libcinfo`, series `doc`).

The `/etc/nsswitch.conf` file defines the order of data in which it is to be queried. An example of `nsswitch.conf` is shown in the file 3.3.4 on page 30. Comments are preceeded by `‘#’` signs. Here, e.g., the entry under “database” `hosts` means that a request is sent to `/etc/hosts` (`files`) via DNS (see Section A on page 103).

The “databases” available over NSS are listed in Table 3.2 on the facing page. In addition, `automount`, `bootparams`, `netmasks` and `publickey` are to be expected in the near future.

`aliases`	Mail aliases implemented by `sendmail`(8); see also manpage for **`aliases`** (**`man 5 aliases`**).
`ethers`	Ethernet addresses.
`group`	For user groups, used by `getgrent`(3); see also manpage for **`group`** (**`man 5 group`**).
`hosts`	For host names and IP addresses, used by `gethostbyname`(3) and similar functions.

Table 3.2: continued overleaf...

`netgroup`	Valid host and user lists in the network for the purpose of controlling access permissions; see also manpage for **`netgroup`** (**`man 5 netgroup`**).
`networks`	Network names and addresses, used by `getnetent`(3).
`passwd`	User passwords, used by `getpwent`(3); see also manpage for **`passwd`** (**`man 5 passwd`**).
`protocols`	Network protocols, used by `getprotoent`(3); see also manpage for **`protocols`** (**`man 5 protocols`**).
`rpc`	"Remote Procedure Call" names and addresses, used by `getrpcbyname`(3) and similar functions.
`services`	Network services, used by `getservent`(3).
`shadow`	"Shadow" user passwords, used by `getspnam`(3); see also manpage for **`shadow`** (**`man 5 shadow`**).

Table 3.2: Available "databases" via `/etc/nsswitch.conf`

The configuration options for NSS "databases" are listed in Table 3.3.

`files`	directly access files, for example, to `/etc/aliases`.
`db`	access via a database.
`nis`	NIS;
`nisplus`	
`dns`	Only usable by `hosts` and `networks` as an extension.
`compat`	Only usable by `passwd`, `shadow` and `group` as an extension.
also	it is possible to trigger various reactions with certain lookup results; details can be found in the manpage for **`nsswitch.conf`** (**`man 5 nsswitch.conf`**).

Table 3.3: Configuration options for NSS "Databases"

`/etc/nscd.conf`

The nscd (Name Service Cache Daemon) is configured in this file (see manpage for **`nscd`** (**`man 8 nscd`**) and manpage for **`nscd.conf`** (**`man 5 nscd.conf`**)). This affects the data resulting from `passwd`, `groups` and `hosts`. The daemon will have to be restarted once the name resolution (DNS) has been altered by a modification in the `/etc/resolv.conf` file. The following command serves this function:

```
earth: # rcnscd restart
```

```
#
# 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.
```

File contents 3.3.2: `/etc/networks`

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

File contents 3.3.3: `/etc/host.conf`

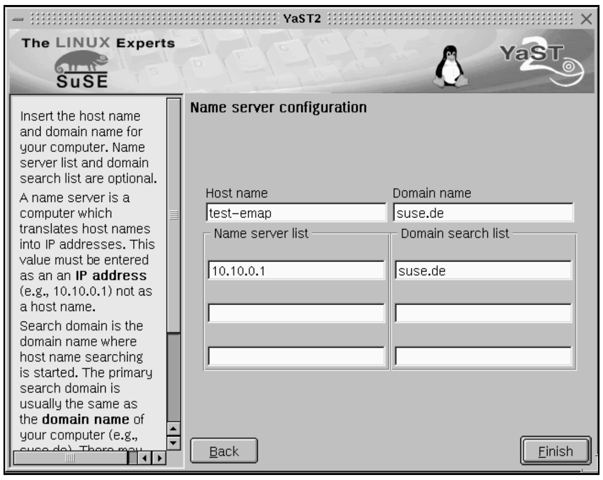
```
#
# /etc/nsswitch.conf
#
passwd:     compat
group:      compat

hosts:      files dns
networks:   files dns

services:   db files
protocols:  db files

netgroup:   files
```

File contents 3.3.4: `/etc/nsswitch.conf`



Insert the host name and domain name for your computer. Name server list and domain search list are optional.

A name server is a computer which translates host names into IP addresses. This value must be entered as an an **IP address** (e.g., 10.10.0.1) not as a host name.

Search domain is the domain name where host name searching is started. The primary search domain is usually the same as the **domain name** of your computer (e.g.,

Name server configuration

Host name: test−emap
Domain name: suse.de

Name server list: 10.10.0.1
Domain search list: suse.de

Figure 3.7: YaST2: Configuring the name server

Caution

If, for example, the caching for `passwd` is activated, it will usually take about 15 seconds until a newly added user is recognized by the system. By restarting nscd, you can reduce this waiting period.

`/etc/resolv.conf`

As is already the case with the `/etc/host.conf` file, this file likewise plays a role in host name resolution, by way of the *resolver* library.

The domain to which the host belongs is specified in this file (keyword **`search`**) and the status of the name server address (keyword **`name server`**) which is to be accessed. Multiple domain names can be specified. When resolving a name that is not fully qualified, an attempt is made to generate one by attaching the individual **`search`** entries. Multiple name servers can be made known by entering several lines, each beginning with **`name server`**. Once again, comments are preceeded by `'#'` signs.

An example of `/etc/resolv.conf` is shown in the file contents 3.3.5 on the following page.

YaST enters the given name server here!

`/etc/HOSTNAME`

Here is the host name without the domain name attached. This file is read by several scripts while the machine is booting. It may only contain one line where the host name is mentioned! This file will also be automatically generated from the settings made in `/etc/rc.config`.

```
# /etc/resolv.conf
#
# Our domain
search cosmos.com
#
# We use helios (192.168.0.1) as name server
name server 192.168.0.1
# End of resolv.conf
```

File contents 3.3.5: `/etc/resolv.conf`

Start-Up Scripts

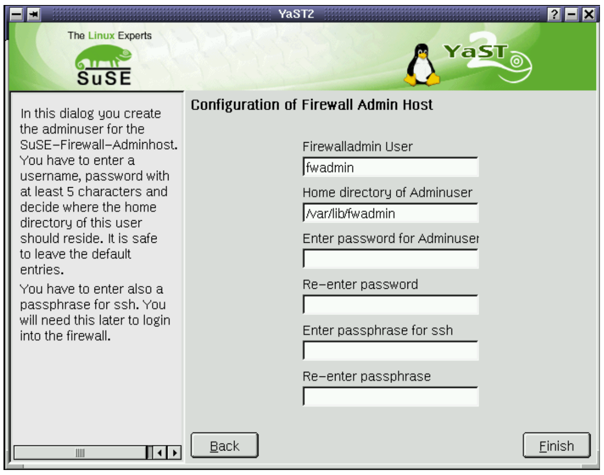
Apart from the configuration files described above, there are also various scripts which load the network programs while the machine is being booted. These are started as soon as the system is switched to one of the *multi-user runlevels* (see also Table 3.4).

`/etc/init.d/network`	This script takes over the configuration for the network hardware and software during the system's start-up phase. During this process, the IP and network address, netmask and gateway specifications entered by YaST in the `/etc/rc.config` file are evaluated.
`/etc/init.d/route`	Serves the purpose of setting static routes over the network.
`/etc/init.d/inetd`	Starts inetd, as long as it is specified in `/etc/rc.config`. This is only necessary if you want to log in to this machine from the network.
`/etc/init.d/portmap`	Starts the portmapper needed for the RPC server such as, for example, an NFS server.
`/etc/init.d/nfsserver`	Starts the NFS server.
`/etc/init.d/sendmail`	Controls the sendmail process depending on the settings in `/etc/rc.config`.
`/etc/init.d/ypserv`	Starts the NIS server depending on the settings in `/etc/rc.config`.
`/etc/init.d/ypbind`	Starts the NIS client depending on the settings in `/etc/rc.config`.

Table 3.4: Some of the start-up scripts for the network programs

Please note that the host name will only be written to the system following a reboot. Some applications will only function with a definitive host name.

3.3.11 The user ‘`fwadmin`’ for the Firewall Administration System (FAS)



In this dialog you create the adminuser for the SuSE–Firewall–Adminhost. You have to enter a username, password with at least 5 characters and decide where the home directory of this user should reside. It is safe to leave the default entries.
You have to enter also a passphrase for ssh. You will need this later to login into the firewall.

Configuration of Firewall Admin Host

Firewalladmin User: fwadmin
Home directory of Adminuser: /var/lib/fwadmin
Enter password for Adminuser:
Re–enter password:
Enter passphrase for ssh:
Re–enter passphrase:

Figure 3.8: The user for the Firewall Administration System (FAS)

In the ‘`fwcdadmin`’ configuration mask (see Figure 3.8), you can set up a user to with which you can carry out the configuration of the SuSE Firewall CD. Enter a user name – the default name is ‘`fwadmin`’. You can also specify a home directory for the user. The default for this is `/var/lib/fwadmin`. In this configuration mask, you should assign a password for the firewall admin user. This password must be at least five characters long. In addition you must define a “pass phrase” and repeat this in the next field to confirm it.

Selecting a good, secure password will finally conclude the basic installation of the SuSE Firewall Adminhost.

3.4 Firewall Administration System (FAS)

Following installation, the system will boot to the graphical login. Log in here as user `‘fwadmin’` with your password. In all cases, FAS will only function once the system has been rebooted.

3.4.1 Logging in as fwadmin

You will arrive at the user desktop for `‘fwadmin’` where you will find an icon which starts the administrations desktop for direct configuration of the firewall CD just by clicking on it (see Figure 3.9).

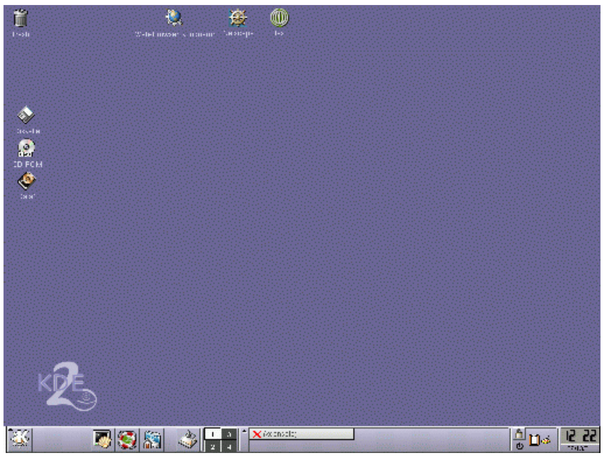


Figure 3.9: Desktop with FAS icon

3.4.2 Starting the Firewall Administration System

FAS is the graphical administration interface for creating the configuration disk for the “SuSE Firewall CD”. FAS supports multiple users and is able to manage several configurations.

FAS is a client/server system made up of the GUI and the fasd server daemon. The fasd (fas daemon) manages the different configurations, makes changes, and checks the entries for accuracy. The front-end receives the user data and forwards it to the server. In this version, communication between the front- and back-ends takes place over Unix domain sockets.

There are several ways to start the Firewall Administration System:

1. Icon on desktop (see Figure 3.9)
2. Menu selection (see Figure 3.10 on the next page)
3. Enter the command **FAS** at the command line (e.g. xterm).

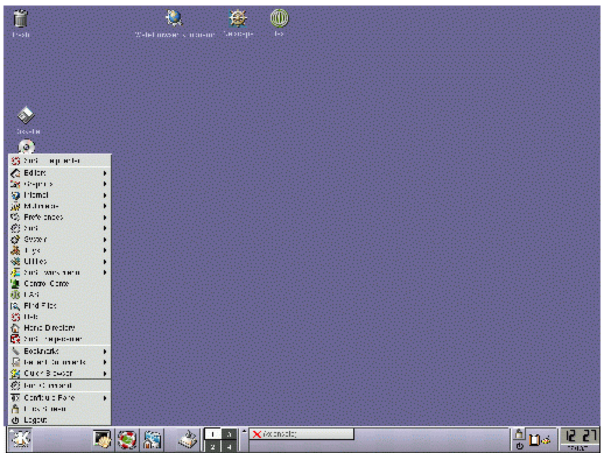


Figure 3.10: K menu with menu item FAS

3.4.3 Creating a new firewall configuration

Once you have started the configuration program FAS, you will have to create a new user account in FAS. To do this, select the item ‘`Create Login`’ in the ‘`File`’ menu. A dialog will emerge where you can enter a new username and password. The username must be at least five characters long. The password must be between five and eight characters long. This password protects the configuration of the “SuSE Firewall on CD”, so the password will be authenticated by the program cracklib.

A rule of thumb: good passwords should not be able to be guessed by others, so please do not use your own birth date, street address or the name of your favorite celebrity etc. They should not be too short but still should be able to be typed quickly so that no one can see what you are typing. Use a mixture of numbers as well as upper- and lower-case letters. Of course, in any case, it is important that you can also remember your password and not have to write it down. Tested-and-true methods of coming up with a password are therefore abbreviations of sentences or expressions which are easy to memorize. Here are some examples:

- NE14TenS (anyone for tennis?)
- AuaEGC (all UNIX-admins eat green cheese)
- Me1st (Me first)
- IwIhagp (I wish I had a good password)

The configuration of the “SuSE Firewall on CD” is saved to a `.tar.gz` archive. To be able to edit a configuration already created, you will have to give this password again.

After logging in, select ‘`New Config`’ under the menu item ‘`Configuration`’. A new window will appear where you can define a name for the configuration. In addition, you will have to enter a description of the configuration. This description can also be used to explain the purpose of the configuration and to document who when and what has changed or has been changed in the configuration. The description of the configuration can be added to later or changed at any time. Use this option, since good documentation is essential.

Confirm with '`OK`'.

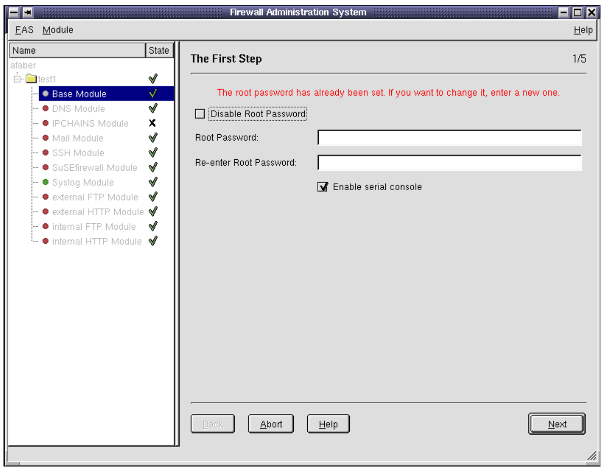
You will now see the name of your new configuration in the left window panel. If you click on the name once, the configuration description will appear to the right. Now you can edit your description. To save changes to your configuration description, click on '`Save the changes`'. Double-click on the configuration name or on the button '`Open Configuration`' to begin creating your configuration.

In the left window panel, a list will appear with the configuration modules, i.e. with the services which you can also use to configure your firewall along with the status of each module. The first thing you need to do is configure the base module ('`Base module`' and '`Syslog Module`'). As long as these modules are in "not configured" status (symbolized by a cross), you cannot choose any of the other modules.

Configuration of the base module

The base configuration takes place in three steps:

1. '`root`' password for the firewall:



Name	State
afaber	
test1	✓
Base Module	✓
DNS Module	✓
IPCHAINS Module	✗
Mail Module	✓
SSH Module	✓
SuSEfirewall Module	✓
Syslog Module	✓
external FTP Module	✓
external HTTP Module	✓
internal FTP Module	✓
internal HTTP Module	✓

The First Step 1/5

The root password has already been set. If you want to change it, enter a new one.

☐ Disable Root Password

Root Password:

Re-enter Root Password:

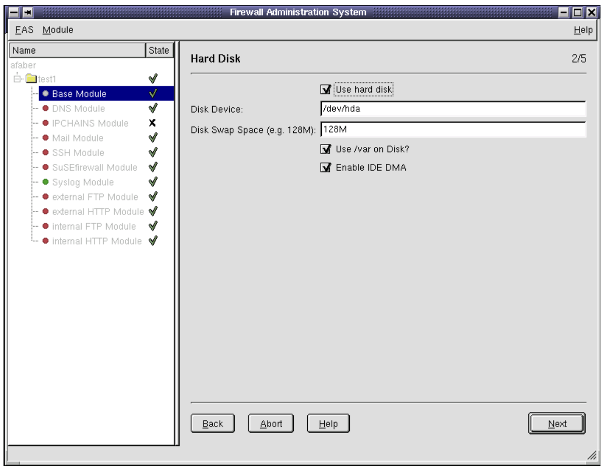
☑ Enable serial console

Figure 3.11: Define root password

In this dialog (see Figure 3.11), the '`root`' password is set for the firewall. If you do not give a '`root`' password (deactivate checkbox), nobody will be able to directly log in to the firewall host as '`root`' during future operation. Access is only possible via ssh and RSA key provided that you have configured ssh.

The checkbox '`enable serial console`' activates the connection of a serial console which can be used to control the firewall.

2. Setup of the (optional) hard disk (see Figure 3.12):



Name	State
afaber	
test1	✓
Base Module	✓
DNS Module	✓
IPCHAINS Module	✗
Mail Module	✓
SSH Module	✓
SuSEfirewall Module	✓
Syslog Module	✓
external FTP Module	✓
external HTTP Module	✓
internal FTP Module	✓
internal HTTP Module	✓

Hard Disk 2/5

[✓] Use hard disk
Disk Device: /dev/hda
Disk Swap Space (e.g. 128M): 128M
[✓] Use /var on Disk?
[✓] Enable IDE DMA

Figure 3.12: Hard disk setup

If you activate '`Use hard disk`', you can make the following settings:

Disk Device: States the hard disk to be integrated: e. g.: `/dev/hda` (first hard disk on the first IDE controller)

Disk Swap Space: Size of the Swap partition: e. g. 128 MB or 64 MB

Checkbox use /var on Disk: Should the directory `/var` be located on the hard disk device?

Enable IDE DMA: activates DMA for IDE

If you use the e-mail proxy and/or enable the caching for the HTTP proxy and/or you want to save log messages to the hard disk, the hard disk must be configured and /var activated.

3. Network interfaces (see Figure 3.13 on the following page):

In this dialog, you will find a list with network interfaces already created. Up to ten interfaces can be given, whereby you must enter a minimum of one internal and one external.

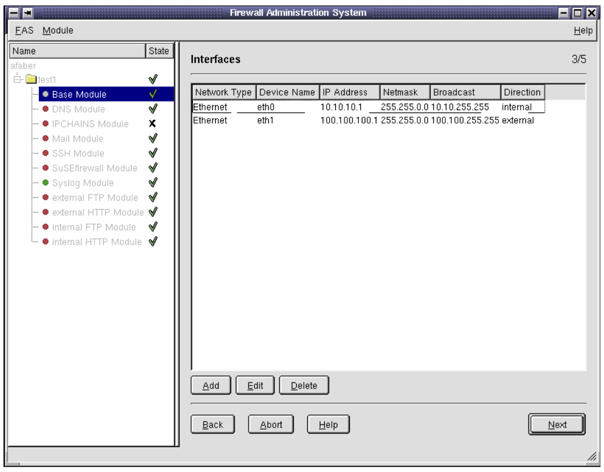
'`New`' adds a new network interface. A window will appear with the '`Interface Dialog`'. The following data is required:

Network Type: ethernet is the "default" (this version only supports ethernet).

Device Name: to be given in consecutive order

IP Address: IP addresses to be assigned to the interface

Netmask: the netmask of the IP address

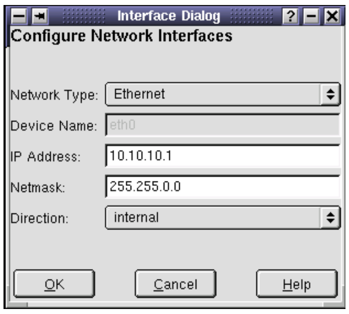


Interfaces 3/5

Name	State
afaber	
test1	✓
Base Module	✓
DNS Module	✓
IPCHAINS Module	✗
Mail Module	✓
SSH Module	✓
SuSEfirewall Module	✓
Syslog Module	✓
external FTP Module	✓
external HTTP Module	✓
internal FTP Module	✓
internal HTTP Module	✓

Network Type	Device Name	IP Address	Netmask	Broadcast	Direction
Ethernet	eth0	10.10.10.1	255.255.0.0	10.10.255.255	internal
Ethernet	eth1	100.100.100.1	255.255.0.0	100.100.255.255	external

Figure 3.13: Network interfaces



Configure Network Interfaces

Network Type: Ethernet
Device Name: eth0
IP Address: 10.10.10.1
Netmask: 255.255.0.0
Direction: internal

Figure 3.14: Adding a new network interface

Purpose: internal/external; is the interface connected to the Intranet, to the DMZ or to the Internet?

'`OK`' confirms the settings, '`Cancel`' aborts the dialog. The newly configured interface will now appear in the list. '`Edit`' reconfigures an already existing interface. Select an interface from the list and choose '`Edit`' to make changes. '`Delete`' deletes the selected interface.

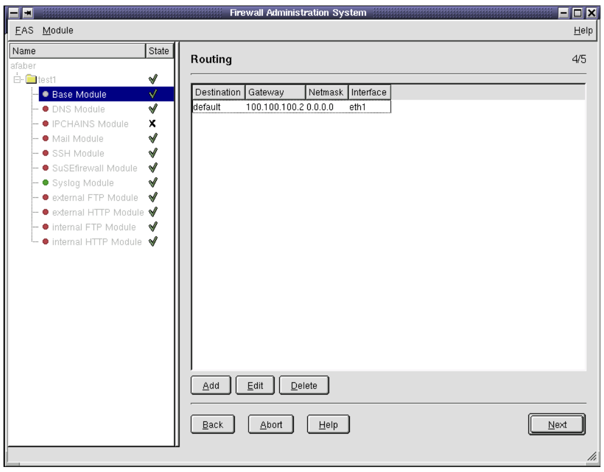
4. Routing (see Figure 3.15 on the next page)

   Here, you can set specific routes. '`New`' creates a new route, '`Edit`' edits an already existing one.

   In the dialog box '`Configure Routing`' (see Figure 3.16 on the facing page), the following settings can be made:

   Destination: Which network or which host should be the routing destination? Enter the network address (e. g.`192.168.0.0`).

   Gateway: If a gateway is necessary, which IP address does it have?

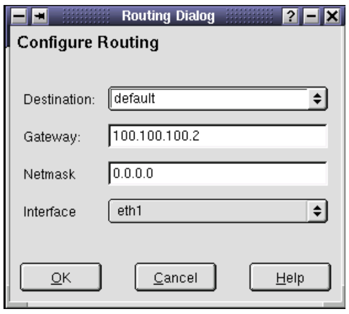


Name	State
afaber	
test1	✓
Base Module	✓
DNS Module	✓
IPCHAINS Module	✗
Mail Module	✓
SSH Module	✓
SuSEfirewall Module	✓
Syslog Module	✓
external FTP Module	✓
external HTTP Module	✓
internal FTP Module	✓
internal HTTP Module	✓

Routing 4/5

Destination	Gateway	Netmask	Interface
default	100.100.100.2	0.0.0.0	eth1

Figure 3.15: Routing configuration



Configure Routing

Destination: default
Gateway: 100.100.100.2
Netmask 0.0.0.0
Interface eth1

Figure 3.16: Routing

Genmask: The corresponding network mask.

Interface: Over which interface should the route run?

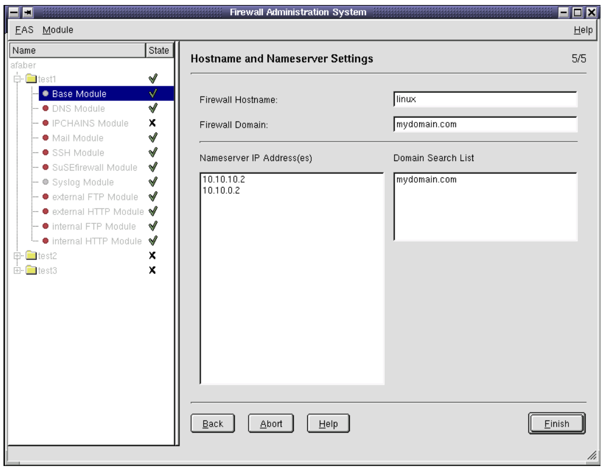
Save the settings with '`OK`' as you did before and abort the configuration with '`Cancel`'. With '`Delete`', you can delete an existing route.

5. Host and domain configuration (see Figure 3.17 on the next page):

Firewall Hostname: Specify the name of the firewall host. Avoid names such as gateway or firewall.

Firewall Domain: Specify the name of the domain where the firewall is.

Nameserver: For proper configuration of the resolver, specify the name server and the search lists, e. g.`your-company-inc.com`.



Hostname and Nameserver Settings 5/5

Firewall Hostname: linux

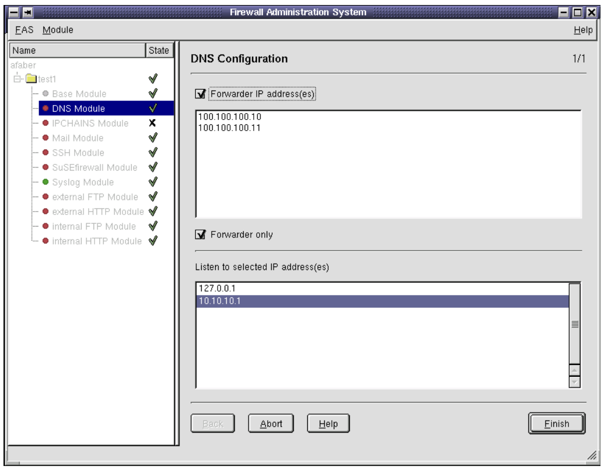
Firewall Domain: mydomain.com

Nameserver IP Address(es)	Domain Search List
10.10.10.2	mydomain.com
10.10.0.2	

Figure 3.17: Host and domain configuration

DNS configuration

In this menu, the name server bind8 will be configured. Detailed descriptions of DNS and bind8 can be found in the appendix of this manual.



DNS Configuration 1/1

[✓] Forwarder IP address(es)

100.100.100.10
100.100.100.11

[✓] Forwarder only

Listen to selected IP address(es)

127.0.0.1
10.10.10.1

Figure 3.18: DNS configuration

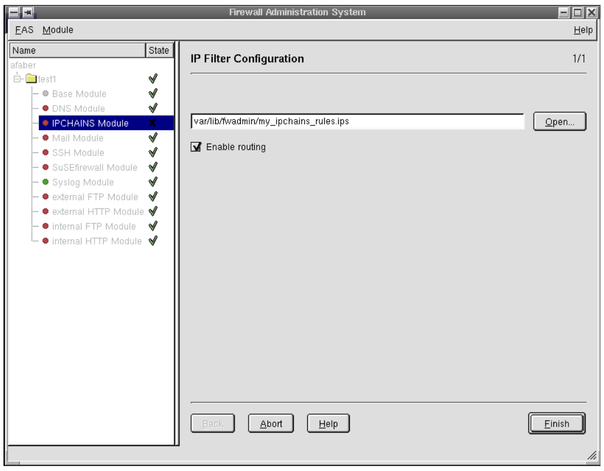
Forwarder IP Addresses: Here, specify whether you want to forward DNS requests to one or more different name servers. To do this, enable the checkbox '`DNS Forwarder IP Addresses`'. This will activate the ListBox. Now you can specify a list of name servers to which DNS requests should be forwarded. If you activate the checkbox '`forward only`', the name server requests will only be forwarded to the name server you state. If it is disabled, the other so-called “root” name servers will be contacted (commonly recognized name servers in the Internet) (see Figure 3.18 on the facing page).

Listen to selected IP Address(es): Normally, the bind8 name server accepts requests from all available network interfaces, that is, from the internal as well as from the external network interfaces, and the loopback interface. In the ListBox, select the interfaces on which DNS requests can be accepted.

IP filter configuration with ipchains

In the FAS-ipchains module (see Figure 3.19), specify a file to which you saved a packet filter configuration using **`ipchains-save`** (see also **`man ipchains(8)`**). This file will be read and subsequently written to the configuration disk.

This configuration can be adopted in this manner by the firewall.



IP Filter Configuration 1/1

var/lib/fwadmin/my_ipchains_rules.ips

[✓] Enable routing

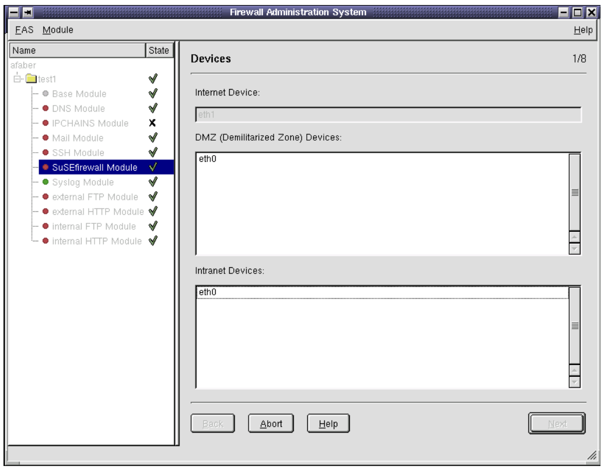
Figure 3.19: IP packet filter configuration with ipchains

Configuration of the SuSE firewall module

This module allows you to configure the SuSE Firewall Script step by step.

1. First, specify which of the network interfaces you configured in the base module is to be designated for the DMZ (Demilitarized Zone) or for your internal network. The interfaces you specified as internal in the base configuration will appear here. Select from the list by clicking with the mouse (see Figure 3.20 on the following page). The interfaces you specified as external in the base module will automatically show up in the text field “Internet Device”.

2. Routing filter: here, you can enable direct routing from IP packets by specifying a source and destination address a well as a destination port (see Figure 3.21 on the next page). You will need this for your services which do not have proxies.



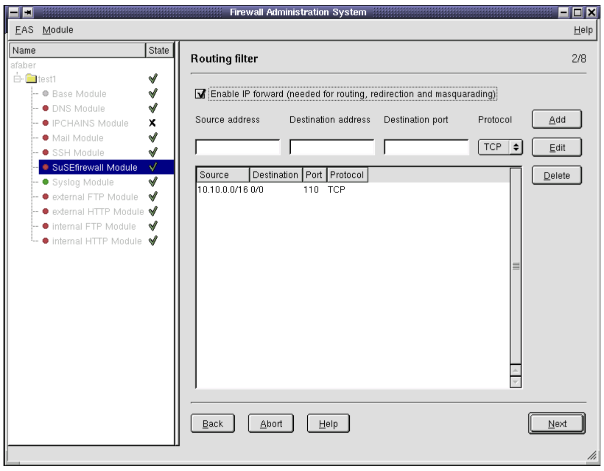
Devices 1/8

Internet Device: eth1

DMZ (Demilitarized Zone) Devices: eth0

Intranet Devices: eth0

Figure 3.20: Selecting devices



Routing filter 2/8

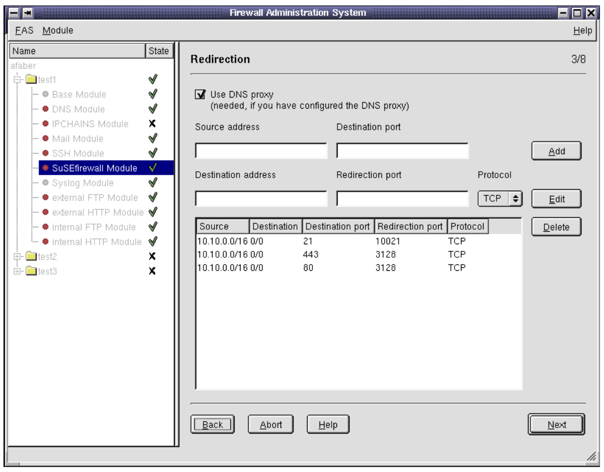
[✓] Enable IP forward (needed for routing, redirection and masquarading)

Source address | Destination address | Destination port | Protocol: TCP

Source	Destination	Port	Protocol
10.10.0.0/16	0/0	110	TCP

Figure 3.21: Routing filter

3. Rerouting: To enable transparent “proxying”, you will need to reroute the packets. For example, if you want to carry out transparent configurations of the internal HTTP proxy. The IP packets have an internal IP address as source address and as destination address, an Internet IP address. These IP packets have port 80 (www) as the destination port. Your HTTP proxy, however, waits on port 3128 for the packets on your firewall. This rerouting procedure, from a destination address with destination port 80 to the required localhost port 3128 can be configured in this screen (see Figure 3.22).

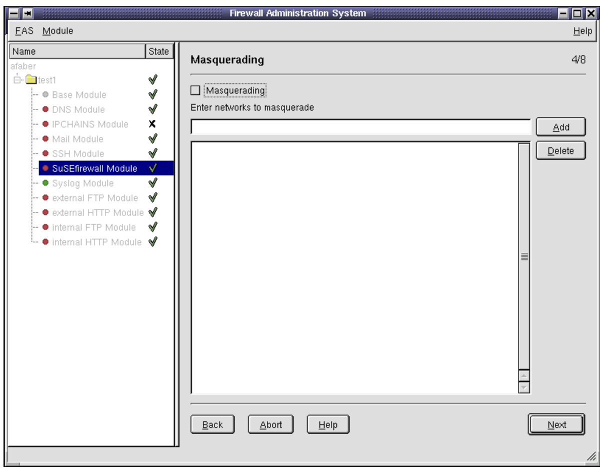


☑ Use DNS proxy
(needed, if you have configured the DNS proxy)

Source	Destination	Destination port	Redirection port	Protocol
10.10.0.0/16	0/0	21	10021	TCP
10.10.0.0/16	0/0	443	3128	TCP
10.10.0.0/16	0/0	80	3128	TCP

Figure 3.22: Packet rerouting

4. If you want to implement masquerading, activate the checkbox ‘`Masquerading`’, otherwise, simply click on next (see Figure 3.23).



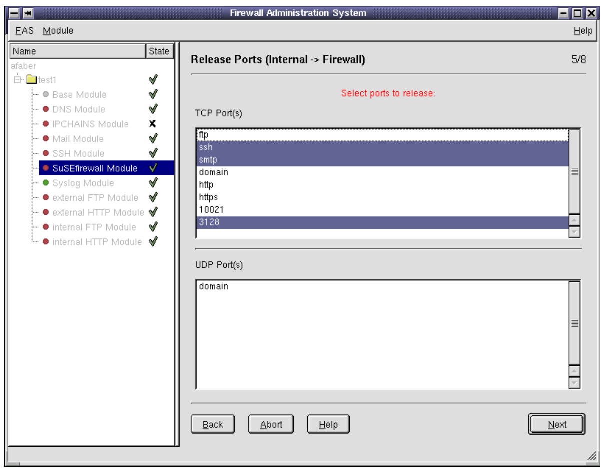
☐ Masquerading
Enter networks to masquerade

Figure 3.23: Masquerading

In this window, you can specify which internal networks are to be masked,

e. g. `192.168.10.0/24`

5. Release of ports for internal access to the firewall. Which firewall services do you want to make available for the internal network? Select from the list provided (see Figure 3.24).



afaber
test1 ✓
Base Module ✓
DNS Module ✓
IPCHAINS Module ✗
Mail Module ✓
SSH Module ✓
SuSEfirewall Module ✓
Syslog Module ✓
external FTP Module ✓
external HTTP Module ✓
internal FTP Module ✓
internal HTTP Module ✓

Release Ports (Internal -> Firewall) 5/8

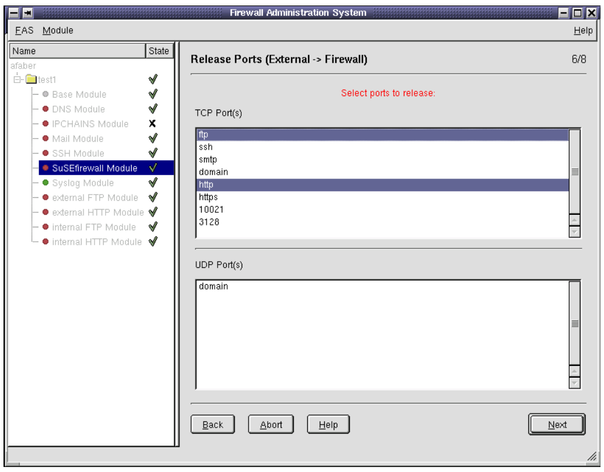
Select ports to release:

TCP Port(s)
ftp
ssh
smtp
domain
http
https
10021
3128

UDP Port(s)
domain

Figure 3.24: Releasing internal ports to the firewall

6. Release of external ports to the firewall: Select from the list of ports you want to release for this purpose, i.e. which can be accessed from the Internet (see Figure 3.25).



afaber
test1 ✓
Base Module ✓
DNS Module ✓
IPCHAINS Module ✗
Mail Module ✓
SSH Module ✓
SuSEfirewall Module ✓
Syslog Module ✓
external FTP Module ✓
external HTTP Module ✓
internal FTP Module ✓
internal HTTP Module ✓

Release Ports (External -> Firewall) 6/8

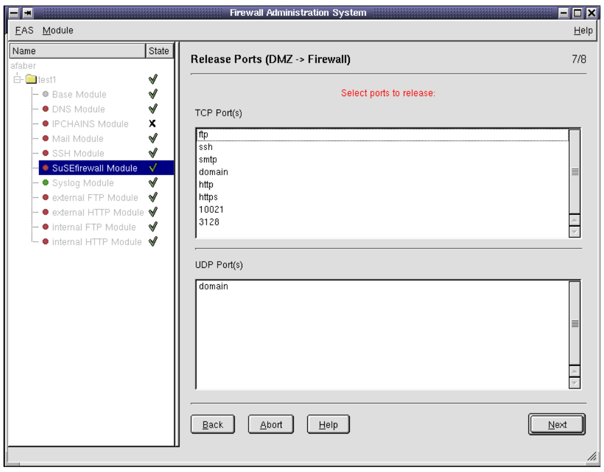
Select ports to release:

TCP Port(s)
ftp
ssh
smtp
domain
http
https
10021
3128

UDP Port(s)
domain

Figure 3.25: Releasing external ports to the firewall

7. Release the ports which are to access the firewall from the DMZ by selecting, once again, from the list provided (see Figure 3.26 on the facing page). If you did not choose any DMZ interface (if in 1.), this screen will be disabled.



Firewall Administration System

Name	State
afaber	
test1	✓
Base Module	✓
DNS Module	✓
IPCHAINS Module	✗
Mail Module	✓
SSH Module	✓
SuSEfirewall Module	✓
Syslog Module	✓
external FTP Module	✓
external HTTP Module	✓
internal FTP Module	✓
internal HTTP Module	✓

Release Ports (DMZ -> Firewall) 7/8

Select ports to release:

TCP Port(s)

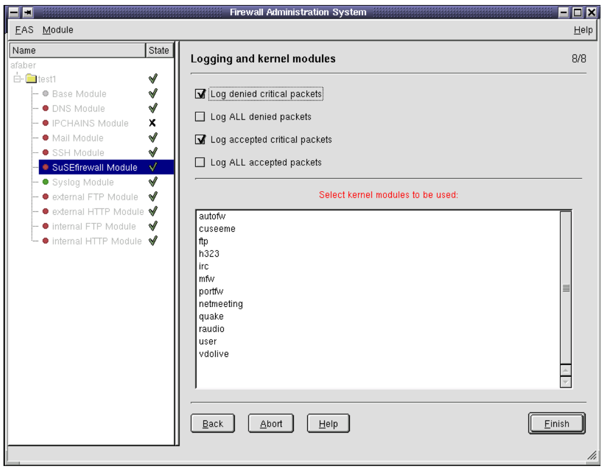
```
ftp
ssh
smtp
domain
http
https
10021
3128
```

UDP Port(s)

```
domain
```

Figure 3.26: Releasing ports from the DMZ to the firewall

8. Protocol and kernel modules.



Firewall Administration System

Name	State
afaber	
test1	✓
Base Module	✓
DNS Module	✓
IPCHAINS Module	✗
Mail Module	✓
SSH Module	✓
SuSEfirewall Module	✓
Syslog Module	✓
external FTP Module	✓
external HTTP Module	✓
internal FTP Module	✓
internal HTTP Module	✓

Logging and kernel modules 8/8

- [✓] Log denied critical packets
- [ ] Log ALL denied packets
- [✓] Log accepted critical packets
- [ ] Log ALL accepted packets

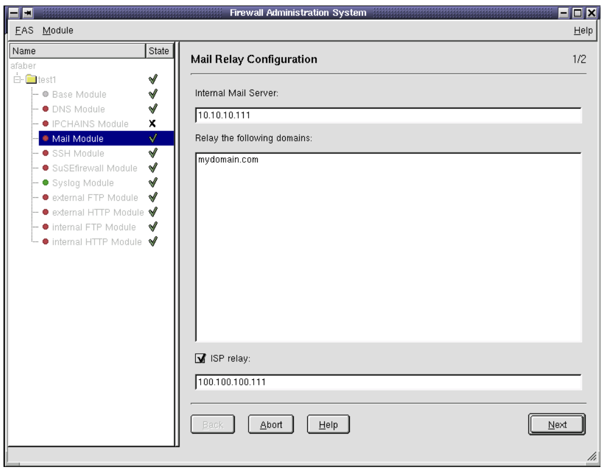
Select kernel modules to be used:

```
autofw
cuseeme
ftp
h323
irc
mfw
portfw
netmeeting
quake
raudio
user
vdolive
```

Figure 3.27: Selection of protocol and kernel modules

- Select the “log level” for the packet filters from the series of checkboxes. In doing so, please be aware that the log files can grow very quickly if you are logging all the packets together.
- In the selection field pertaining to the kernel modules, choose the modules you will need if you are using certain applications which operate independently of the firewall (see Figure 3.27).

Configuring the mail relay (Postfix)



Mail Relay Configuration 1/2

Internal Mail Server: 10.10.10.111

Relay the following domains: mydomain.com

ISP relay: ✓ 100.100.100.111

Figure 3.28: Configuration of the mail relays – Dialog 1

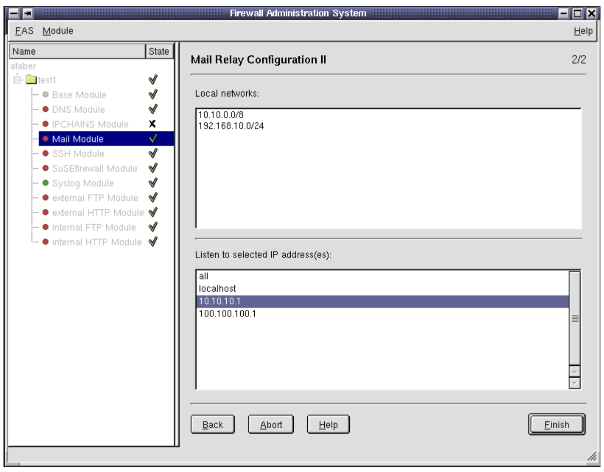
Internal mail server: Specify the IP address or the name of your internal mail server (see Figure 3.28).

Checkbox ISP relay: Activate this checkbox if you forward all outgoing e-mail to the SMTP server of your Internet Service Provider (ISP). Specify the name (FQHN) or the IP address of the SMTP server of your provider.

Relay the following domains: Can be used to specify the domain names to receive any relayed mail and to forward mail to your mail server.

Local networks: List of your networks. Postfix decides by means of this list if a network can send e-mails over the firewall. Enter your networks here (e. g. `192.168.0.0/24`) (see Figure 3.29).

Listen to IP: Here, enter the IP addresses where the mail relay receives requests.



Mail Relay Configuration II 2/2

Local networks:

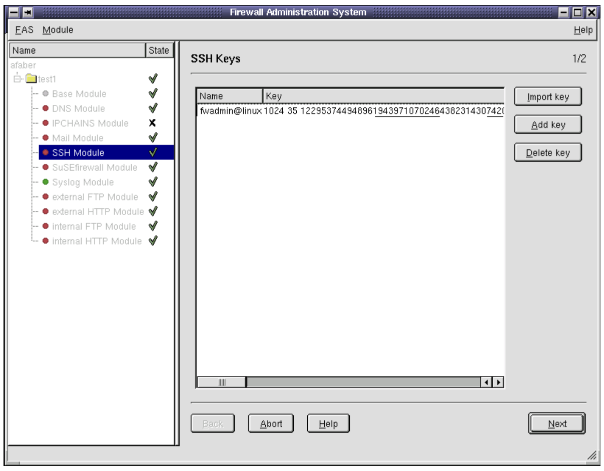
```
10.10.0.0/8
192.168.10.0/24
```

Listen to selected IP address(es):

```
all
localhost
10.10.10.1
100.100.100.1
```

Figure 3.29: Configuration of the mail relays – Dialog 2

SSH configuration:



Name	State
afaber	
test1	✓
Base Module	✓
DNS Module	✓
IPCHAINS Module	✗
Mail Module	✓
SSH Module	✓
SuSEfirewall Module	✓
Syslog Module	✓
external FTP Module	✓
external HTTP Module	✓
internal FTP Module	✓
internal HTTP Module	✓

Name	Key
fwadmin@linux	1024 35 122953744948961943971070246438231430742(

Figure 3.30: Adding and deleting SSH keys

SSH keys: Here, you have the option of saving your "ssh public key" for accessing the firewall as `'root'` so that you can, if necessary, log on to the firewall host. You have two options here (see Figure 3.30):

- You can import a key. An "ssh key" normally is located in the user's home directory in the `.ssh/` directory as `identity.pub`. Select this file. The key will then appear in the list (see Figure 3.31).

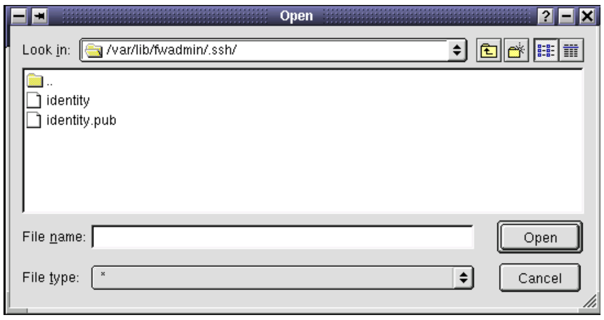


Figure 3.31: Importing the key

- You can also enter a key by way of "copy&paste". If you click on '`add key`', a dialog will open up (see Figure 3.32 on the facing page). In the text field, you can specify your "ssh public key" and confirm with '`OK`'. This key will now appear in the list below.

Delete key: Select the key to be deleted. Click on '`delete key`'. The selected key will be deleted.

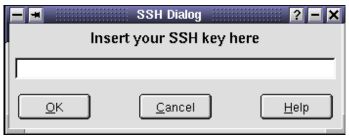
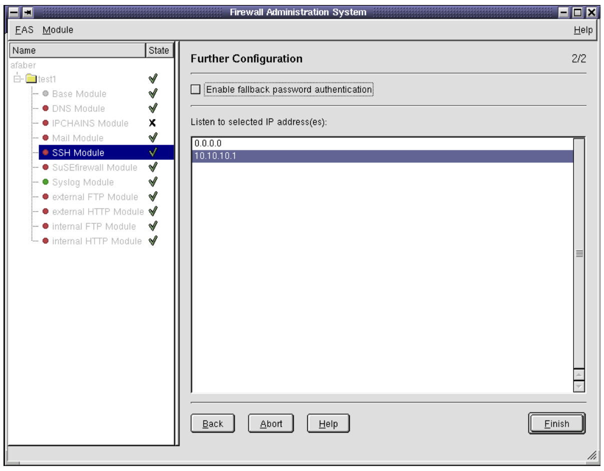


Figure 3.32: Adding the key

Checkbox password authentication: If this checkbox (see Figure 3.33) is activated, you would be able to log on to the firewall using your password. Do not do this! We advise you to only allow access via RSA keys.

Listen to selected IP address: On which network interface should ssh accept requests? Enter the IP addresses in the list field.



Name	State
afaber	
test1	✓
Base Module	✓
DNS Module	✓
IPCHAINS Module	✗
Mail Module	✓
SSH Module	✓
SuSEfirewall Module	✓
Syslog Module	✓
external FTP Module	✓
external HTTP Module	✓
internal FTP Module	✓
internal HTTP Module	✓

Further Configuration 2/2

☐ Enable fallback password authentication

Listen to selected IP address(es):

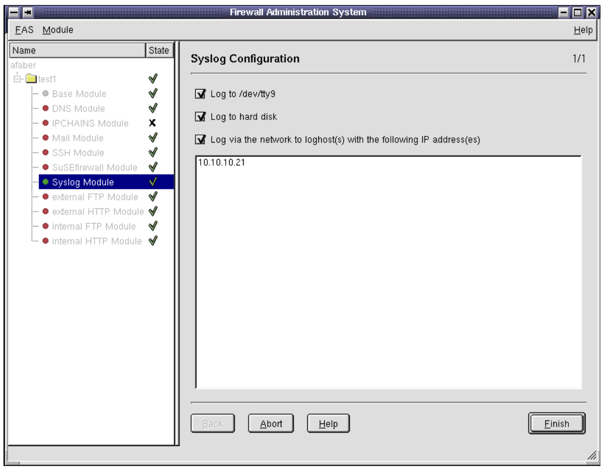
0.0.0.0
10.10.10.1

Figure 3.33: SSH configuration

Syslog configuration:

In this dialog (see Figure 3.34), you can configure the behavior of the syslog daemon.

You have the option of directing the output of the syslogd(8) to the hard disk and/or to a log host. Specify a list of IP host addresses where you want the logs to be written. These hosts must also be configured to be able to receive the logs. The SuSE Linux Firewall Adminhost is already geared toward these tasks.



Name	State
afaber	
test1	✓
Base Module	✓
DNS Module	✓
IPCHAINS Module	✗
Mail Module	✓
SSH Module	✓
SuSEfirewall Module	✓
Syslog Module	✓
external FTP Module	✓
external HTTP Module	✓
internal FTP Module	✓
internal HTTP Module	✓

Syslog Configuration 1/1

[✓] Log to /dev/tty9

[✓] Log to hard disk

[✓] Log via the network to loghost(s) with the following IP address(es)

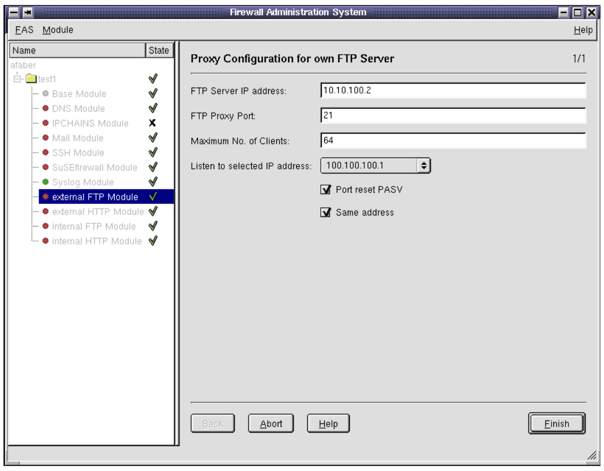
```
10.10.10.21
```

Figure 3.34: Syslog configuration

The configuration of the syslog-ng on the SuSE Firewall Adminhost will be explained more thoroughly below.

FTP access – external to internal:

If you are administrating your own FTP server, you will have to make the following settings here (see Figure 3.35):



Proxy Configuration for own FTP Server 1/1

FTP Server IP address: 10.10.100.2

FTP Proxy Port: 21

Maximum No. of Clients: 64

Listen to selected IP address: 100.100.100.1

[✓] Port reset PASV

[✓] Same address

Figure 3.35: Configuration of the FTP server

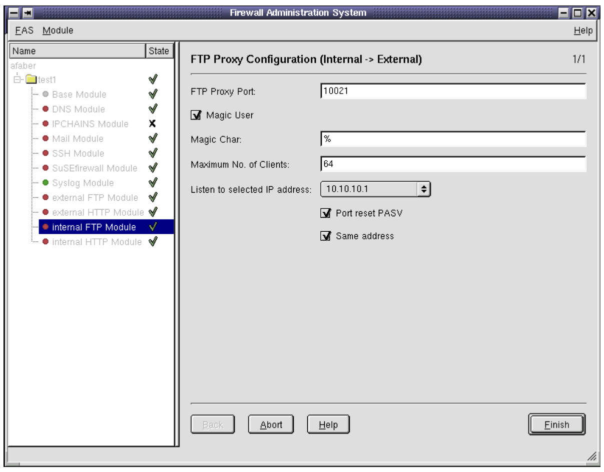
FTP Server IP: The IP address of your Intranet or DMZ FTP server

Port: The port where your FTP server is listening, normally port 21. With this, your firewall is seen externally as an FTP server. Request an alias entry in the DNS for your firewall from your provider, e. g.: ftp.mycompany.com.

Maximal Clients: Maximum number of FTP clients which can be simultaneously connected to the FTP server

Listen to IP: IP address of the interface where the FTP server accepts connections

Configuration of the FTP proxy: internal to external:



FTP Proxy Configuration (Internal -> External) 1/1

FTP Proxy Port: 10021

☑ Magic User

Magic Char: %

Maximum No. of Clients: 64

Listen to selected IP address: 10.10.10.1

☑ Port reset PASV

☑ Same address

Figure 3.36: Configuration of the FTP proxy

FTP Proxy Port: 10021
Port to which the FTP proxy responds.

MagicUser: This parameter allows you to specify the destination FTP server you selected in the username: user[@host[:port]]. This may appear as follows:

```
> ftp user@remoteftp.remote.org:21
```

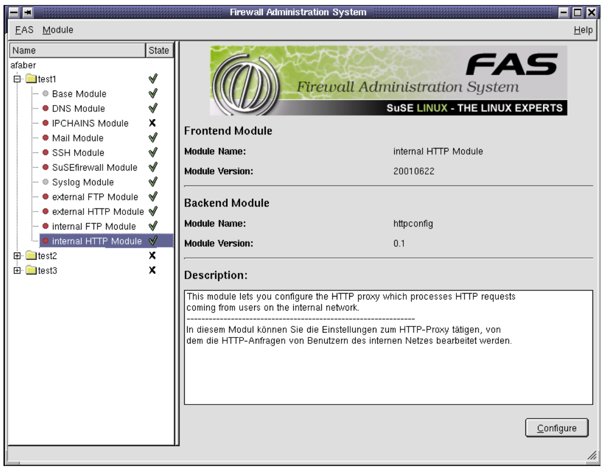
Magic Char: The MagicChar character is normally set to "%". If the option '`Magic User`' is enabled, any character can be used for this.

Maximal Clients: Maximum number of open connections on the FTP proxy.

Listen to selected IP address: IP address over which the FTP proxy responds.

Configuration of the HTTP proxy for internal to external connections

In this module, you can make settings for the HTTP proxy. This module processes HTTP requests by internal network users (see Figure 3.37).



Frontend Module

Module Name: internal HTTP Module

Module Version: 20010622

Backend Module

Module Name: httpconfig

Module Version: 0.1

Description:

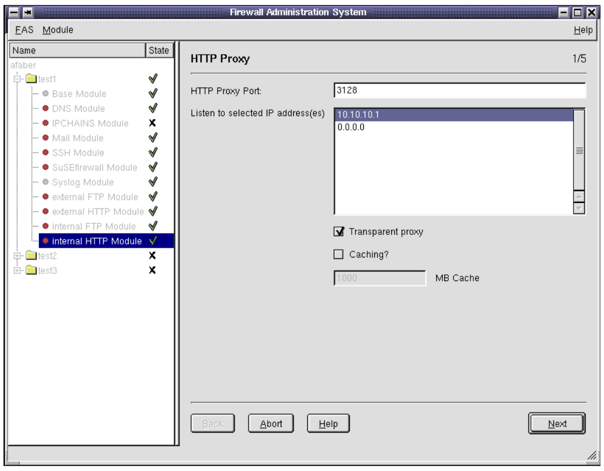
This module lets you configure the HTTP proxy which processes HTTP requests
coming from users on the internal network.

In diesem Modul können Sie die Einstellungen zum HTTP-Proxy tätigen, von
dem die HTTP-Anfragen von Benutzern des internen Netzes bearbeitet werden.

Figure 3.37: Configuration of the HTTP proxy – Start screen

HTTP proxy

The following entries are required for the configuration of the HTTP proxy:



HTTP Proxy 1/5

HTTP Proxy Port: 3128

Listen to selected IP address(es): 10.10.10.1, 0.0.0.0

[✓] Transparent proxy

[ ] Caching?

1000 MB Cache

Figure 3.38: Configuration of the HTTP proxy – Dialog 1

HTTP Proxy Port State the port where the proxy should accept internal HTTP requests. Default setting is port `3128` (see Figure 3.38).

Listen to selected IP addresses Here, define the IP address of the firewall from which HTTP requests can be received. In the selection list, you will find the interfaces specified in the base configuration as internal. 0.0.0.0 stands for all firewall interfaces.

Transparent Proxy Basically, this only monitors the proxy on port 3128. If an HTTP request is submitted, this will run over port 80 and will not be processed. The option '`Transparent Proxy`' must be activated in order to process this request. In conjunction with this, a "redirect" rule (rerouting) must exist, which will reroute the request inside the firewall to port 3128. If this option is not enabled, all clients, or the internal proxy, will have to have the firewall entered as proxy.

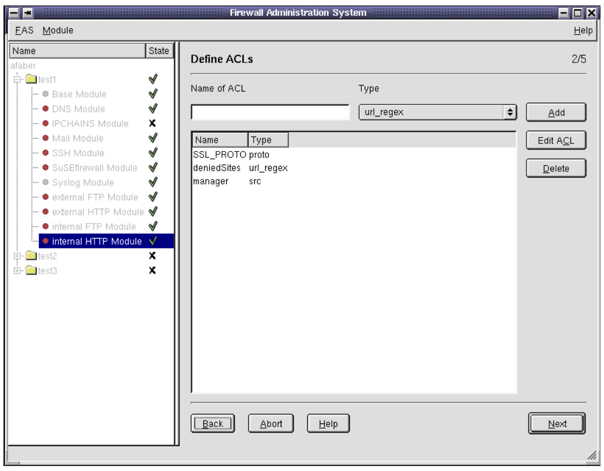
Caching For repeated HTTP requests, you can also activate the option '`Caching?`' to prevent valid pages from being reprocessed. The size of the cache can be defined in the entry field provided and should not be less than 100MB.

Caution

The firewall should not be used as a buffering HTTP proxy, since this is not the actual purpose of the firewall. A better solution is utilizing another independent host.

If you have made all the changes you want, confirm the settings with '`Next`'.

Defining ACLs



Define ACLs 2/5

Name of ACL:

Type: url_regex

Name	Type
SSL_PROTO	proto
deniedSites	url_regex
manager	src

Figure 3.39: Configuration of the HTTP proxy – Dialog 2

ACL (Access Control List) Define who can use the proxy and what these users are able to access in the Internet (see Figure 3.39).

Name for ACL Here, you can name the list to be added.

Next, select a ‘`Type`’ for your ACL. You have the choice between:

- **dest (destination)** Specification of destination addresses.
- **proto (protocol)** Here, you can define the respective protocols.
- **src (source)** Definition of the source addresses.
- **url_regex** Specification of URL addresses.

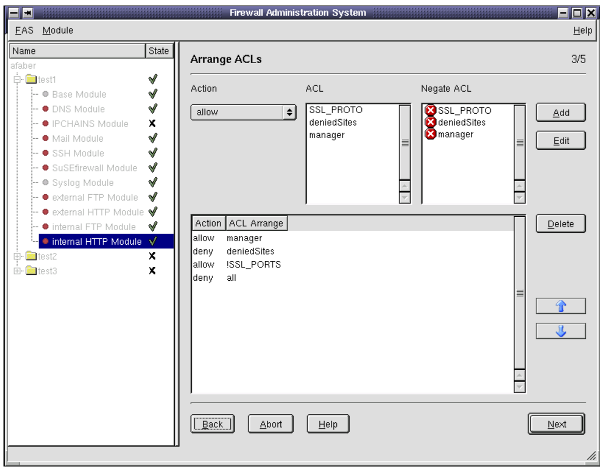
Add Add the new ACL from to the list of already existing ACLs.

Delete You can also delete ACLs using this button.

Edit ACL In this window, you can enter or change values which are to be attributed to an ACL you selected. You can insert new values or edit and delete already existing ones in the entry field.

If you have made all the changes you want, confirm the settings with ‘`Next`’.

Arrange ACLs



Action: allow

ACL: SSL_PROTO, deniedSites, manager

Negate ACL: SSL_PROTO, deniedSites, manager

Action	ACL Arrange
allow	manager
deny	deniedSites
allow	!SSL_PORTS
deny	all

Figure 3.40: Configuration of the HTTP proxy – Dialog 3

You will find various options for setting up the rules (ACL) on this module page. The selected settings for ‘`ACL`’ and ‘`Negate ACL`’ are linked by the logical “AND” (see Figure 3.40).

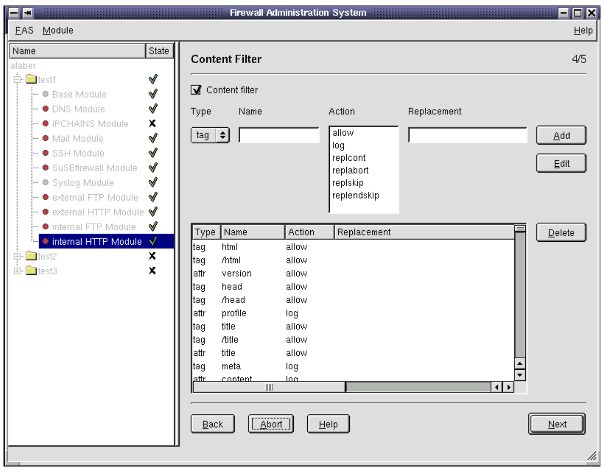
- In the ‘`Action`’ selection field, choose between ‘`allow`’ to permit or ‘`deny`’ to prohibit Internet access which you will define below.
- Specify via ‘`ACL`’ an already existing list to which the settings will apply.
- Specify an ACL which should be negated via ‘`Negate ACL`’. In this manner, you can, for instance, allow retrieval of Internet sites (selection under ‘`ACL`’), which will, however, not run over SSL ports (selection under ‘`Negate ACL`’).
- A new rule can be integrated via ‘`Add`’.

- Click on the respective rule in the list field to edit it. At this point, the settings in the above portion will be applied. You can now modify them and save the changes with '`Edit`'.
- Click on '`Delete`' to remove rules and select the respective item.
- In the lower part of the window, you will find a list field with entries pertaining to all actions and their ACL settings.

Caution
The order of rules in the list field is very important, since the list you set up will be processed from top to bottom. According to the first match, access to the requested URL will either be accepted or denied.

- To move a rule up or down one notch, click on the rule and activate the button in the right window margin and move up or down with the corresponding arrows.
- If you have made all the changes you want, confirm the settings with '`Next`'.

Content filter



☑ Content filter

Type: tag
Action: allow, log, replcont, replabort, replskip, replendskip

Type	Name	Action	Replacement
tag	html	allow	
tag	/html	allow	
attr	version	allow	
tag	head	allow	
tag	/head	allow	
attr	profile	log	
tag	title	allow	
tag	/title	allow	
attr	title	allow	
tag	meta	log	
attr	content	log	

Figure 3.41: Configuration of the HTTP proxy – Dialog 4

To filter or search for certain HTML page contents and, if needed, to block contents, activate the checkbox next to '`Content Filter`' (see Figure 3.41).

Note
You should be able to program HTML in order to configure this filter. The "tags" and "attributes" discussed in the following are HTML components, the programming language in which web sites are written.

All the added filter settings can be found in the lower window panel. There, you will see names attributed to types, actions defined and substitutions specified. Please note that all undefined HTML tags or attributes will be denied.

Proceed as follows:

- To add a filter rule, select the respective type from the selection field with the same name. You can choose between '`tag`' and '`attr`' for an attribute. In the '`Name`' entry field, specify the text of the "tag" or "attribute" and then select an action. Multiple labels for actions are also possible. You can choose from:

  allow permits the selected tag/attribute.

  log sets an entry into the so-called "log file" and can be used later for setting up filter strategies.

  replcont replaces the tag/attribute and continues evaluating the attribute.

  replabort replaces the tag/attribute and stops evaluating the attribute.

- Both of the following actions serve to remove entire coded sections from the requested documents:

  replskip shows the beginning of an omission, which will not be forwarded to the user.

  replendskip shows the end of the omission.

- In the entry field next to '`Replace by`', enter the string which will be used instead of the first one. No blanks spaces are permitted. Instead, use `&#32;` to insert a blank space. Here, of course, it only makes sense to insert something if you have selected '`replcont`', '`replabort`', '`replskip`' or '`replendskip`' as an action.

- To edit a filter rule, select it from the list field. You will then see the values in the entry and selection fields located in the upper portion of the window. Modify the values and confirm by clicking on '`Edit`'.

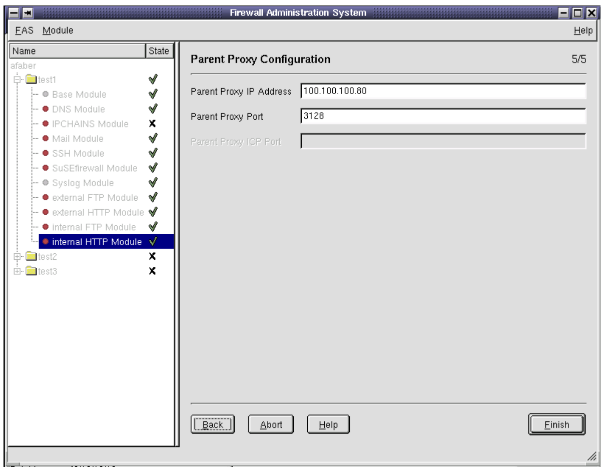
- To remove filter rules, select them from the list field and click on '`Delete`'.

Parent proxy configuration

If your provider supplies specific proxy for HTTP requests, the IP address of the proxy can be entered in the field next to '`IP address of the parent proxy`'. The provider's HTTP port is then entered under '`Parent Proxy Port`' (see Figure 3.42 on the next page).

If the '`Content Filter`' is disabled, you can specify under '`Parent Proxy ICP Port`' (ICP = internet caching protocol) the provider's ICP port if their proxy supports ICP.

If you have made all the changes you want, confirm the settings with '`Next`'.



Parent Proxy Configuration 5/5

Name	State
afaber	
test1	✓
Base Module	✓
DNS Module	✓
IPCHAINS Module	✗
Mail Module	✓
SSH Module	✓
SuSEfirewall Module	✓
Syslog Module	✓
external FTP Module	✓
external HTTP Module	✓
internal FTP Module	✓
internal HTTP Module	✓
test2	✗
test3	✗

Parent Proxy IP Address: 100.100.100.80

Parent Proxy Port: 3128

Parent Proxy ICP Port:

Figure 3.42: Configuration of the HTTP proxy – Dialog 5

Configuration of the HTTP proxy for external to internal connections

If you are not managing your own web server, you will not have to configure anything here. Otherwise, please make the following settings (see Figure 3.43 on the facing page):

Web Server IP: The IP address of your web server located on your Intranet.

Web Server Port: Normally port 80.

Proxy Port: The port listens on the proxy, normally port 80. In this way, your firewall becomes the web server as viewed from the outside. Therefore, request an alias for your firewall from your provider, e. g.: `www.my-company.com`

Listen to IP: IP address of the interface where the proxy accepts connections.

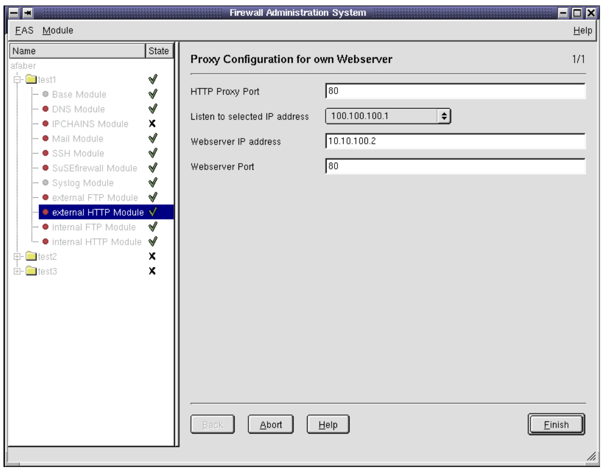
Saving the configuration

A configuration can be saved by clicking on '`Save configuration`' in the '`Configuration`' menu. Once you have exited the configuration and modified the settings, you will be asked if you want to save the changes. The configuration is archived with **`tar`** and compressed with **`gzip`** and saved to the hard disk in the file

`/var/lib/fas/<username>/configs/<configurationname>.tar.gz`.

No normal user on the adminhost can read this configuration. Only `'root'` has the necessary permissions for this.

In order to create a floppy disk, insert it into the drive, select the configuration you want to have saved to disk and then click in the menus, first on '`Configuration`' and then on '`write to floppy`'. Now, the disk will be written. The



Name	State
afaber	
test1	✓
Base Module	✓
DNS Module	✓
IPCHAINS Module	✗
Mail Module	✓
SSH Module	✓
SuSEfirewall Module	✓
Syslog Module	✓
external FTP Module	✓
external HTTP Module	✓
internal FTP Module	✓
internal HTTP Module	✓
test2	✗
test3	✗

Proxy Configuration for own Webserver 1/1

HTTP Proxy Port	80
Listen to selected IP address	100.100.100.1
Webserver IP address	10.10.100.2
Webserver Port	80

Figure 3.43: Configuration of the external HTTP proxy

configuration must have previously been recognized as "properly configured", in which case, a green check mark will appear in the left portion of the dialog.

3.4.4 Editing an existing firewall configuration

To edit an existing firewall configuration, start the Firewall Administration System FAS. Log on as the user you used to create the configuration via the menu '`FAS`' → '`Login`', and then enter your password. Now you will obtain a list of the configurations previously created by this user.

Select the configuration from the list which you want to edit. Double-clicking on the configuration name in the list located in the left window panel lets you edit one of the services in its respective module.

3.4.5 Testing the configuration

The configuration created using the administration application still has to be tested before it can be put to productive use.

To do this, the firewall is started offline (connected neither to the Internet nor to your Intranet). To carry out the testing, the firewall is connected directly to the adminhost (crossover cable).

Testing for the packet filter can easily be done with a port scanner. To this end, the program nmap is installed on the adminhost. If you carry out a portscan from the adminhost, please be aware that you must disable the packet filter of the adminhost. Log in as user '`root`' and enter the following command in a console:

```
root@adminhost: # SuSEfirewall stop
```

In this way, you can ensure that all returning IP packets are able to be received by the adminhost. Do not forget to reactivate the packet filter after completing the test:

```
root@adminhost: # SuSEfirewall start
```

Following the port scan, the scanning results and the log file should be documented on the firewall and retained.

Make sure that the following services are running properly:

- Name server test: e.g. with **`nslookup`** plus name of the firewall. Check the log file, e.g. via **`grep named /var/log/messages`**. There should not be any messages containing “error”.
- Mail relay test: by sending an e-mail and checking the log files `/var/log/mail` and `/var/log/messages`. Search the log files for “postfix” (**`grep postfix /var/log/mail`**
- FTP proxy test: e.g. via **`telnet`** to your firewall.
- Test the HTTP proxy by attempting to access the Internet from a client.
- Test `ssh` by logging on to the firewall from a client.

Always verify all processes by referring to the log files.

3.4.6 Documenting the configuration, the tests and the results

It is essential that you document the configuration, any tests performed, and their results. Record, in other words, what is permitted or prohibited by the configuration, and how this is guaranteed.

By means of such documentation, you may be able to detect configuration errors and remedy them. This documentation is also necessary for auditing the firewall.

3.5 Monitoring the Firewall

As already mentioned several times, a firewall’s functionality is quite limited if is not accompanied by constant monitoring. Some tools for monitoring the firewall are located on the adminhost. The most important sources of information are the log files, which are written either to the loghost or the hard disk, depending on the configuration.

The following programs are available as tools for analyzing log files.

- xlogmaster
- logsurfer
- sylog-ng

The following network or packet sniffers can be used to monitor your firewall:

- ntop
- tcpdump
- ethereal

With the following port scanners, you can examine the firewall on open ports and monitor the packet filter configuration:

- nmap
- nessus

In addition, you can, of course, use shell or perl scripts that you have written yourself.

Configuration of syslog-ng

In order to configure the login daemon **`syslog-ng(8)`** on the loghost, log in to a console as user `‘root’` or enter the command **`su - root`** in an xterm. If the SuSE Linux Adminhost for Firewall is not your loghost, copy the program `/opt/fas/sbin/fas_syslogng_config.pl` to your loghost.

Start the program as `‘root’` with the following command:

```
root@loghost: # /opt/fas/sbin/fas_syslogng_config.pl
```

Specify the hostname of your firewall. Define whether syslog-ng is to be started when the system is booted. The default setting is "yes".

The hostname of the firewall must be able to be resolved to an IP address. You can determine if this is the case by referring to an entry in the `/etc/hosts` file on your loghost. For this, refer to the example 3.5

```
# Firewall logging entry
# IP address of firewall   hostname.mydomain.com hostname
# e.g.
192.168.0.1    stargate.mydomain.com  stargate
```

File contents 3.5.1: Sample entry in the /etc/hosts file

4 SuSE Linux Live CD for Firewall

4.1 Introduction

The “Live CD” for SuSE Linux Firewall on CD is the “executive branch” of the firewall. The Live CD is a minimal SuSE Linux equipped to meet security needs. This applies to the available applications as well as to the kernel itself. The “SuSE Firewall on CD” is, in addition, an “Application Level Gateway”, that means, due to security reasons, IP packet routing should not set be up. Forwarding requests to services is managed by applications (= proxies).

Proxies and non-forwarding IP packets are not enough to prevent (undesired) Internet IP packets from reaching the Intranet or vise versa. This firewall functionality is adopted by the kernel packet filter which is configured by ipchains(8). This is where the idea of the Live CD comes into play. That is, the operating system and all the applications are located on a CD, on a read-only filesystem. A RAM disk, where the Live CD is mounted, is generated when booting. The “status quo ante” can simply be restored by rebooting the machine. Updating the “SuSE Linux Firewall on CD” is likewise very simple because of this: Just swap the Live CD for the current update CD and restart your firewall host.

Configuring the system and the services is done by disk, where all the necessary configuration files are saved. This disk is mounted “read only” when booting. Just to be on the safe side, the disk should also be write-protected. The data on the configuration disk will be copied to the RAM disk and the disk itself removed from the filesystem.

Hardware requirements:

- The “SuSE Linux Live CD for Firewall” is executable on any **`ix86`**, where x=5. At least a **`Pentium II`** is recommended.
- At least `128 MB` RAM is required,
- a 3,5-inch floppy disk drive
- a bootable CD-ROM drive and
- at least two network interfaces

This chapter does not provide any instructions on how to configure firewall services, but rather, technical documentation for the well-versed administrator willing to grapple with the internal aspects of the system. We will also describe details relating to the configuration files of the services provided on the Live CD. For configuration purposes, the Firewall Administration System (FAS) on the SuSE Firewall Administration Host should be used.

4.2 Description of the SuSE Linux Live CD for Firewall

The “SuSE Linux Live CD for Firewall” is a Live File System CD from which all the applications will run directly. Theoretically, the firewall host could be driven without a hard disk. However, you should note that a hard disk for the cache and spool directories is required by proxy services such as Squid or postfix. A hard disk is likewise required if you want to save the syslogd log messages locally. In order to set up the firewall system, insert the CD and the configuration disk created by the SuSE Firewall Adminhost and boot the host.

4.3 Services on the Firewall

“Application level gateways”, or proxies, are located on the “SuSE Linux Live CD for Firewall” for the most common and essential Internet protocols:

DNS (engl. *Domain Name System*): IP addresses can be converted into “Fully Qualified Domain Names” and vice versa with bind8

SMTP postfix manages the transportation of e-mail messages,

HTTP/HTTPS – the WWW Protocol: Squid, httpf, transproxy, tinyproxy

FTP ftp-proxy-suite is used for transmitting files from one host to another.

SSH “Remote logins” with encrypted transfers take place via openssh, the authentication program for RSA key pairs.

All these applications involve open source software. All these processes run on the “SuSE Linux Live CD for Firewall” in chroot environments.

In order to raise the security level of the system even higher, the following programs and kernel modules are used for the Linux Kernel (2.2.19) implemented by the Live CD: `secumod`, `compartment`, `OpenWall-Patches`.

4.3.1 IPCHAINS

The packet filter of the Linux kernel 2.2.xx is configured using the ipchains application. ipchains allows for extremely flexible configuration of the kernel packet filter. To do this, refer to the manpage for ipchains(8).

Normally, ipchains(8) recognizes three packet filters-“Chains”: input-chain, forward-chain and output-chain.

These three filters determine the fate of an IP packet arriving at an interface. It is advisable to determine a policy for these “chains” which denies all packets. This will then be the default behavior of the chain: Everything which is not expressly permitted will be denied.

IP packets cycle through the chains and will be treated in respect to their address of origination, destination and port, i.e. forwarded, discarded or rejected. There are basically two options provided by the “SuSE Linux Live CD for Firewall” for configuring ipchains filter rules. The first and rather convenient option is to utilize the SuSE Firewall Script. The SuSE Firewall Script is configured depending on the configured and activated services. With its Firewall Administration System (FAS), the SuSE Firewall Adminhost provides easy access to the SuSE Firewall Script. The second available option is configuring your own packet filter with ipchains. First, we will give you an overview of the SuSE Firewall Script and subsequently, instructions for setting up your own packet filter.

The SuSE Firewall Script and configuration

Below is a technical description of how filter rules are generated as well as some background on the new SuSE Firewall Script design. However, this documentation does *not* offer instructions on how to configure the firewall – the configuration file `/etc/rc.config.d/firewall.rc.config` contains sufficient comments which simplify the process. Furthermore, you will find some sample configurations in the `EXAMPLES` file in the `/usr/share/doc/packages/SuSEfirewall` directory. Recently, an FAQ list has been added to the same directory. Please be aware that “firewall” is referred to throughout this documentation, even though this script is technically not a firewall itself. This package is a so-called “packet filter” which accepts or denies data on the TCP/IP level. In other words: if you are operating a WWW server and you want to open communication between it and the Internet, this script will not provide any protection if the web server as such were to exhibit a security problem. However, it will enable you to protect services which should not be accessed, keep potential hackers in the dark about your configuration, as well as recognize intrusions by detecting filter violations.

The design of the SuSE Firewall Script satisfies the following demands:

- **Above all, security**

  Those installing and configuring this package expect that the script will do everything possible to secure the system. A user must, however, take into account that some things are more difficult or perhaps (in rare cases) not at all configurable in this package. Also, setting up such as firewall is associated with some time investment.

- **No compromises to the system in case of errors**

  An error in the script or in the configuration file should not lead to open filter rules. However, this goal is, in certain areas, difficult to realize. In any case, SuSE has made a strong effort to achieve this goal.

- **Simple configuration**

  Simple questions are asked – but also, only so many are asked as absolutely necessary. This is, of course, the greatest challenge – but you will still need to have prior TCP/IP knowledge, administrator skills, a knack for system security and patience. . .

- **Automated configuration**

  To simplify the configuration and also, for better support of dynamic systems, the SuSE Team has tried to determine as much information as possible at the very moment the firewall script starts.

- **Support of dynamic IP addresses + networks**

  Not only a dynamic IP address, but also several network cards (an infinite amount) for the Internet as well as for the internal network are supported.

The following section contains frequent references to filter rules and networks. Filter rules, which apply to internal interfaces or networks, is only processed if they are configured and available immediately when the firewall script starts. (If they were only accessible after starting the script, for instance, if the interface were "down", network communication would no longer be possible!)

The SuSE Firewall Script procedure which is located in `/sbin`, can be outlined as follows:

a) First, the configuration file, `/etc/rc.config.d/firewall.rc.config`, is read. Before anything can happen, the user will first have to configure `/etc/rc.firewall`. These files will be created automatically if you are using FAS.

b) A search is then conducted for auxiliary programs such as sed, awk, grep, ifconfig, netstat and, of course, ipchains. If any one of these programs cannot be detected, an error message will abort the script. Next, the kernel version is verified to determine whether ipchains is supported. The script will likewise be terminated if a 2.0 kernel is identified, or a warning issued if the kernel version cannot be identified at all.

c) Now, the interpretation of the configuration file follows, along with the evaluation of the important data at hand. In this way, data for all the network cards configured in the script, such as IP address and netmask, will be scanned. If an interface is not "up" (e. g. an ISDN or a PPP interface), filter rules will still be generated later in the script procedure, but only a few rudimentary security mechanisms will be in place. After setting up a connection with ISDN, for example, the SuSE Firewall Script should be started again.

d) Now you can begin! First, the rules are reset and the default for incoming packets as well as for packets for routing the firewall set to "discard". If, for some reason, a packet in the script does not have any rules, this packet will always be discarded. Packets wanting to exit the firewall will be set to the target value "allow".

e) If the firewall is set to "route", this will activate routing.

f) The `/proc` system allows you to easily configure the kernel while the system is running. The script makes use of this for the purpose of activating some security options. For many of these options, the option `FW_KERNEL_SECURITY` in the configuration file must first be set to "yes", since it could otherwise have rather complex consequences.

g) Then – most importantly – all traffic via the interface "localhost" is released.

h) Now, the script will handle the rules pertaining to "IP spoofing" and "bypass prevention". These rules ensure that external/internal attacks which pretend to be from another network segement are recognized and refused. At the same time, attempts to directly break into the network will be impeded.

i) The redirecting rules which follow can be used to reroute attacks on the internal network or local ports to a special firewall port.

j) A user connected to an internal and trustworthy network can specify in the configuration file whether the firewall should be protected against internal attacks or not. If not, from this point on, any kind of traffic on the internal network will be released to the firewall.
The option for this is `FW_PROTECT_FROM_INTERNAL`.

Then, rules for ICMP, TCP and UDP are generated.

k) ICMP rules are generated in two phases. First, special ICMP packages are allowed or denied according to the configuration: Packages such as "ping" to the firewall, "source quench" messages of the next front-end router as well as response packages for similar programs such as traceroute. The second phase is comprised of general configurations prohibiting dangerous ICMP packages and allowing the essential ones. The internal network can always access the firewall with "ping".

l) For TCP rules, the configured services are first released to the outside, then to trustworthy networks, and finally, to those services designated for the internal network. Subsequently, port 113 will be configured to send a connection reset in order to prevent unnecessary wait times (e. g. when sending e-mails). Now, a noteworthy automatic process comes into play:
If `FW_AUTOPROTECT_GLOBAL_SERVICES` has been set to "yes", these services will be protected by filter rules which do *not* listen to any special interfaces (to `0.0.0.0` or `INADDR_ANY`). In this way, all high ports for incoming connections can be released, if necessary, but with the assurance that your databases etc. which may be running (on port `4545`, for instance) will still be protected. In conclusion, rules will be generated as to whether and how unprivileged ports (between `1024` and `65535`) can be accessed: Not at all; only name servers as defined in `/etc/resolv.conf`; only connections originating from a given port (not recommended since this safeguard is easy to circumvent); all connections. Internal systems can *always* access unprivileged ports – with the exception of services, which are all protected by AUTOPROTECT.

m) The same is done for UDP, except that a connection reset for port `113` is not required.

n) Next, the script handles the routing rules for defining on which outside systems access can be made to an internal network. These routing rules should, of course, not be used for obvious reasons!

o) Now the masquerading rules are configured.

p) The configuration of additional log mechanisms for special packets is very important, e.g. forbidden TCP packets attempting to establish a connection, etc. If `LOG_*_ALL` is set, *every* packet really will be logged, which will generate quite a few log entries, and should thus only be used for troubleshooting.

q) Last but not least, a few optimization rules for SSH, FTP, WWW, Syslog and SNMP, for making these faster and more secure (in terms of transfer over the network).

This completes the script. If there is an error, the return value is 1, otherwise, the return value is 0.

Now it is necessary to initialize the firewall during the booting process. The name of the actual rc script is `/sbin/init.d/firewall`. It is called up in runlevel 2 with `S04firewall_init` and `S99firewall_setup`. `S04` only generates rudimentary rules and does not return any errors. This is only done when the `S99` script is run, once all services and interfaces have been properly configured. K51firewall removes all filter rules at ("shutdown") and allows any packet, while disabling the routing.

The firewall should always be started dynamically as

```
/sbin/init.d/rc2.d/S99firewall_setup start
```

using dialup scripts, so that filter rules will only be generated if it is configured to do so.

TRACEROUTE

To activate traceroute and other similar applications, the following must be configured for them to function properly.

```
FW_ALLOW_FW_TRACEROUTE=yes  (at the very end of the expert-config)
FW_ALLOW_FW_PING=yes
FW_ALLOW_INCOMING_HIGHPORTS_UDP=yes
```

If you only make these settings and are using the SuSE Firewall, your system is, however, in still not secure as it is!

To further increase the security level of a firewall server, you should:

- Minimize any services to insecure networks (such as the Internet), only use applications which are considered safe (e. g. postfix, ssh etc.) and configure these carefully. In addition, you will need to ascertain that these applications are not, in fact, exhibiting any security gaps. If at all possible, do *NOT* run any services as `'root'` and do not run any services in a "chroot" environment.
- Do not rely on any software which you have not tested yourself first.
- Check your server's integrity on a regular basis.
- If you want to use the firewall/bastion server as a masquerading or a routing server, which is not recommended, you should check to see if proxy services can be implemented instead, such as Squid for the WWW, smtpd for e-mail

etc. Also, beware of "chroot" here as well. Do not let any processes run as `'root'`. Disable this machine's routing.

Below, you will find explanations on the SuSE Firewall Script configuration file `/etc/rc.config.d/firewall.rc.config`.

As already mentioned, you will not have to set up this file by hand. Use the Firewall Administration System on the SuSE Firewall Adminhost. However, if you want to edit this file for your own purposes, by all means read the Firewall Script documentation and review your script thoroughly before running it. No guarantee or support can be offered for self-made configurations.

Note

Please observe the following note:
Configuring these settings and using the SuSE Firewall Script will not, on its own, make your system secure on its own. There is *NOTHING* which can simply be installed all at once and guarantee full protection against security leaks.

For a "secure" system, you should also keep in mind the following:

Back up any services provided for unreliable networks (such as the Internet) with software which has been developed to accommodate security requirements (e. g. postfix and ssh). Do not implement any software originating from unreliable sources. Moreover, you should regularly check the security of your server.

In order to be able to start the firewall script, the variable `START_FW` in the `/etc/rc.config` file must be set to "yes". If the SuSE Firewall Adminhost is being implemented, this will take place automatically.

If this server is a firewall functioning as a proxy (no routing between networks), or if you are an end-user connected simultaneously to the Internet and the Intranet, you will need to configure the items 2), 3), 9) and possibly 7), 10), 11), 12), 14) and 18) as well.

If the machine is functioning as a firewall and is to provide routing and masquerading functionality, you will then need to change the items 2), 3), 5), 6), 9), and possibly 7), 10), 11), 12), 15), 18).

If you know exactly what you want to do, you can reconfigure the items 8), 16), 17), 18) and the expert options 20) 21) and 22) in the outermost margin of the file. However, SuSE recommends that you refrain from doing this.

If you wish to implement "diald" or "ISDN dial-on-demand", you will want to edit the item 18).

To use applications such as **`traceroute`** together with your firewall, you will have to set the following options: 11) (only UDP), 19) and 20) to "yes".

If you want to start all the packet filter rules yourself, configure a static IP and netmask for these if they do not already exist. Note the examples 2), 3) and 4).

Please be aware that if you use service names, they will be located in `/etc/services`. Note that there is no service "DNS", the correct name is "domain"; for e-mail, "smtp" etc.

Every time routing takes place between interfaces, with the exception of masquerading, **`FW_ROUTE`** will have to be set to `yes`. Otherwise, use the variables **`FW_FORWARD_TCP`** and/or **`FW_FORWARD_UDP`**.

1. Should the firewall script be started at the system boot?

   For this, the variable **`START_FW`** in `/etc/rc.config` must be set to "yes".

2. Which network interfaces are connected to the Internet or to unreliable networks?

   Specify all unreliable networks here. You can enter an unlimited number of interfaces, each separated by a blank space, e. g. "eth0" or "ippp0 ippp1".

   `FW_DEV_WORLD=""`

   You can specify a static IP address and a network mask address and thereby force a packet filter rule to be loaded for an interface which is not yet available.

   `FW_DEV_WORLD_[device]="IP_ADDRESS NETMASK"`

   You will still have to define **`FW_DEV_WORLD`** first! The entry will then appear as follows:

   `FW_DEV_WORLD_ippp0="10.0.0.1 255.255.255.0"`

3. Which network interfaces are associated with the internal network? Specify all trustworthy network devices here. If you are not connected to a trustworthy network (e. g. if you only have a dialup connection) then leave the list blank. You can enter an unlimited number of network devices, each separated by a blank space, e. g. "eth0" or "eth0 eth1 ippp0".

   `FW_DEV_INT=""`

   You can specify a static IP address and a network mask address and thereby force a packet filter rule to start for an interface which is not yet available:

   `FW_DEV_INT_[device]="IP_ADDRESS NETMASK"`

   You will still have to define **`FW_DEV_INT`** first! See the following example for the internal interface "eth0":

   `FW_DEV_INT_eth0="192.168.1.1 255.255.255.0"`

4. Which interface is linked to the DMZ?

   Specify all network devices linked to the DMZ. Caution: Configure the variables **`FW_FORWARD_TCP`** and **`FW_FORWARD_UDP`** which specify the services to be provided for the Internet, and set the variable **`FW_ROUTE`** to "yes". You have the choice of: "tr0", "eth0 eth1" or "".

   `FW_DEV_DMZ=""`

   You can specify a static IP address and a network mask address and thereby force a packet filter rule to start for an interface which is not yet available:

   `FW_DEV_DMZ_[device]="IP_ADDRESS NETMASK"`

   You will still have to define **`FW_DEV_DMZ`** first! See the following example for the DMZ interface "eth1":

   `FW_DEV_DMZ_eth1="192.168.1.1 255.255.255.0"`

5. Should “routing” be activated between the Internet, the DMZ and the internal network? For this, you will need **`FW_DEV_INT`** or **`FW_DEV_DMZ`**.

   You will only need to set the variable to "yes" if you either want to mask internal machines or allow access to the DMZ. This option overwrites the variable **`IP_FORWARD`** in `/etc/rc.config`.

   If you set this option, nothing will happen yet. Either activate the masquerading with the variable **`FW_MASQUERADE`** further below, or configure **`FW_FORWARD_TCP`** and/or **`FW_FORWARD_UDP`** to define what is to be routed. You can only specify "yes" or "no" here, the default being "no".

   `FW_ROUTE="no"`

6. Do you want to mask outgoing internal networks? To do this, use **`FW_DEV_INT`** and **`FW_ROUTE`**.

   “Masquerading” means that all your internal machines which use Internet services, will appear as if they were originating from the firewall. Please note that it is more secure to communicate over proxies with the Internet than to implement masquerading.

   Options: "yes" or "no", default: "no"

   `FW_MASQUERADE="no"`

   Which internal hosts/networks should have access to the Internet? Only these networks can go on the Internet and will be masked. Leave the list blank or specify any number of networks/hosts, each separated by a blank space, a protocol type and service can be added to each network/host, separated by a comma.

   FW_MASQ_NETS=""

   In addition, specify the interface where the masking is to take place, e. g. "ippp0" or "$FW_DEV_WORLD"
   `FW_MASQ_DEV="$FW_DEV_WORLD"`

7. Do you want to protect the firewall from internal networks? This can be accomplished with the variable **`FW_DEV_INT`**. If this variable is set to "yes", internal computers can only use the services you provided.

   Options: "yes" or "no", basic setting: "yes"

   `FW_PROTECT_FROM_INTERNAL="yes"`

8. Do you want to protect all globally running services?

   If you specify "yes" here, all network access attempts to TCP and UDP, not associated with a particular IP address or not otherwise explicitly permitted, will be prohibited on this machine. See **`FW_*SERVICES_*`**. "0.0.0.0:23" will be protected, for example, but "10.0.0.1:53" will not. Options: "yes" or "no", default is "yes".

   `FW_AUTOPROTECT_GLOBAL_SERVICES="yes"`

9. Which services on the **Firewall host** are to be opened to the Internet or any other unreliable network, to the DMZ, or to the internal network? (See Nr.13 & 14, if you want to route your network traffic over the firewall.) Specify all port numbers or port names, each separated by a blank space. TCP-based

services (e. g. SMTP, WWW) must be set in **`FW_SERVICES_*_TCP`**– and UDP services (syslog) in **`FW_SERVICES_*_UDP`**. For IP protocols (such as GRE for PPTP or OSPF for routing), **`FW_SERVICES_*_IP`** must be specified by the protocol's name or number (see also `/etc/protocols`).

Options: blank list or port number, port name (see `/etc/services`) or port zones (e. g. 1024:2000), each separated by a blank space, or services (TCP) which will be visible from the outside, normally "smtp domain".

`FW_SERVICES_EXTERNAL_TCP=""`

Services (UDP) which are to be visible from the outside, normally "domain":

`FW_SERVICES_EXTERNAL_UDP=""`

Services (all other) which are to be visible for the DMZ, e. g. VPN/routing, terminating at the firewall:

`FW_SERVICES_EXTERNAL_IP=""`

Services (TCP) which are to be visible for the DMZ, normally "smtp domain":

`FW_SERVICES_DMZ_TCP=""`

Services (UDP) which are to be visible for the DMZ, normally "domain syslog":

`FW_SERVICES_DMZ_UDP=""`

Services (all other) which are to be visible for the DMZ, e. g. VPN/routing, terminating at the firewall:

`FW_SERVICES_DMZ_IP=""`

Services (TCP) which are to be visible for the internal network, normally "ssh smtp domain":

`FW_SERVICES_INTERNAL_TCP=""`

Services (UDP) which are to be visible for the internal network, normally "domain syslog":

`FW_SERVICES_INTERNAL_UDP=""`

Services (all other) which are to be visible for the DMZ, e. g. VPN/routing, terminating at the firewall:

`FW_SERVICES_INTERNAL_IP=""`

10. Which services should be reachable from trustworthy networks in the Internet?

    Specify the trustworthy networks in the Internet and the TCP/UDP services they can use.

    Options: blank lists or any IP address and/or networks, each separated by a blank space.

    `FW_TRUSTED_NETS=""`

    Specify a blank list next to **`FW_SERVICES_TRUSTED_*`**, or the port numbers and the known port names (see `/etc/services`), or the port zones, each separated by a blank space, e. g. "25", "ssh", "1:65535", "1 3:5"

Services (TCP) which are to be made available to trustworthy networks/hosts, normally “ssh”:

`FW_SERVICES_TRUSTED_TCP=""`

Services (UDP) which are to be made available to trustworthy networks/hosts, normally “syslog time ntp”:

`FW_SERVICES_TRUSTED_UDP=""`

Services (all other) which are to be visible to trustworthy hosts, e. g. VPN/routing, terminating at the firewall:

`FW_SERVICES_TRUSTED_IP=""`

Sometimes, some trustworthy hosts are to have access to certain services, and other trustworthy hosts to other services. Here, you will have the chance to set up this option according to the pattern: "trusted_net,protocol,port", so for example, "10.0.1.0/24,tcp,80 10.0.1.6,tcp,21":

`FW_SERVICES_TRUSTED_ACL=""`

11. Is access to high, unprivileged (>1023) ports allowed?

    You can allow ("yes") or deny ("no") all access to the high ports; or allow any access originating from a given port (specification of port number or service name, which can, however, be easily circumvented) or allow only name servers (DNS) – which you authorize – to have access.

    If you want to enable active FTP, you will have to set the TCP variable to "ftp-data". For passive FTP, this is not necessary. Still keep in mind that you cannot use any rpc requests. If you want to allow this, you will have to set the port zone "600:1023" in **`FW_SERVICES_EXTERNAL_UDP`**.

    Options: "yes", "no", "DNS", port number or port name, Default is "no".

    Incoming connections to port numbers >= 1024, TCP, normally: "ftp-data" (unfortunately!):

    `FW_ALLOW_INCOMING_HIGHPORTS_TCP="yes"`

    Incoming connections to port numbers >= 1024, UDP, normally: "DNS" or "domain ntp":

    `FW_ALLOW_INCOMING_HIGHPORTS_UDP="yes"`

12. Do you have one or more of the following services running? These require a bit of caution, otherwise they will not function. For services which you want to provide, set the variable to "yes" and the others to "no". Default values are "no".

    `FW_SERVICE_DNS="no"`
    If "yes", port (or domain) 53 must have **`FW_SERVICES_*_TCP`**. The variable **`FW_ALLOW_INCOMING_HIGHPORTS_UDP`** must be set to "yes".

    `FW_SERVICE_DHCLIENT="no"`
    If you are using dhclient to obtain an IP address, **`FW_SERVICE_DHCPD`** must be set to "yes".

    `FW_SERVICE_DHCPD="no"`
    If this host is a DHCP server, set **`FW_SERVICE_DHCPD`** to "yes".

FW_SERVICE_SAMBA="no"
If you want to implement Samba as a client or server on this host, set **FW_SERVICE_SAMBA** to "yes".

For a Samba server,
FW_SERVICES_{WORLD,DMZ,INT}_TCP="139" will still have to be set. Overall, it is a bad idea to use Samba on a firewall host.

13. Which services to be accessible from the Internet may access the DMZ or internal network?

    With this option, you can allow access for your mail server, for instance. This host must possess a valid IP address. If an open connection is to be established to your network, then just use this option to access your DMZ. The following values are possible: Leave the variable empty (the best choice) or use the following syntax for the forwarding rules. The rules are to be separated by blank spaces.

    A forwarding rule consists of:

    a) Source IP/network,

    b) Destination IP (dmz/internal),

    c) Destination port (or IP protocol), separated by a comma

```
# Forward TCP connections
# Beware of using this!
FW_FORWARD_TCP=""
# Forward UDP connections
# Beware of using this!
FW_FORWARD_UDP=""
# Forward other IP protocol connections (for VPN setups)
# Beware of using this!
FW_FORWARD_IP=""
```

14. Which services can be used on masked hosts (in your DMZ or internal network)?

    REQUIRES: FW_ROUTE, FW_MASQUERADE

    With these variables, you can allow access to your mail server, for instance. The host must be located in a masked network segment and cannot have an official IP address.

    If **FW_DEV_MASQ** is set to your external interface, you will likewise have to set the variables **FW_FORWARD_*** from your internal network to the DMZ for these services.

Caution

Due to security reasons, you should not use this feature, as it opens a security gap in your internal network. If your web server were to be compromised, so too would your entire network become vulnerable to attack.

Options: Leave the entry blank (best choice) or use the following syntax for the forward masquerade rules. The rules are to be separated by blank spaces.

a) Source IP/network,

b) Destination IP (dmz/internal)

c) Destination port, separated by commas,
e. g.: "4.4.4.4/12,20.20.20.20,22 12.12.12.12/12,20.20.20.20,22"

```
# Forward TCP connections to masqueraded host
# Beware of using this!
FW_FORWARD_MASQ_TCP=""
# Forward UDP connections to masqueraded host
# Beware of using this!
FW_FORWARD_MASQ_UDP=""
# it is not possible to masquerade other IP protocols,
# hence no _IP variable
```

15. Which services should be rerouted to a local port on the firewall host?

    This can be used to relay all your internal users over your HTTP proxy or to forward all incoming requests transparently to port 80 on a secure web server.

    Option: Leave the list blank or use the following syntax for redirecting rules:

a) Source IP/network,

b) Destination IP/network,

c) original specification port,

d) local port to which the network traffic is to be rerouted,
e. g.: "10.0.0.0/8,0/0,80,3128 0/0,172.20.1.1,80,8080"

```
 Redirect TCP connections
FW_REDIRECT_TCP=""
 Redirect UDP connections
FW_REDIRECT_UDP=""
```

16. Which log level should be forced?

    You can decide if accepted or denied packets are to be logged. You can also specify the log level, "critical", or all. Note that **`*_ALL`** should only be activated for debugging purposes.

    Options for the variable: "yes" or "no", **`FW_LOG_*_CRIT`** default is "yes".

17. Do you want additional kernel TCP/IP security features? If you set this variable to "yes", the following kernel options will be set:

```
# Log critical denied network packets
FW_LOG_DENY_CRIT="yes"
# Log all denied packets
FW_LOG_DENY_ALL="no"
# Log critical accepted packets
FW_LOG_ACCEPT_CRIT="yes"
# Log all accepted packets
FW_LOG_ACCEPT_ALL="no"
```

```
icmp_ignore_bogus_error_responses,
icmp_echoreply_rate,
icmp_destunreach_rate,
icmp_paramprob_rate,
icmp_timeexeed_rate,
ip_local_port_range,
log_martians,
mc_forwarding,
rp_filter,
routing flush
```

Options: "yes" or "no", default is "yes":
`FW_KERNEL_SECURITY="yes"`

18. Should the routing remain active if the packet filter rules are reset? For this, you will need **`FW_ROUTE`**.

    If you use “diald” or “dial-on-demand” via ISDN and IP packets are to be sent over the Internet, you will have to activate this function. Then, masquerading and routing will not be deactivated when the script terminates. You might need this functionality if you have DMZ. Note, however, that this is **not** secure! If you unload the IP filter rules but still have the connection, your network could be vulnerable to attacks.

    Options: "yes" or "no", default setting: "no"

    `FW_STOP_KEEP_ROUTING_STATE="no"`

19. Should ICMP echo pings to the firewall host or to the DMZ be allowed? The **`FW_ROUTE`** is required for **`FW_ALLOW_PING_DMZ`**

    Option: "yes" or "no", default is "no"

```
# Allow ping to firewall
FW_ALLOW_PING_FW="yes"
# Allow ping to DMZ host.
FW_ALLOW_PING_DMZ="yes"
```

Creating your own packet filter with ipchains

If you do not want to use the SuSE Firewall Script, you will also have the option of configuring your own packet filter with ipchains. You can adjust the filter rules s to meet your particular need on a Linux host such as the SuSE Firewall Adminhost and save this packet filter setup with ipchains-save to a file.

```
root@adminhost # ipchains-save > myfiltersetup 2>/dev/null
```

Using FAS, you can then load this file along with your packet filter configuration to the “SuSE Linux Live CD for Firewall” configuration. These filter rules will be saved to the `/etc/ipchainsrc` file on the configuration disk and loaded by the `/sbin/init.d/ipchains` script when booting the firewall host.

4.3.2 DNS

The name server BIND Version 8 is used to enable name resolution over the firewall. If you wish to learn more about DNS, please read the chapter in the appendix of this manual. BIND is configured as forwarding/caching-only server. That means, all requests will be passed on to the forwarders.

4.3.3 MAIL

postfix

A more secure, quicker and more flexible modular mail transport agent which is utilized on the Live CD as mail relay. Note that a built-in hard disk is absolutely essential for using the mail relay function, because postfix first has to route the incoming e-mails to a buffer.

4.3.4 HTTP proxy

The “SuSE Linux Live CD for Firewall” utilizes a multitude of proxies to enable the most refined access control to HTTP/HTTPS services as possible. Two separate proxy branches are used to govern internal access to the outside and external access to inside networks, respectively.

squid

Squid is the “http-proxy” per se and offers extensive configuration options as well as access controls for clients to the WWW by way of ACLs (engl. *access control lists*).

The internal HTTP proxy Squid can be configured to be both transparent as well as non-transparent. In *non-transparent* mode, the following protocols are supported:

- http
- https
- FTP

The prerequisite for this is that the clients (web browser) likewise enters this as a proxy in its settings.

Note

Squid can, however, only process HTTP transparently! Transparent HTTPS or FTP do not function with Squid.
There is currently no solution available for transparent HTTPS apart from being guided straight through the firewall.
The proxy suite can be used for transparent FTP (internal FTP in the FAS).

More exact information on Squid can be found in the appendix.

httpf, tinyproxy

The application httpf is responsible for "content filtering". It is not actually a proxy, as proxy functionality is furnished by the program tinyproxy.

It is more specifically a filtering proxy which can prevent downloading and executing program code. This occurs in that only known and benign language elements are forwarded to the web browser. Even the HTTP header entries can be filtered so that, for instance, no information can be posted to the server via the client host's operating system.

The configuration can be generated using FAS on the adminhost.

transproxy

transproxy governs external access to an internal web server.

Altogether, these applications offer the option of multi-faceted filtering: ACLs for clients, servers, ban lists, content filtering, web caching.

4.3.5 FTP proxy

suse-ftp-proxy

FTP service is split up into two channels. First: setting up the internal connection to the outside and secondly: setting up the external connection to the inside/DMZ.

Internal to external:

magic user

Enables the user name, the host, and the destination FTP server's port to be provided (e. g. `user@ftp.firma.com:2345`). So, automatic execution of the USER command.

default-ftp-server

External to internal:

default-ftp-server

4.3.6 ssh

openssh enables usage of a shell on a remote host where the connection is encrypted.

4.3.7 chroot, secumod, compartment, kernel capabilities

To raise the security level on the firewall, run the services on the SuSE Live CD for Firewall in a `chroot` environment. The “secumod” kernel module and “compartment” will also be implemented. Setting capability bits in the kernel increases the security of the system applications.

chroot With chroot, an application can change its view to the filesystem irrevocably by defining a new "root" for the filesystem. As soon as the application has applied itself to this segment of the filesystem, this segment will adopt the role of the entire filesystem for this application. The rest of the filesystem no longer exists as far as this application is concerned. Even if the program has somehow crashed, the hacker would remain in this chroot() environment and thus not be able to damage the actual system.

secumod Special kernel module which raised the system security level. This involves defining

- the trusted path,
- adding a ban on hard links to files not belonging to the user,
- protecting the proc filesystem,
- not tracing symlinks if they do not correspond to the process currently running (exception: the process is running under `’root’` permissions).
- protecting fifos,
- syscall table checking,
- logging,
- setting capabilities.

compartment Enables execution of applications/services in chroot jails, with unprivileged users/groups. It supports scripts which are called up before the program actually starts (e. g. in order to set up a chroot() environment.) Supports the usage of kernel capabilities.

kernel caps Kernel capability bits

Increasing security by limiting the capabilities of executable programs. The capabilities are documented in

`/usr/include/linux/capability.h`

Using the application compartment is another relatively simply option for specifying the capabilities for an application.

4.4 The Configuration Disk

The configuration disk contains the complete system and application level gateway configurations.

The configuration disk must be accommodated with an ext2 filesystem and obtain the label "SuSE-FWFloppy" via the following command:

Insert a disk formatted with an ext2 filesystem.

```
/sbin/e2label/dev/fd0SuSE-FWFloppy
```

The configuration disk will not be recognized without this label. The FAS (Firewall Administration System) on the SuSE Linux Firewall Adminhost will automatically generate the filesystem and the label.

The configuration disk will be read as the Live CD is being booted.

4.4.1 Creating the configuration disk

The configuration disk can be generated on the SuSE Linux Firewall Adminhost using the administration desktop FAS. The FAS on the SuSE Linux Firewall Adminhost will generate a tar archive which will then be decompressed on the disk. This is the recommended procedure. Changing the configuration files with an editor should only be left to experts who really know what they are doing. No support is offered for manual configuration.

4.4.2 The configuration files

The following overview of the configuration files located on the disk is for information purposes only; the files will already be generated by the admin desktop:

/etc/hosts (see man hosts)
/etc/hosts.allow (see man 5 hosts_access)
/etc/hosts.deny (see man 5 hosts_access)
/etc/inittab (man inittab)
/etc/isdn/
/etc/cipe/
/etc/live-setup.conf
/etc/passwd
/etc/ppp/
/etc/localtime
/etc/modules.boot

In the `/etc/modules.boot` file, you can specify loadable kernel modules to be started along with the system. They will be listed with their relative pathnames in the `/lib/modules/<kernelversion>/` directory. If necessary, additional options can be posted to the module such as irq or the IO addresses of the hardware used. Lines which begin with "#" will be ignored. If a "-" precedes the module name, an attempt will be made to unload the module (this may be necessary if the automatic hardware recognition tries to load the wrong network module, for instance).

Example:

```
# In this file you can provide a list of modules
# and options, that have to be loaded on system boot.
# Module name can be used relative, i.e. net/ppp.o.
#
# Module name [options]
#
#  z. B.:
#misc/cipcb.o
#net/slhc.o
#net/ppp.o
#net/ppp_deflate.o
#net/ppp_mppe.o
-net/de4x5.o   # This module will be removed
 net/tulip.o    # This module will be loaded
```

/etc/named/ contains the zone files of the name server.

/etc/named/master/ contains the master zone files.

/etc/named/slave/ contains the slave zone files.

/etc/named/root.hint contains the addresses of the root name server.

/etc/named.conf responsible for the configuration of named.

/etc/pam.d/ this directory contains the PAM (Pluggable Authentication Module) configuration files

/etc/permissions.local sets the access permission of and to applications, files etc.

/etc/postfix configuration directory for postfix; the major configuration file are:

- `/etc/postfix/master.cf`,
- `/etc/postfix/main.cf`,
- `/etc/postfix/virtual`,
- `/etc/postfix/transport`,
- `/etc/postfix/access`.

/etc/proxy-suite/ configuration directory for FTP proxies,

/etc/rc.config SuSE Linux central configuration file;

/etc/rc.config.d/ contains further configuration files which are read by SuSEconfig;

/etc/rc.config.d/firewall.rc.config configuration file for the SuSE Firewall Script. It is generated by the FAS on the SuSE Linux Adminhost for Firewall.

/etc/resolv.conf configuration for the resolver library, list of name server as well as search list.

/etc/route.conf file containing the information related to generating the static kernel routing table.

/etc/runlevel.firewall This file is responsible for consigning scripts/applications to firewall runlevels. If a script is to be started in a certain runlevel, it is to be entered in the corresponding column in this file.

/etc/securetty list of the ttys where the user `'root'` can log on

/etc/shadow Contains the encrypted passwords. Normally, these are listed in the form of a "*" for each user, that is, no login is possible. The deciphered passwords are only available for user `'root'` to see. In FAS, you do not even need to give a password for `'root'` in which case access to the firewall would only be possible via ssh and RSA key.

/etc/squid.conf configuration file for the HTTP proxy Squid

/etc/ssh/ Contains the configuration files for openssh: `ssh_config` and `sshd_config`.

/etc/su1.priv Configuration file for the **`su1`** command that permits selected users to run programs with under uid=0 (i.e. as `'root'`).

/etc/syslog.conf Configuration of the syslog daemon. To this end, read the following manpages: **`man 5 syslog.conf`**, **`man 8 syslogd`** and **`man 3 syslog`**.

The loghost and the messages to be logged are entered in this file. The entry for the loghost should appear as follows:

```
*.*   @hostname.domain.tl  (or the IP address)
```

/etc/syslog.socks The log daemon `syslogn` has to create a writable socket for all services started in the `chroot` environment; this process is associated with the following files:

```
/var/named/dev/log,
/var/squid/dev/log,
/var/chroot/rinetd/dev/log,
/var/chroot/ftp-intern/dev/log,
/var/chroot/ftp-extern/dev/log
```

/sbin/, /sbin/init.d/ This directory contains self-written init scripts such as IP packet filter scripts.

/root/, /root/.ssh/, /root/.ssh/authorized_keys This file contains the rsa public key of the fwadmin user on the SuSE Linux Firewall Adminhost.

This enables the user `'fwadmin'` to log in on the firewall as `'root'`. RSA keys, which are protected by yet an additional pass phrase, handle user authentication. This pass phrase is generated during the installation of the adminhost by way of the YaST2 module "Firewall Admin Host".

You can also generate the RSA keys at the command line using the command **`ssh-keygen(1)`**. You can read more about it in the manpage (**`man ssh-keygen`**).

It is highly recommended to use the administration desktop FAS on the SuSE Firewall Adminhost for creating configuration files.

4.5 Boot Parameters

You can give boot parameters at the linuxrc boot prompt as usual (see reference manual, kernel boot parameters).

The following restrictions apply:

Due to security reasons, `init=/bin/bash` or the like cannot be entered at the boot prompt for obtaining privileged permissions in a shell.

Boot parameters beginning with "init=" will be ignored.

5 Implementing the Firewall

Requirements for successful implementation:

You have manually created a configuration for the SuSE Linux Live CD for Firewall by way of the SuSE Firewall Adminhost, or a configuration floppy. Or you have enlisted the services of the SuSE Linux Solutions AG for setting up a firewall and you now want to begin operating the firewall host. Starting up the firewall should proceed in several steps. First, check to see if your host boots using the configuration you have set up, and also to see if the selected services start and are available for use. Next, check to see if the IP filter of the kernel is working the way you have configured it.

5.1 Booting the Firewall Host

1. Start the host and then open the BIOS setup program. Check the settings for the time and date.
2. Configure the boot sequence so that the host boots from the CD first – if possible, only from the CD.
3. You must assign a BIOS password in order to prevent, for instance, changes to the boot sequence, or to prevent the firewall from being booted from disk.
4. Save the BIOS configuration.
5. Insert the SuSE Linux Live CD for Firewall.
6. Insert the configuration floppy.
7. Reboot the host.
8. Note any errors which may appear when booting and revise the respective configuration files if need be. If services do not start (message: "<Service xyz> failed/skipped"), the corresponding configuration file for the service <xyz> contains an error. If this is the case, recreate the configuration disk using the FAS tool on the adminhost or revise the configuration saved there and rewrite this to the configuration disk.

You have, of course, already defined the configuration of the hard disk using FAS, but if the partitions 1 (swap) and 2 (var) of the firewall host do not correspond to the settings in the `/etc/live-setup.conf` file, a dialog screen will appear. If you want to reconfigure the hard disk, answer '`yes`'.

5.2 Testing the Firewall

Before employing the firewall for daily use, testing should be done to ensure that, first of all, the packet filter is configured correctly and that second of all, the (configured) proxies all start and requests are processed properly. Adminhost tools are available on the SuSE Firewall for these purposes (see 3 on page 17), such as nmap, nessus, xlogmaster, logsurfer, http clients etc. Detailed documentation for these programs is available in the `/usr/share/doc/packages/` directory on the SuSE Firewall Adminhost. Manpages are also available for each program.

Only when every single test has been successfully completed will you be able to start up the firewall. Please document all tests you carry out.

5.3 Operation

To operate your SuSE Linux firewall, first connect the internal network only to the firewall and furnish yourself a laptop to be used as an external network. Then, close the connection to the internal network and establish a connection to the Internet. If possible, test your firewall externally.

Check your setup. Test EVERYTHING!

5.3.1 Internal testing

Tests you should carry out:

- Are all services available?
- Test if the permitted services are working from the internal clients. Can you access http/s, send e-mails, transfer data via FTP?
- Are your restrictions effective?
- Test your packet filter. You can do this with a port scanner such as nmap. Follow the log messages of your firewall on the loghost or on your firewall itself. Let a packet sniffer run simultaneously to detect restricted packets or to see if response packets are not being sent.
- Are the log files being written to the loghost?

Fixing errors:

Determine the source of the problem. Search for the logfiles according to process name, e. g. postfix or named:

```
# grep postfix /var/log/messages
```

or

```
# grep named /var/log/messages
```

Many programs can be switched to “verbose mode”. In this way, you can obtain detailed information (which can, however, be quite extensive . . .).

5.3.2 External testing

Test externally to see if the available services are working. For instance, you can send e-mails to the internal network. You should be able to see the postfix messages in `/var/log/mail` on the firewall host, as to whether the e-mails were accepted and were able to be delivered to the internal mail server. Check to see if the packet filter is working. This can be verified by a port scanner. At the same time, you will find the kernel packet filter messages in `/var/log/messages` as well as in the log directories of the loghost. These messages will be reflected in the "DENY" and "ACCEPT" messages. Try to set up connections to explicitly restricted ports and attempt to find the corresponding log entries and match them to their corresponding events.

If you are using a loghost, check to see if the log messages are being transmitted in their entirety.

5.3.3 “Really” going online

Only after you have completed all these tests, connect the firewall host to the Internet and to your Intranet and begin productive operation.

Note

Constantly monitor your log files. This is the only way you can ensure a timely response to attacks or failures. If unusual events occur, react immediately. Raise the log level (if possible) and refine your log analysis.

6 Help

In this chapter, you will find information as to how you can create a setup concept for a firewall solution in your network using the “SuSE Linux Firewall on CD”. You also have the option of making use of the services of the SuSE Linux Solutions AG to have a plan drawn up which is tailored to your specific needs.

6.1 Troubleshooting

You will find help here if the adminhost is not able to be installed or if the SuSE Linux Live CD for Firewall is not booting.

6.1.1 Problems Installing the Adminhost

If you have trouble installing the SuSE Linux Firewall Adminhost, SuSE Linux Installation Support is available to you free of charge. First, it is worth your while to take a glance at the support database. Here, you will find a realm of information on a variety of installation problems – a keyword search will help you here:

`http://sdb.suse.de`

6.1.2 Problems Booting the Live CD

To simplify your troubleshooting, you should observe and note the following:

- What happens when you boot?
- Are services not being started which have already been configured (skipped/failed messages)?
- Do error messages appear on the console?
- Are there kernel error messages on tty9/tty12? To see these messages , change to this console using `Alt` + `F9` or `Alt` + `F12`.

Problems integrating the network:

Observe and note the nature of the problems. For example, on the firewall, test whether the network interfaces are configured correctly (log in to the firewall as user `'root'`):

```
root@firewall: # ifconfig
```

This command will show you which interfaces are configured and which IP addresses, network masks, etc. are set up. If necessary, correct the configuration using FAS and create a new configuration floppy. The host will have to be restarted to commit the new configuration.

- Which services function from the client side (Intranet)? Should they be functioning? The log files can help you determine whether it is a problem involving unauthorized access.

  Try reproducing the errors.

- Is external access to available resources not functioning? Which services are affected? Should the resources really be accessible?

  Test using ps(1) whether the process is available; that is, whether the process is accessible. If services are not accessible, check your logfiles. Here you will find messages about why a particular service was not started or whether an unauthorized really has been attempting to make use of the service.

  Test from several clients whether the firewall host is accessible and whether the proxies are responding.

6.2 Security Policy and Communication Analysis

To ensure that the internal network's connection to the Internet (or to any other "unprotected" network) is secure, a few things will need to be clarified first. This includes outlining a security policy for your own network as well as undertaking a communication analysis.

6.2.1 Security Policy

The security policy provides the basis for working with all programs, hosts and data. In addition, it outlines how to guarantee the monitoring of security guidelines and how (internal or external) breaches of this policy are to be handled. In order to draft a security policy, it is best to produce a communication analysis. To this end, the following topics are of utmost importance:

- An analysis of your security requirements is necessary. What needs to be protected?
- Are there areas of the Intranet which contain especially sensitive data (e. g. personell department, data critical of the company etc.). Where is this data located?
- Who can access the data? Are there various levels of authorization?
- Should data be available over the network?
- Which services should be accessible internally? Which services should be accessible from internal to external (e-mail, surfing, data transfer) and which services, external to internal (e-mail, web services, data transfer)?

The list of questions to be clarified for a security policy must be drawn up individually and answered.

6.2.2 Communication Analysis

The most important aid for carrying out a communication analysis is a communication matrix.

The services available for client hosts and users are represented here in a table format. This matrix is then mapped to the proxies/IP filter rules.

Setting up a communication matrix:

Make a list of all clients/servers on your network. Then define which protocols may be used by which clients. Also, state in which direction each packet can be sent or received.

An example for HTTP protocol: The client host1 needs to access a web server in the internal network, but should not be able to establish a connection to an external web server. The entry in the communication matrix will then appear as depicted in the example shown below.

Example of a communication matrix:

Protocol	icmp		ftp		ssh		smtp		http		https		. . .
Client	internal	external	i.	e.	i.	e.	i.	e.	i.	e.	i.	e.	
host1	x	–	x	–	–	–	x	–	x	–	–		
host2	x	–	–	–	x	–	x	–	–	–	–	–	
host3													
. . .													
hostn													

With the help of such a communication matrix, you will always obtain an overview of the communication constellations within the network. This will not only simplify the configuration of your network, but also error analysis.

6.3 Services Offered by SuSE Linux Solutions AG

SuSE Linux Solutions AG offers its services to you in the areas of:

- Consulting:
  Clarification of security needs, communication analysis
- Planning:
  - Hardware requirements
  - Network planning/integration

- Installation:
  - On-site installation of the firewall solution
  - Conducting tests on-site/remote
  - Documentation
- Maintenance:
  - Business support (available at cost)
  - Online maintenance via modem/call-back (by consulting/business support)

6.4 Updates

One of the most important aspects of operating a firewall is constant maintenance of the implemented software. As soon as a security gap is discovered and remedied, the respective software must be updated.

If such a security update affects the "SuSE Linux Live CD for Firewall", a new Live CD will be sent to you. This update CD can simply be inserted into the CD drive of the firewall host and the firewall then restarted.

Security-related updates for the SuSE Linux Adminhost for Firewall will be made available on the SuSE FTP server. This package can be loaded to the adminhost and installed by giving a command as exemplified below:

```
root@adminhost:# rpm -Uhv newpackage.rpm
```

Also, please follow the instructions posted on the security mailing lists and the SuSE web server:

`http://lists2.suse.com/archive/suse-security`

`http://www.suse.de/de/support/security/index.html`

6.5 What to Do If . . .

6.5.1 Intrusion Detection and Event Display

A “properly” configured Linux/UNIX system can, in and of itself, be considered quite secure. System-immanent hazards which are associated with a complex system such as Linux or UNIX are more easily recognized than on other operating systems because UNIX has been used and further developed for over 30 years. UNIX also forms the basis of the Internet. Nevertheless, configuration errors can occur and security holes can turn up time and again. There will always be security flaws. Security experts and hackers are in perpetual “competition” to be one step ahead of the other. What qualifies today as secure may already be vulnerable tomorrow.

Signs of intrusion into your system

Any abnormal behavior on your firewall system can be considered a sign that a you system is being compromised, e.g.:

- increased processor load,
- unusually heavy network traffic,
- unusual processes or
- processes are being started by non-existent users.

Indications in the system logfiles, in the system configuration – How can an intrusion be recognized?

First, you should be clear as to which actions are to be defined as intrusions. Unfortunately, it is normal these days that a host connected to the Internet will be scanned for open and vulnerable ports. It is just as common that ports having been recognized, somewhere along the way, as being vulnerable (pop3/qpopper, rpc-mountd, smtp/sendmail) are attacked. Most of these attacks are carried out by “script kids”. Pre-fabricated “exploits” which are published on relevant web sites (e. g. `http://www.rootshell.com`) are used. These web pages also exist as valuable information sources for network and system administrators. They can usually be identified by the fact that the hack is only carried out once and is not repeated if the action fails in the first attempt. The first measure you should take in case of such an event is to check that a break-in really did happen. Furthermore, you should raise the log level and refine the evaluation (e. g. a selective search aimed at an intruding IP address in the logfiles or any unusual port numbers).

What you consider to be a dangerous attack on your system remains up to your discretion. But at least you should determine how you are going to react to such an event.

What to do if you suspect an intrusion? The following literature is recommended for such cases:

“Steps for Recovering from a Unix Root Compromise” (`http://www.cert.org/tech_tips/root_compromise.html`)

RFC 2196

Site Security Handbook

Both publications describe procedures which should follow any successful break-in. The documents provide a formal starting point as to how a company, agency or educational institution may react. The procedure discussed there necessitates a certain amount of memory to be able to take “snapshots” of the system, as well as requires colleagues to carry out the analysis and to examine the security problem. Situations are described which may necessitate that punitive action be taken.

What to do if you suspect a successful attack:

- It is important to stay calm. Hasty actions can destroy important information (e. g. processes initiated by the hacker).
- The course of action as well as who to inform should be outlined in the security policy. The communication should not take place by e-mail, but by telephone or by fax.
- Physically cut the network connection to the firewall. It is not a good idea to shut down the computer. Important information could be lost, e. g. programs which were manually started by the hacker.
- List all running processes with `ps` and search for processes which do not typically occur in normal firewall operation.
- Draw up a process table when setting up the firewall which can later serve as a basis for comparison.
- Check the running processes for ties to unusual TCP or UDP ports.
- Were the packet filter rules changed?
- Compare all configuration files with the original configuration. This is very easy with the “SuSE Linux Firewall on CD”, since the configuration is already saved on the adminhost. In doing so, however, it is essential not to restart the firewall host, since otherwise, any changes to the filter rules could be lost!
- Also back up all logfiles. The logfiles could be legally admissible evidence. Document all steps you have taken. If you save the logfiles locally to the hard disk, you should make a 1:1 copy of the hard disk for documentation purposes.
- Save your data to CD or to another medium (tape drive, ZIP drive).
- Analyze the logfiles: Who tried when and from where (IP address, domain name, possibly even username) to access which services/ports? Was an attempt made to uncover passwords (multiple failed login attempts with the same user name)?

- Make sure that you are using a uniform and exact time source. It is important that the hardware clock of the firewall host and the log host are as closely in sync as possible. Use a common time source for all your servers whenever possible. Only in this way can events be kept track of accurately.
- Which parts of the security policy were violated? This is especially important in regard to internal intrusions.
- You may want to deactivate the user account in your network, the account which carried out the attack. The relevant procedure pertaining to internal violations should also be regulated (security or company policy).

6.5.2 External Attacks

Inform the system administrator responsible for the address block (via postmaster or the domain's abuse address).

The question is raised as to what information needs to be passed on. The report of an incident or attack should contain enough information to ensure that the other party can investigate the problem. However, consider that your contact person could be the one who has carried out the attack. Here is a list of possible information you can provide. You can decide which of the following pieces of information you want to give:

- Your e-mail address,
- Telephone number
- Your IP address, hostname, domain name,
- the IP addresses, hostnames, domain names affected by the hacking incident
- the date and time of the intrusion, preferably, with the time zones in relation to the GMT,
- a description of the attack,
- how the attack was recognized,
- excerpts of the logfiles relating to the attack,
- a description of the logfile format,
- statement of advisories and security information which describes the nature and severity of the attack,
- what you want the contact person to do: Close an account, confirm the occurence of an attack, issue a report for information purposes only, request for further observation.

Once you have gone through all the data security and documentation procedures, set up your firewall again. Raise the log level of each application if possible. You can be quite sure that the hacker will try to infiltrate your system again. This will be the opportunity to catch the intruder red-handed.

Examples:

Log in to the console.

Examine the logfiles.

For example:

```
# grep DENY /var/log/messages | less
```

With this command you will see all the lines contained in a `DENY`, that is, the lines which were listed by the kernel's packet filter. For instance, you can now search for certain unusual IP addresses (frequently occurring DENY messages which correspond to IP addresses on one or more port numbers). Find out exactly what happened. Using the tools previously mentioned, examine the logfiles according to definable criteria. Problem: Sometimes it is unclear what to look for until you find it ... It is also possible that a system administrator from another network is reporting a complaint, via postmaster or abuse mail address, that attacks have been occurring from your network to remote hosts. Take such complaints seriously. Request that logs of all intrusions are sent so that you may have a clue as to the date and time as well as the method of attack on the external system. Attempt to verify the circumstances surrounding the incident. A hacker, intruder or attacker may have already overstepped the security boundaries and misused your network for his intentions. In respect to this, the reputation of your business likewise hangs in the balance.

Once you have backed everything up and documented it, review your firewall configuration. After remedying possible errors or after shutting down services you deem at risk, restart your firewall. This shows the clear advantage of the "SuSE Linux Firewall on CD": The original status of the firewall will be restored simply by rebooting the firewall host, making a complicated reinstallation of the operating system and bringing in the backups wholly unnecessary.

6.5.3 Advantage of the Live Filesystem of the "SuSE Linux Firewall on CD"

One of the greatest advantages of the SuSE Firewall is that its initial state can be restored simply by booting it. In any case, you should, of course, keep in mind that any possible configuration errors may reappear.

If the firewall has been breached, the method of intrusion should be investigated in order to be able to correct configuration errors. If security holes are found in applications, SuSE Linux AG provides updates for the affected applications – meaning in the case of the "SuSE Linux Live CD for Firewall", a new CD.

You can find more information here:

`http://www.cert.org` and

`http://www.first.org`

6.6 Professional Help and Support

If you suspect an attack, but are uncertain as to whether the attack has led to damage to your network, despite having carried out the review procedures as described above, you should definitely introduce initial security measures. That means, first and foremost, physically disconnecting the network .

Direct your inquiries to the vendor where you purchased the SuSE Firewall on CD or consult SuSE directly. In such cases, you can also call the professional support at SuSE Linux GmbH or enlist the services of SuSE Linux Solutions AG.

6.6.1 Support Regulations

Installation support coverage

The installation support is there to assist you in installing your SuSE Linux System, ready for operation. This also applies to the central system components which allow for fundamental operation. It includes:

- Support ranging from the installation of the SuSE Linux base system on a host from the “Admin CD for Firewall” to the successful startup of the “Firewall Administration Systems” (FAS).
- The configuration of the basis hardware of this host with the graphical installation tool YaST2, i.e. the central components of the PC, excluding peripheral devices but including the configuration of an ethernet card.

All the topics not mentioned here are not covered by installation support.

Time period of installation support

The installation support for the Admin CD for Firewall lasts for a period of 30 days following the registration date.

How do I reach the SuSE Support Team?

You can reach our support team by e-mail, fax or postal mail:

- **E-mail:**

Address:	fw-support@suse.de
Processing:	weekdays

- **Fax:**

Fax number:	`(09 11) 74 05 34 77`
Processing:	weekdays

- **Postal mail:**

Address:	SuSE GmbH – Support – Schanzäckerstr. 10 D-90443 Nürnberg
Processing:	weekdays

6.6.2 Professional Services

Even if an operating system comes with all the necessary facilities: it will only be a viable alternative for use in the corporate environment in combination with professional and qualified support services. SuSE guarantees this kind of service for Linux. All information on this can be found at the central support portal for SuSE Linux:

`http://www.suse.de/en/support/`

Individual Projects and Consulting

You'd like to use SuSE Linux at your place of business. We offer competent consultation and solutions enabling you to optimize the performance of Linux in your field of computing.

We have a large amount of experience in the deployment of Linux servers because we've been dealing with Linux since the early days. This is where the experience of our consultants comes into play; you can use the know-how of our experts to successfully realize your projects. Our strength lies in our versatility; Databases, security issues, Internet connection or company-wide networks, Linux is, with the right software, a powerful platform for your applications.

Our services range from the conception, implementation and configuration of server systems to a complete infrastructure consultation.

You may want, for instance, to implement your Internet presence on a SuSE Linux basis and are therefore looking for web server, e-mail and secure server solutions? Our consultants can help you to conceptualize and implement the right solution.

Are you managing a complex heterogenous network in which you'd like to integrate Linux? We offer consultation and support for the design and rollout of complex server solutions.

Do you have special requirements or needs that aren't fulfilled by standard software? We can assist you further in individualized system development.

On-site support is provided by the company *SuSE Linux Solutions AG*:

SuSE Linux Solutions AG
Mergenthalerallee 45-47
D-65760 Eschborn
Tel: + 49 / 6196 / 50 95 10
Fax: + 49 / 6196 / 40 96 07
E-Mail: solutions@suse.de

Represented by our Regional Service Centers in Hamburg, Berlin, Bonn, Stuttgart, Frankfurt, Munich and Nuremberg, as well as our Support and Development Center in Nuremberg.

- Rollout and Implementation Services
- Infrastructure Consultation
- Intranet Server Solutions
- Internet Server Solutions
- Development of Client-specific Requirements
- Complete Solutions
- E-Commerce

6.6.3 Training

Our specialists train system administrators and programmers in such a way that they are able to make use of the wide range of possibilities in Linux as quickly as possible – and thus be able to work productively. For more information on a training courses, have a look at `http://www.suse.de/en/support/training/`.

6.6.4 Feedback

We always appreciate your tips, hints and problem descriptions. We will help you if your problem is a straightforward one, or if we already have the solution at hand. Your feedback may provide us with useful information to help us avoid this problem in our next release, thus helping other SuSE Linux customers via our WWW server or the Support DataBase.

We make every effort to construct a SuSE Linux system which meets the wishes of our customers as closely as possible. We therefore much appreciate any criticism of our CD or of this book. We think that this the best way to correct significant errors and to maintain our high standard of quality.

Send feedback any time to feedback@suse.de via electronic mail, or you can send us a letter or fax.

6.6.5 Further Services

We would also like to draw your attention to our services that are available around the clock, free of charge:

- **SuSE WWW Server**

  `http://www.suse.com`

  Up-to-date information, catalogs, ordering service, support form, support database

- **SuSE Mailing Lists** (information and discussions via electronic mail):
  - suse-announce@suse.com – announcements concerning SuSE GmbH (German)
  - suse-announce-e@suse.com – announcements concerning SuSE GmbH (English)
  - suse-linux@suse.com – Discussions concerning the SuSE Linux Distribution (German)
  - suse-linux-e@suse.com – Discussions all about SuSE Linux (English)
  - proxy-suite@suse.com – Discussion concerning the SuSE proxy suite (English)
  - suse-adabas@suse.com – Information and discussion concerning Adabas-D in SuSE Linux (German)
  - suse-applix@suse.com – Discussions concerning the **Applixware** package of SuSE GmbH (German)
  - suse-axp@suse.com – SuSE Linux on alpha processors (English)
  - suse-domino@suse.com – Information and discussion concerning SuSE Linux and Lotus Domino (German)
  - suse-ham@suse.com – SuSE Linux and amateur radio (German)
  - suse-ham-e@suse.com – SuSE Linux and amateur radio (English)
  - suse-ibm-db2@suse.com – SuSE Linux and IBM DB2 (English)
  - suse-isdn@suse.com – ISDN with SuSE Linux (German)
  - suse-informix@suse.com – Information and discussion concerning Informix in SuSE Linux (English)
  - suse-laptop@suse.com – SuSE Linux on laptops (German)
  - suse-motif@suse.com – SuSE Linux on Motif (English)
  - suse-oracle@suse.com – Information and discussion concerning Oracle in SuSE Linux (English)
  - suse-ppc@suse.com – SuSE Linux on Power PC processors (English)
  - suse-security@suse.com – Discussion of security issues in SuSE Linux (English)
  - suse-security-announce@suse.com – Announcement of security-related errors and updates (English)
  - suse-sparc@suse.com – SuSE Linux on Sparc processors (English)

  To subscribe to a list, send an electronic mail message to:

  <LISTNAME>`-subscribe@suse.com`

  An automatic response will be sent back which you will need to confirm.

  For <LISTNAME>, substitute the name of the mailing list you want to subscribe to; e. g.suse-announce-subscribe@suse.com, to receive regular announcements.

  The procedure is similar for unsubscribing from a list:

<LISTNAME>`-unsubscribe@suse.com`

Please be sure to send the **`unsubscribe`** mail with the proper electronic mail address.

- **SuSE FTP Server**

  `ftp://ftp.suse.com`

  current information, updates and bug fixes

  Log in to the system as user `‘ftp’`.

6.7 Bibliography

- D. Brent Chapman & Elizabeth D. Zwicky: *Einrichten von Internet-Firewalls*, O’Reilly 2000.
- Wolfgang Barth: *SuSE Firewall Book*, SuSE Press 2001.
- *Maximum Linux Security*, SAMS 1999.
- Robert L. Ziegler: *Linux Firewalls*, New Riders 1999.

A DNS – Domain Name System

DNS (Domain Name System) is needed to resolve the domain and host names into IP addresses. In this way, the IP address `192.168.0.20` is assigned to the host name `earth`, for example.

A.1 Starting the Name Server BIND

The name server BIND8, as well as the new version BIND9, is already preconfigured in SuSE Linux, so that you can easily start it right after you have installed the Linux distribution.

If you already have a functional Internet connection and have entered 127.0.0.1 as name server for the localhost in `/etc/resolv.conf`, you should normally already have a working name resolution without having to know the DNS of the provider. BIND thus carries out the name resolution via the root name server, a notably slower process. Normally, the DNS of the provider should be entered with its IP address in the configuration file /etc/named.conf under **`forwarders`** in order to ensure effective and secure name resolution. If this works so far, the name server will run as a pure “caching-only” name server. Only when you configure its own zones will it become a proper DNS. A simple example of this can be found in the doc directory under `/usr/share/doc/packages/bind8/sample-config`. However, you should not set up any official domains until you have actually been assigned one from the institution responsible for ’.de’ it is the DENIC eG, for instance. Even if you have your own domain and it is managed by the provider, you are better off not to use it, as BIND would otherwise not forward any more requests for this domain and thus the provider’s webserver, for example, would not be accessible for this domain.

In order to start the name server, the following is entered at the command line (as root):

```
rcnamed start
```

If “done” appears to the right in green, this means `named`, as the name server process is called, has been started successfully. You can immediately test the functionality of the name server on the local system with the `nslookup` program. The localhost should show up as the default server with the address 127.0.0.1. If this is not the case, the wrong name server has probably been entered in `/etc/resolv.conf` or this file does not exist. For the first test, you should enter `nslookup` “localhost” or “127.0.0.1” at the prompt, which should always work. If you receive an error message instead, such as “No response from server”, you should check to see if `named` is actually running using the following command

```
rcnamed status
```

If the name server does not start or is exhibiting faulty behavior, you can find the possible causes of this logged in “/var/log/messages”.

If you have a dial-up connection, you will have to be sure that BIND8, once it starts up, will check the root name server. If it does not manage this, because an Internet connection has not been made, this can cause the DNS requests not to be resolved other than for locally defined zones. BIND9 behaves differently, but requires quite considerably more resources than BIND8.

In order to implement the name server of the provider, or one that you already have running on your network as “forwarder”, you must enter one or more of these in the **`options`** section under **`forwarders`**:

```
        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;
                };
```

The IP addresses used in the example are arbitrary and therefore, must be adjusted to your personal environment.

After **`options`** follow the zones. There should at least be the entries for “localhost”, “0.0.127.in-addr.arpa”, and “.”. The entries for “type hint” do not need to be modified, as they function in their present state.

Also, please be sure that a “;” follows each entry and that the braces are properly set.

If you have made changes to the configuration file `/etc/named.conf` or to the zone files, you will have to adjust BIND to reread these files. You can do this by entering **`rcnamed reload`**.

Otherwise, you can completely restart the name server by giving the command **`rcnamed restart`**. If you need to stop the name server, enter: **`rcnamed stop`**.

If `named` should be already started when booting, you will only have to reset the entry in `/etc/rc.config`, **`START_NAMED=no`**, to **`START_NAMED=yes`**.

A.2 The Configuration File /etc/named.conf

All the name server settings BIND8 and BIND9 are to be made in the `/etc/named.conf` file. The zone data itself, however, is the host names, IP addresses, etc. For the domains to be administered, separate files are to be stored in the `/var/named` directory, but more on this in the next chapter.

The `/etc/named.conf` is roughly divided up into two areas: one is the **`options`** section for general settings and the other consists of **`zone`** entries for the individual domains. Additional sections for **`logging`** and **`acl`** type entries can be added. Comment lines begin with a `‘#’` sign or a `‘//’`.

A minimalistic `/etc/named.conf` looks like example A.2.1 on the next page.

```
        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";
        };
```

File contents A.2.1: Example of a named.conf

This example works for both BIND8 and BIND9, since no special options are used which are only understood by one version or the other. BIND9.1.1 accepts all BIND8 configurations and makes note of options not implemented at start-up. Special BIND9 options are, however, not supported by BIND8.

A.2.1 Important configuration options in “options”

directory "/var/named"; specifies the directory where BIND can find the files containing the zone data.

forwarders 10.0.0.1; ; is used to specify the name server(s) (mostly of the provider) to which DNS requests, which cannot be responded to directly, are forwarded.

forward first; causes DNS requests to be forwarded first before an attempt is made to resolve them via the root name server. Instead of **`forward first`**, **`forward only`** can be written to have all requests forwarded and the root name servers will no longer respond. This makes sense for firewall configurations.

listen-on port 53 127.0.0.1; 192.168.0.1; ; tells BIND which network interface and which port it is listening to. The **`port 53`** specification can be left out, since 53 is still the default port. If this entry is completely omitted, usually all interfaces will be implemented

query-source address * port 53; This entry is necessary if a firewall is blocking external DNS requests. This tells BIND to post requests externally from port 53 and not from any of the ports greater than 1024.

allow-query 127.0.0.1; 192.168.1/24; ; defines the networks from which clients can post DNS requests. **`/24`** at the end is an abbreviated expression for the netmask, in this case 255.255.255.0.

allow-transfer ! *; ; controls which hosts can request zone transfers, this example cuts them off completely due to the **`! *`**. Without this entry, zone transfers can be requested anywhere without restrictions.

allow-update ! *; ; this option controls the external write-access to the zone data. This makes it possible for clients to register themselves in the DNS, which is not recommended for security reasons. Without this entry, zone updates are generally prohibited, this example would not change anything, since **`! *`** likewise prohibits any action.

statistics-interval 0; In the absence of this entry, BIND8 generates several lines of statistical information in `/var/log/messages`. Specifying 0 suppresses these completely, otherwise the time in minutes can be given here.

cleaning-interval 720; This option defines at which time intervals BIND8 clears its cache. This activity triggers an entry in /var/log/messages each time it occurs. The time specification is in minutes. The default is 60 minutes.

interface-interval 0; BIND8 regularly searches the network interfaces for new or no longer existing interfaces. If this value is set to **`0`**, this will not be carried out and BIND8 will only listen at the interfaces detected at start-up. Otherwise, the interval can be defined in minutes. The default is 60 minutes.

notify no; no prevents other name servers from being informed when changes are made to the zone data or when the name server is restarted.

A.2.2 The Configuration Section “Logging”

What, how and where archiving takes place can be extensively configured in BIND8. Normally, the default settings should be sufficient. The following example is the simplest form of such an entry and will completely suppress “logging”:

```
logging {

        category default { null; };

};
```

A.2.3 The Structure of Zone Entry

After **`zone`**, the name of the domain to be administered is specified, arbitrarily **`my-domain.de`** followed by **`in`** and a block of relevant options enclosed in curly brackets. If you want to define a “slave zone”, the **`type`** is simply switched to

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

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

`slave` and a name server has to be specified, which administers this zone as **`master`** (but which can also be a "slave").

The options:

type master; **`master`** indicates that this zone is administered on this name server. This assumes that your zone file has been properly created.

type slave; This zone is transferred from another name server. Must be used together with **`masters`**.

type hint; The zone **`.`** of the type **`hint`** is used for specification of the root name servers. This zone definition can be left alone.

file "my-domain.zone" or file "slave/other-domain.zone"; This entry specifies the file where zone data for the domain is located. This file is not required by **`slaves`**, since its contents are read by another name server. In order to differentiate between master and slave files, the directory `slave` is specified for the slave files.

masters { 10.0.0.1; }; This entry is only needed for slave zones. It specifies from which name server the zone file is to be transferred.

A.2.4 Structure of Zone Files

Two types of zone files are needed: one serves to assign IP addresses to host names and the other does the reverse: supplies a given IP address with an IP address.

'.' has an important meaning in the zone files here. If host names are given without ending with a '.', the zone will always be appended. Thus, complete host names specified with a complete domain must end with a '.' so that the domain is not added to it again. A missing point or one in the wrong place is probably the most frequent cause of name server configuration errors.

The first case to consider is the zone file "world.zone" which is responsible for the domain 'world.cosmos'.

Line 1: **`$TTL`** defines the standard TTL which applies for all the entries in this file, here 2 days. TTL stands for "time to live".

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

Line 2: The **`SOA control record`** begins here:

- The name of the domain to be administered is **`world.cosmos`** in the first position. This ends with a '.', since otherwise, the zone would have to be appended a second time. Alternatively, a `'@'` can be entered here. Then, the zone would be extracted from the corresponding entry in `/etc/named.conf`.
- After **`IN SOA`** is the name of the name server in charge as master for this zone. In this case, the name is extended from **`gateway`** to **`gateway.world.cosmos`**, since it does not end with a `'.'`.
- Afterwards, an e-mail address of the person in charge of this name server will follow. Since the `'@'` sign already has a special significance, `'.'` is to be entered here instead, for **`root@world.cosmos`**, consequently **`root.world.cosmos.`**. The `'.'` sign at the end cannot be neglected, otherwise, the zone will still be added here.
- A `'('` follows at the end here, including the following lines up until `')'` into the SOA record.

Line 3: The **`serial number`** is an arbitrary number which is increased each time this file is changed. It is needed to inform the secondary name servers (slave servers) of changes. For this, a ten-digit number of the date and run number, written as YYYYMMDDNN, has become the customary format.

Line 4: The **`refresh rate`** specifies the time interval at which the secondary name servers verify the zone **`serial number`**. In this case, 1 day (1D = 1 day).

Line 5: The **`retry rate`** specifies the time interval, at which a secondary name server, in case of error, attempts to contact the primary server again. Here 2 hours (2H = 2 hours).

Line 6: The **`expiration time`** specifies the time frame after which a secondary name server discards the cached data if it has not regained contact to the primary server. Here, it is a week (1W = 1 week).

Line 7: The **`minimum time to live`** states how long the results of the DNS requests from other servers can be cached before they become invalid and have to be requested again.

Line 9: The **`IN NS`** specifies the name server responsible for this domain. The same is true here, that **`gateway`** is extended to **`gateway.world.cosmos`** because it does not end with a `'.'`. There can be several lines like this, one for the primary and one for each secondary name server. If **`notify`** is not set to **`no`** in `/etc/named.conf`, all the name servers listed here will be informed of the changes made to the zone data.

Line 10: The MX record specifies the mail server which accepts, processes and forwards e-mails for the domain **`world.cosmos`**. In this example, this is the host **`sun.world.cosmos`**. The number in front of the host name is the preference value. If there are multiple MX entries, the mailserver with the smallest value is taken first and, if mail delivery to this server fails, an attempt will be made with the next highest value.

Line 12-17: These are now the actual address records where one or more IP addresses are assigned to the host names. The names are listed here without a **`.`**, since they are entered without a domain added and can all be appended with **`world.cosmos`**. Two IP addresses are assigned to the host **`gateway`**, since it has two network cards.

The pseudo-domain **`in-addr.arpa`** is used to assist the reverse lookup of IP addresses into host names. This will be appended, for this purpose, to the network components described here in reverse order. **`1.168.192.in-addr.arpa`** thus results from **`192.168.1`**.

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

Line 1: $TTL defines the standard TTL which applies to all entries here.

Line 2: 'Reverse lookup' should be activated with this file for the network 192.168.1.0. Since the zone is called '1.168.192.in-addr.arpa' here, it is, of course, undesirable to add this to the host name; therefore, these are all entered complete with domain and ending with '.'. The rest corresponds to the previous example described for world.cosmos.

Line 3-7: See the previous example for world.cosmos.

Line 9: This line also specifies the name server here which is responsible for this zone. This time, however, the name is entered, including the domain and ending with '.'.

Line 11-13: These are the pointer records which point to an IP address at the relevant host name. Only the last part of the IP address is entered here at the beginning of the line, and the last '.' omitted. Now, if the zone is appended to this and the '.in-addr.arpa' is ignored, the entire IP address will be backwards.

In this form, the zone files are usable for both BIND8 and BIND9. Zone transfers between different versions should also not normally be an issue.

A.3 DNS Sample Configuration

In the sample configuration of a name server illustrated here, we will make the following assumptions: Your domain name is world.cosmos, you have a gateway host which establishes the Internet connection and two company internal networks connected to one another. This host will be our name server and will receive the name "gateway". Your network appears as follows:

Network 1 includes the hosts:

- gateway IP 192.168.1.1
- earth IP 192.168.1.2
- mars IP 192.168.1.3

Network 2 includes the hosts:

- gateway IP 192.168.0.1
- sun IP 192.168.0.2
- moon IP 192.168.0.3

The hosts earth and mars are only connected via gateway with sun and moon, just as all hosts can only access the Internet via gateway.

The following files are necessary for configuring the name server:

named.conf the central configuration file,

world.zone contains the host table,

192.168.1.zone for the subnet with the hosts earth and mars,

192.168.0.zone for the subnet with the hosts sun and moon,

localhost.zone contains the IP address of the localhost,

127.0.0.zone loopback,

root.hint contains the Internet root servers.

The `localhost.zone`, `127.0.0.zone` and `root.hint` files are automatically created during installation of BIND 8.

The file `/etc/named.conf` will have to be adjusted, as shown in the file A.3 on the following page.

`acl` is the access control list which defines from which IP addresses access can be made to thed DNS server.

The **`directory`** entry specifies where the other configuration files are located. `/var/named/` is the default.

The process ID of **`named`** is stored in the **`pid-file`**.

`listen-on Port` defines the port from which the name server receives requests.

`zone` entries specifies the configuration files to which IP addresses and host names are assigned to one another. This files will have to be set up in the next step in the `/var/named/` directory.

The complete host table is entered in the `world.zone` file (see file A.3 on page 113). In our example, this looks like the following:

The option '`$TTL`' specifies the "time to live". '`SOA`' stands for "start of authority" and introduces the pre-set record: host name, e-mail address, where the "@" sign is substituted by a ".", serial number (date and double-digit version number) and TTL. '`NS`' marks the name server. '`A`' signals that the IP addresses of the hosts will follow in the domain. The name server "gateway" has two IP addresses, since it belongs to two subnets.

In the next two zone files, the reverse lookup takes place for both subnets.

Now, before you can test your name server, the entry in the `/etc/rc.config` file will have to be set to **`START_NAMED=yes`**. Subsequently, the other hosts can be informed of the IP address of the name server as well, for example, via YaST1 ('`System administration`' -> '`Configure network`' -> '`Configuration name server`').

Now, if you enter **`nslookup earth`** in a console, you should see an output of the name server with IP address as well as the IP address of the host earth. If the name server is not functioning, you will find the cause in the `/var/log/messages` file.

A.4 Further Information

- Documentation on bind8: `file:/usr/share/doc/packages/bind8/html/index.html`.
- A sample configuration can be found at:
  `/usr/share/doc/packages/bind8/sample-config`

```
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 "world.cosmos" in {
        type master;
        file "world.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";
};
```

File contents A.3.1: File named.conf

```
$TTL 2D

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

          IN NS       gateway
          IN MX       10 sun

gateway   IN A        192.168.0.1
          IN A        192.168.1.1
sun     IN A        192.168.0.2
moon      IN A        192.168.0.3
earth      IN A        192.168.1.2
mars      IN A        192.168.1.3
```

File contents A.3.2: File world.zone

```
$TTL 2D

0.168.192.in-addr.arpa. IN SOA gateway.world.cosmos.
                                  root.world.cosmos. (
                               2001040501     ; serial
                               1D             ; refresh
                               2H             ; retry
                               1W             ; expiry
                               2D )           ; minimum

          IN NS                gateway.world.cosmos.

1         IN PTR               gateway.world.cosmos.
2         IN PTR               sun.world.cosmos.
3         IN PTR               moon.world.cosmos.
```

File contents A.3.3: The file 192.168.0.zone

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

          IN NS                gateway.world.cosmos.

1         IN PTR               gateway.world.cosmos.
2         IN PTR               earth.world.cosmos.
3         IN PTR               mars.world.cosmos.
```

File contents A.3.4: The file 192.168.1.zone

- manpage for **`named`** (**`man 8 named`**), in which the relevant RFCs are named along with the manpage for **`named.conf`** (**`man 5 named.conf`**) in particular.

B DHCP

B.1 The DHCP Protocol

The purpose of the “Dynamic Host Configuration Protocol” is to assign network settings centrally from a server rather than to configure them locally on each and every workstation. A client configured to use DHCP doesn’t have control over its own static address but is enabled to fully auto-configure itself in line with what the server side is telling him.

One way to use DHCP is to identify each client using the hardware address of its network card (which is fixed in most cases), and then to supply that client with identical settings each time it connects to the server. But DHCP can also be configured such that the server assigns addresses to each “interested” host “dynamically” from an address pool which is set up for that purpose. In the latter case, the DHCP server will try to assign the same address to the client each time it receives a request from it (and even over longer periods). This of course won’t work if there are more client hosts in the network than network addresses available.

With these possibilities, DHCP can make life easier for system administrators in two ways. Any changes (even larger ones) related to addresses and the network configuration in general can be implemented centrally by editing the server’s configuration file, which is much more convenient than reconfiguring lots of client machines. Also it’s much easier to integrate machines – and in particular new machines – into the network, as they can be given an IP address from the pool. The possibility of retrieving the appropriate network settings from a DHCP server can be especially useful in the case of laptops regularly used in different networks.

A DHCP server not only supplies the IP address and the netmask, but also the host name and the domain name, as well as the gateway and the name server addresses to be used by the client.

In addition to that, DHCP allows for a number of other parameters to be configured in a centralized way, such as a time server from which clients may poll the current time, or even a print server.

In the following section, we will give you an overview of DHCP, without describing the service in every detail. In particular, we want to show you, using the DHCP server dhcpd, how simple it can be, even in your own network, to carry out the entire setup by DHCP, from one central point.

B.2 DHCP Software Packages

SuSE Linux 7.2 comes with three packages related to DHCP, all included in series `n`.

The first of these is the DHCP server `dhcpd` distributed by the Internet Software Consortium, or ISC. This is the program which assigns and manages the corresponding information for the network. Normally with SuSE Linux there is only this one program available as far as the server is concerned, but you may choose from two different DHCP client programs. SuSE Linux includes both the dhcpclient – which is also from the ISC – and the "DHCP client daemon" which is provided in the package `dhcpcd`.

SuSE Linux 7.2 installs `dhcpcd` by default. The program is very easy to handle and is launched automatically on each system startup to look out for a DHCP server. It does not need a configuration file to do its job and should work out of the box in most standard setups.

If you administer a more complex network, you might need the ISC's dhclient which can be controlled via the configuration file `/etc/dhclient.conf`. No matter whether you want to include an additional domain in the search list, or even to emulate the behavior of a Microsoft DHCP client – if you are knowledgeable about networks you'll find that the dhclient gives you numerous possibilities to customize it to your needs, down to the last detail.

B.3 The DHCP Server dhcpd

The core of any DHCP system is the *dynamic host configuration protocol daemon*. What this server does is to "lease" addresses and to watch how they are used, in line with the settings as defined in the configuration file `/etc/dhcpd.conf`. By changing the parameters and values in this file, a system administrator can influence the program's behavior in numerous ways.

Let's have a look at a basic sample `/etc/dhcpd.conf` file:

This simple configuration file should be sufficient to get the DHCP server to assign IP addresses to the hosts of your network. One thing to remember, though, is to include a semicolon (;) at the end of each line. Without that character, you'll find that dhcpd won't even start!

As you might have noticed, the above sample file can be divided into three different sections.

In the first one, we've defined how many seconds an IP address is "leased" to a requesting host by default (`default-lease-time`) before it should apply for renewal. The section also includes a statement on the maximum period for which a machine may keep an IP address assigned by the DHCP server without applying for renewal (`max-lease-time`).

In the second part, some basic network parameters are defined on a global level:

- The line `option domain-name` defines the default domain of your network.

```
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;
 }
```

File contents B.3.1: The configuration file `/etc/dhcpd.conf`

- With the entry `option domain-name-servers`, you can specify up to three values for the DNS servers used to resolve IP addresses to host names (and vice versa). Ideally, you should have configured a name server on your machine or somewhere else in your network before setting up DHCP. This name server should also define a host name for each dynamic address, and vice versa. If you want to learn how to configure your own name server, please read chapter A on page 103.
- The line `option broadcast-address` defines the broadcast address to be used by the requesting host.
- With `option routers` you can tell the server where to send data packets which cannot be delivered to a host on the local network (according to the source and target host address and the subnet mask provided). In most cases, and especially in smaller networks, this router will be identical with the Internet gateway.
- With `option subnet-mask`, you specify the netmask assigned to clients.

Beneath these general settings, a network, including a subnet mask, is defined. To finish off, we specify the address range that the DHCP daemon should use to assign IP addresses to interested clients. In our example, clients may be given any address between `192.168.1.10` and `192.168.1.20`, as well as `192.168.1.100` and `192.168.1.200`.

If your server has more than one network card, you should edit the file `/etc/rc.config.d/dhcpd.rc.config` and specify the interfaces that dhcpd is intended to use, under `DHCPD_INTERFACE`.

After editing these few lines, you should be able to activate the DHCP daemon by issuing the command **`rcdhcpd start`**. The server is ready for use immediately after that. You could also do a basic check to see whether the configuration file is syntactically correct, by entering the command **`rcdhcpd syntax-check`**. If you encounter any unexpected problems with your configuration, i. e. the server aborts with an error or doesn't return “done” upon startup, you should be able to find out what has gone wrong by looking for information either in the main system log, `/var/log/messages`, or on console 10 (`Strg` + `Alt` + `F10`).

If you want dhcpd to be enabled automatically on system startup, you should edit the file `/etc/rc.config`, to set the variable `START_DHCPD` to `yes`. Naturally, you can also use the rc.config editor of YaST2 to do this.

Congratulations, you've just set up your own DHCP server!

B.4 Assigning Fixed IP Addresses to Hosts

Now that we're done setting up the server to assign dynamic addresses, it's time to have a closer look at *static* addresses and the way to configure them. As mentioned above, with DHCP it's also possible to assign a predefined, fixed address to one host each time the latter sends a request to the server.

As might be expected, addresses that have been assigned explicitly will always take priority over addresses from the pool of dynamic addresses. Furthermore, a static address will never expire in the way a dynamic address would (i. e. in case there aren't enough addresses available anymore so that the server needs to redistribute them among hosts).

To identify a host configured to get a *static* address, the DHCP daemon fetches the hardware address of that host. This is a numerical code consisting of six octet pairs, which is unique to each network device sold in the world, e. g. `00:00:45:12:EE:F4`.

If the appropriate lines, like the ones in B.4, are added to the configuration file B.3.1 on the preceding page, the DHCP daemon will assign the same set of data to the corresponding host under all circumstances.

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

File contents B.4.1: Entry added to the configuration file

The structure of this entry should be almost self-explanatory:

The first line sets the DNS name of the newly defined host (host *host name*), and the second one its MAC address. On any network-enabled Linux host, this address can be determined very easily with the command **`ifconfig`** (look for `HWaddr` in the output). Windows machines, too, have commands to elicit this kind information from the system: on Windows 95/98/ME, you'd enter **`winipcfg`** and on Windows NT/2000 **`ipconfig /all`**.

In the above example, a host with a network card having the MAC address *00:00:45:12:EE:F4* is automatically assigned the IP address `192.168.1.21` and the host name `earth`.

The type of hardware to be entered should be `ethernet` in nearly all cases, though `token-ring` (which is often found on IBM systems) is also supported.

B.5 The Finer Points

As we've said at the beginning of this chapter, these pages are only intended to provide a brief survey of what you can do with DHCP. If you are interested in further information, the page of the *Internet Software Consortium* on the subject (`http://www.isc.org/products/DHCP/`) will prove a good source to read about the details of DHCP, including about version 3 of the protocol which is currently in beta testing. Apart from that, you can always rely on the man pages for further help, expecially **`man dhcpd`**, **`man dhcpd.conf`**, **`man dhcpd.leases`** and **`man dhcp-options`**. Also, if you look around you'll be able to find one of the several books on the *Dynamic Host Configuration Protocol* that have been published over the years, and which take an in-depth look at the topic.

Incidentally, `dhcpd` can even supply requesting hosts with a file which contains a bootable operating system kernel, and which is defined in the configuration file by the parameter *filename*. This allows you to build client hosts which don't need a hard disk, i. e. they're enabled to load both their operating system and their network data over the network (*diskless clients*). Which could be an interesting option for both cost and security reasons. Now add the package alice to all this, and you'll be able to do some really amazing things. But that's another story…

C Proxy Server: Squid

The following chapter describes how caching web sites assisted by a proxy server works, and what the advantages are of using Squid.

The most popular proxy cache for Linux/UNIX platforms is Squid. We will discuss its configuration, the specifications required to get it running, how to configure the system to do transparent proxying, how to gather statistics about the cache's use with the help of programs like Calamaris and cachemgr and how to filter web contents with squidgrd.

C.1 What is a Proxy Cache?

Squid acts as a proxy cache. It behaves like an agent which receives requests from clients (in this case web browsers) and passes them to the specified server provider. When the requested objects arrive at the agent, it stores a copy in a disk cache.

Benefits arise when different clients request the same objects: these will be served directly from the disk cache, which is much faster than obtaining them from the Internet and, at the same time, saving overall bandwith from the system.

Tip

Squid covers a wide range of features, such as defining hierarchies of proxy servers to distribute the load, defining strict access control lists to all clients willing to access the proxy and, with the help of other applications, allowing or denying access to specific web pages. It can also obtain statistics about the most visited web sites, user usage of the Internet, and much more.

Squid is not a generic proxy. It normally communicates between HTTP connections. It also supports the protocols FTP, Gopher, SSL and WAIS, but it does not support other Internet protocols such as Real Audio, news or videoconferencing. Because Squid only supports the UDP protocol to provide communication between different caches, many other multimedia programs are not supported.

C.2 Some Facts About Cache Proxying

C.2.1 Squid and Security

It is also possible to use Squid together with a firewall, to secure internal networks from the outside, using a proxy cache. The firewall denies all external services except for Squid, forcing all World Wide Web connections to be established by the proxy.

If the firewall configuration includes a DMZ, set the proxy there. In this case, it is important that all computers in the DMZ send their log files to hosts inside the secured network.

One way to implement this feature is with the aid of a so-called "transparent" proxy. This is covered in Section C.6 on page 130.

C.2.2 Multiple Caches

"Multiple Caches" means configuring different caches so that objects can be exchanged between them, reducing the total system load as well as increasing the chances of finding an object already in the local network. It enables the configuration of cache hierarchies so that a cache is able to forward object requests to sibling caches or to a parent cache. It can get objects from another cache in the local network or directly from the source.

Choosing the appropiate topology for the cache hierarchy is very important, because we do not want to increase the overall traffic on the network. For example, in a very large network it is possible to configure a proxy server for every subnetwork and connect it to a parent proxy, connected in its turn to the proxy cache from the ISP.

All this communication is handled by ICP (Internet Cache Protocol) running on top of the UDP protocol. Data transfers between caches are handled using HTTP (Hyper Text Transmission Protocol), which is based on TCP, but for these kinds of connections it is preferable to use faster and simpler protocols capable of reacting to incoming requests within a maximum of one or two seconds.

In order to find the best server from which to get the desired objects, one cache sends an ICP request to all sibling proxies. These will answer the requests via ICP responses with a HIT code if the object was detected or a MISS if it was not. If multiple HIT responses were found, the proxy server will decide which server to from, download depending on factors such as which cache sent the fastest answer or which one is closer. If no satisfactory responses have been sent, the request will be sent to the parent cache.

Tip

To avoid duplication of objects in different caches in our network, other ICP protocols are used such as CARP (Cache Array Routing Protocol) or HTCP (Hyper-Text Cache Protocol). The more objects maintained in the network, the greater the possibility of finding the one we are looking for.

C.2.3 Caching Internet Objects

Not all objects available in our network are static. There are a lot of dynamically generated CGI pages, visitor counters, or encrypted SSL content documents. This is the reason such objects are not stored in the cache: every time you access one of these objects, it will already have changed again.

But the question remains as to how long all the other objects stored in the cache should stay there. To determine this, all objects in the cache are assigned three different states:

1. **FRESH:** When this object is requested, it is sent without comparing it to the the original object on the web to see if it has changed.
2. **NORMAL:** The original server is queried to see if the object has changed. If it changed, the cache copy is updated.
3. **STALE:** The object is no longer considered valid, and will be downloaded again from the server.

Web and proxy servers find out the status of an object by adding headers to these objects such as “Last modified” or “Expires” and the corresponding date. Other headers specifying that objects must not be cached are used as well.

Objects in the cache are normally replaced due to a lack of free hard disk space, using algorithms such as LRU (Last Recently Used), which serve to replace cache objects. The essential principle is to replace less requested objects.

C.3 System Requirements

The most important thing is to determine the maximum load that our system will have to bear. It is therefore important to pay more attention to the load peaks because these might be more than four times the day’s average. When in doubt, it would be better to overestimate the system’s requirements, because having Squid working close to the limit of its capabilities could lead to a severe loss in the quality of service.

In the following sections, several system factors will be presented in order of importance.

C.3.1 Hard Disk

Speed plays an important role in the caching process so should be of utmost concern. In hard disks, this parameter is described as “random-seek time” in milliseconds; as a rule of thumb: the lower this value, the better.

According to the Squid User’s Guide (`http://www.squid-cache.org`), for a system using only one disk, the formula for calculating the number of requests per second from the seek time of the disks is quite easy:

requests per second = 1000 / seek time

Squid enables multiple disks to be used simultaneously, increasing the number of requests per second. For instance, if you have three disks with the same seek time of 12 milliseconds, using the following formula will result in:

requests per second = 1000 / (seek time / number of disks) = = 1000 / (12/3) = 250 requests per second

In comparison to using IDE or SCSI disks, SCSI is preferable; newer IDE disks, however, have similar seek times to SCSI and, together with DMA- compatible IDE controllers, increase the speed of data transfer without considerably increasing the system load.

Size of the Disk Cache

In a small cache, the probability of a HIT (finding the requested object already located there) will be small because the cache is quickly filled up and, in this case, the less requested objects will be replaced by newer ones. On the other hand, if `1 GB` is available for the cache and the users only need `10 MB` a day to surf, it will take more than 100 days to fill the cache.

Probably the easiest way to determine the cache size needed is to consider the maximum transfer rate of our connection. With a `1 MB/s` connection, the maximum transfer rate will be `125 KB/s`. If all this traffic ends up in the cache, in one hour it will add up to `450 MB`, and, assuming that all this traffic is generated in only 8 working hours, it will reach `3.6 GB` in one day. Since the connection was not used up to its maximum capacity (otherwise we would have procured a faster one), we could assume that the total amount of data going through the cache is about `2 GB`. In the example, to keep all the browsed data of *one* day in the cache, we will require `2 GB` of disk space for Squid.

Summing up, Squid tends to read and write smaller blocks from or to the disk, so that how quickly it detects these objects on the disk is more important than having a fast disk.

C.3.2 RAM

The amount of memory required by Squid directly correlates to the amount of objects allocated in the cache. Squid also stores cache object references and frequently requested objects in memory to speed up retrieval of this data. The memory is one million times faster than a hard disk! (Compare the search time of a hard disk, about 10 milliseconds, with the 10 nanoseconds access time of the newer RAM memories)

Every object in RAM memory has a size of `72 bytes` (for "small" pointer architectures like **`Intel`**, **`Sparc`**, **`MIPS`**, etc. For Alpha it is `104 bytes`. If the average size of an object on the Internet is about `8 KB` and we have `1 GB` disk for the cache, we will be storing about 130,000 objects, resulting in close to `10 MB` RAM for meta-data alone.

Squid also stores other data in memory, such as a table with all used IP adresses, a fully qualified domain names cache, hot objects (the most requested), buffers, access control lists, etc.

It is very important to have more than enough memory for the Squid process, because if it has to be swapped, the system performance will be dramatically reduced. In order to assist us in cache memory management, we can use the tool cachemgr.cgi, as discussed later on in Section C.7.1 on page 133.

C.3.3 CPU

Squid is not a program that requires intensive CPU usage. The load of the processor is only increased while the contents of the cache are being loaded or checked. Using a multi-processor machine doesn't increase the performance of the system. To increase efficiency it is better to buy faster disks or add more memory.

Some examples of configured systems running Squid are available at `http://wwwcache.ja.net/servers/squids.html`.

C.4 Starting Squid

Squid is already preconfigured in SuSE Linux so that you can easily start it immediately after installation. A prerequisite for a smooth start is an already configured network, at least one name server and, of course, Internet access. Problems can arise if a dial-up connection is used with dynamic DNS configuration. In cases such as this, at least the name server should be clearly entered, since Squid will only start if it does not detect a DNS in the `/etc/resolv.conf`.

To start Squid, enter at the command line, as `'root'`:

```
rcsquid start
```

For the initial start-up, the directory structure will first have to be defined in `/var/squid/cache`. This is done automatically by the start script `/etc/init.d/squid` and can take a few seconds or even minutes. If `done` appears to the right in green, Squid has been successfully loaded. You can test Squid's functionality on the local system by entering `localhost` and `Port 3128` as proxy in the browser. In order to allow all users to access Squid and thus the Internet, you will only need to change the entry in the configuration file `/etc/squid.conf` from `http_access deny all` to `http_access allow all`. However, in doing so you should be aware that Squid is made completely accessible to anyone by this action. Therefore, you should, whatever the case, define ACL's which control access to the proxy. But more on this in the next chapter.

If you have made changes in the configuration file `/etc/squid.conf`, you will have to instruct Squid to load the changed file. You can do this by entering:

```
rcsquid reload
```

Or you can restart Squid:

```
rcsquid restart
```

Also, the following command is important:

```
rcsquid status
```

With this, you can determine whether the proxy is running and with

```
rcsquid stop
```

you can halt Squid. The latter can take a while since Squid waits up to half a minute (`shutdown_lifetime`) before dropping the connections to the clients and then it still as to write its data to the disk. If Squid is halted with **`kill`** or **`killall`**, this can lead to the destruction of the cache which will then have to be fully removed in order to be able to restart Squid.

If Squid dies after a short period of time, even though it has seemingly been started successfully, this could be the result of a faulty name server entry or a mssing `/etc/resolv.conf` file. The cause of the start failure would then be logged by Squid in the `/var/squid/logs/cache.log` file.

If Squid should be loaded automatically when the system boots, you will only need to reset the entry `START_SQUID=no` to `START_SQUID=yes` in the `/etc/rc.config` file.

An uninstall of Squid will neither remove the cache nor the log files. You would have to manually delete the `/var/squid` directory.

Local DNS Server

Setting up a local DNS server such as **`BIND-8`** or **`BIND-9`** makes absolute sense even if the server does not manage its own domain. It will then simply act as a “caching-only DNS” and will also be able to resolve DNS requests via the root name server without requiring any special configurations. If you enter this in the `/etc/resolv.conf` with the IP address `127.0.0.1` for **`localhost`**, Squid will detect a vaild name server when it starts up. Configuring a name server is, however, a chapter in itself and will therefore not be described here at length. It is sufficient, however, to install the package and to start BIND. The name server of the provider should be entered in the configuration file `/etc/named.conf` under **`forwarders`**, along with its IP address. If you have a firewall running, even if it is just the personal firewall, you will have to make sure that the DNS requests will be allowed through.

C.5 The Configuration File /etc/squid.conf

All HTTP proxy server settings are to be made in the `/etc/squid.conf` file. To be able to start Squid for the first time, no changes are necessary in this file, but external clients will initially be denied access. The proxy needs to be made available for the **`localhost`** and, usually, with `3128` as port. The options are extensive and ample documentation and examples are provided in the preinstalled `/etc/squid.conf` file. Nearly all entries begin with a # sign (the lines are commented out) and the relevant specifications are to be found at the end of the file. The values given almost always correlate with the default values, so that removing the comment signs without changing any of the parameters actually has little effect in most cases. It is better to leave the sample as it is and to reinsert the options along with the modified parameters in the line below. In this way, you can easily interpret the default values and the changes.

If you have updated an earlier Squid version, it is recommended that you edit the new `/etc/squid.conf` and only apply the changes you made in the previous file. If you try to implement the old `squid.conf` again, you are running a risk that the configuration will no longer function, since options are always being modified and new changes added.

General Configuration Options

http_port 3128 This is the port where Squid listens for client requests. The default port is `3128`, but `8080` is also common. You have the option here of specifying several port numbers, separated by blank spaces.

cache_peer <hostname> <type> <proxy-port> <icp-port> Here, you can enter a parent proxy as "parent", for example, if you want to, or use that of the provider. As `<hostname>`, the name and IP address of the proxy to be used are entered, and as `<type>`, `parent`. For `<proxy-port>`, the port number is to be entered which is also specified by the operator of the parent for use in the browser, usually `8080`. You can set the `<icp-port>` to `7` or `0` if the ICP port of the parent is not known and its use is irrelevant to the provider. In addition, `default` and `no-query` should be specified after the port numbers in order to strictly prohibit the use of the ICP protocol. Squid will then behave like a normal browser, as far as the provider's proxy is concerned.

cache_mem 8 MB This entry defines the maximim amount of disk space Squid can use for the caches. The default is `8 MB`.

cache_dir ufs /var/squid/cache 100 16 256 The entry `cache_dir` defines the directory where all the objects are to be stored on disk. The numbers at the end indicate the maximum disk space in `MB` to be used as well as the number of directories in the first and second level. The `ufs` parameter should be left alone. The default is `100 MB` occupied disk space in the `/var/squid/cache` directory, and to create 16 sub-directories inside it which each contain 256 more sub-directories. When specifying the disk space to be used, you should always leave sufficient reserve disk space. Values from a minimum of 50 to a maximum of 80 percent of the available disk space make the most sense here. The last two numbers for the directories should only be increased with caution, since too many directories can also lead to performance problems. If you have several disks which are to share the cache, you can enter several `cache_dir` lines.

cache_access_log /var/squid/logs/access.log path for log message

cache_log /var/squid/logs/cache.log path for log message

cache_store_log /var/squid/logs/store.log path for log message

These three entries specify the path where Squid will log all of its actions. Normally, nothing is changed here. If Squid is experiencing a heavy usage burden, it might make sense to distribute the cache and the log files over several disks.

emulate_httpd_log off If the entry is set to `on`, you will obtain readable log files. Some evaluation programs cannot interpret this, however.

client_netmask 255.255.255.255 With this entry you can mask the logged IP addresses in the log files to hide the clients' identity. The last digit of the IP address will be set to zero if you enter `255.255.255.0` here.

ftp_user Squid@ With this, you can set the password which Squid should use for the anonymous FTP login. The login `'anonymous'` and your e-mail address as password are generally used to access public FTP servers, which saves you the trouble of entering your username and password each time you download FTP. `Squid@` without the domain is the default, since the clients can originate from any domain. It can still make sense, however, to specify a valid e-mail address here, since some FTP servers can check these for validity.

cache_mgr webmaster An e-mail address to which Squid sends a message if it unexpectedly crashes. The default is `webmaster`.

logfile_rotate 0 If you call up **`squid -k rotate`**, Squid can rotate secured log files. The files are numbered, depending on the number given, after reaching the specified value, the oldest file at that point will be overwritten. This value here normally stands for `0` because archiving and deleting log files in SuSE Linux is carried out by a cronjob which can be found in the configuration file `/etc/logfiles`. The period of time after which the files are deleted is defined in the `/etc/rc.config` file via the `MAX_DAYS_FOR_LOG_FILES` entry.

append_domain <domain> With `append_domain`, you can specify which domain will automatically be appended when none is given. Usually your own domain is entered here, so entering `www` in the browser suffices to guarantee access to your own web server.

forwarded_for on If you set the entry to `off`, Squid will remove the IP address and the system name of the client from the HTTP requests.

negative_ttl 5 minutes; negative_dns_ttl 5 minutes Normally you don't need to change these values. If you have a dial-up connection, however, the Internet may, at times, not be accessible. Squid will make a note of the failed requests and will then refuse to issue new ones, even though the Internet connection has been re-established. In a case such as this, you can change the `minutes` to `seconds` and then, after clicking on `Reload` in the browser, the dial-up process should be re-engaged after a few seconds.

never_direct allow <acl_name> If you want to prevent Squid from taking requests directly from the Internet, you can use the above command to force connection to another proxy. You need to have previously entered this in `cache_peer`. If `all` is specified as the `<acl_name>`, you will force all requests to be forwarded directly to the `parent`. This might be necessary, for example, if you are using a provider which strictly stipulates the use of its proxies or denies its firewall direct Internet access.

Options for Access Controls

Squid provides an intelligent system which controls access to the proxy. By implementing so-called “ACL’s”, it can be configured easily and comprehensively. This involves lists with rules which are processed sequentially. ACL’s must be defined first, before they can be used. Some default ACL’s such as `all` and `localhost` already exist. Defining an ACL on its own will not yet yield any results until it is put to use, e. g. in conjunction with `http_access`, when the defined rules are implemented:

acl <acl_name> <type> <data> An ACL requires at least three specifications to define it. The name `<acl_name>` can be arbitrarily chosen. For `<type>`, you can select from a variety of different options which can be found in the `ACCESS CONTROLS` section in the `/etc/squid.conf` file. The specification for `<data>` depends on the individual ACL type and can also be read from a file, e. g. via host names, IP addresses or URLs. Here are some simple examples:

```
acl mysurfers srcdomain .my-domain.com
acl teachers src 192.168.1.0/255.255.255.0
acl students src 192.168.7.0-192.168.9.0/255.255.255.0
acl lunch time MTWHF 12:00-15:00
```

http_access allow <acl_name> `http_access` defines who is allowed to use the proxy and also who can access what on the Internet. For this, ACL’s are specified. `localhost` and `all` have already been defined above, which can deny or allow access via `deny` or `allow`. A list containing any number of `http_access` entries can be created, processed from top to bottom and, depending on which occurs first, access will be allowed or denied to the respective URL. The last entry should always be `http_access deny all`. In the following example, the `localhost` has free access to everything while all other hosts are denied access completely.

```
http_access allow localhost
http_access deny all
```

Another example, where the previously defined ACL’s are used: The group `'teachers'` always has access to the internet, while the group `'students'` only gets access Monday to Friday during lunch time.

```
http_access deny localhost
http_access allow teachers
http_access allow students lunch time
http_access deny all
```

The list with the `http_access` entries should only be entered, for the sake of readability, at the designated position in the `/etc/squid.conf` file. That is, between the text

```
\# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
```

and the closing

```
http_access deny all
```

redirect_program /usr/bin/squidGuard With this option, a redirector such as SquidGuard, which is able to block unwanted URLs, can be specified. Internet access can be individually controlled for various user groups with the help of proxy authentication and the appropriate ACL's. SquidGuard is a package of its own, which can be separately installed and configured.

authenticate_program /usr/sbin/pam_auth If users need to be authenticated on the proxy, a corresponding program such as pam_auth can be specified here. When accessing pam_auth for the first time, the user will see a login window where the username and password must be entered. In addition, an ACL is still required so that only clients with a valid login can surf the Internet:

```
acl password proxy_auth REQUIRED

http_access allow password
http_access deny all
```

The `REQUIRED` after `proxy_auth` can be substituted by a list of permitted usernames.

ident_lookup_access allow <acl_name> With this, you will manage to have an ident request run through for all ACL-defined clients in order to find out each user's identity. If you apply `all` to the `<acl_name>`, this will be valid for all clients. Also, an ident daemon must be running on all clients, for Linux, you can install the pidentd package for this purpose, for **`Windows`**, there is free software available to download from the Internet. To ensure that only clients with a successful ident lookup are permitted, a corresponding ACL will also have to be defined here:

```
acl identhosts ident REQUIRED

http_access allow identhosts
http_access deny all
```

Here, too, you can replace the `REQUIRED` with a list of permitted usernames. Using `ident` can slow down the access time quite a bit, since ident lookups will have to be repeated for each request.

C.6 Transparent Proxy Configuration

The usual way of working with proxy servers as follows: the web browser sends requests to a certain port in the proxy server and the proxy provides these required objects, whether they are in its cache or not. When working in a real network, several situations may arise:

- For security reasons, it is recommended that all clients use a proxy to surf the Internet.
- All clients must use a proxy whether they are aware of it or not.
- In larger networks already using a proxy, it is possible to spare yourself the trouble of reconfiguring each machine whenever changes are made in the system.

In all these cases a transparent proxy may be used. The principle is very simple: the proxy intercepts and answers the requests of the web browser, so that the web browser receives the requested pages without knowing where they are coming from. This entire process is done transparently, hence the name.

C.6.1 Kernel Configuration

First of all we have to make sure that the proxy server's kernel has support for transparent proxy. Otherwise we will have to add this option to the kernel, and compile it again. More on this topic is available in the *SuSE Linux Reference Guide*.

In the entry corresponding to Networking Options, select '`Network Firewalls`', and then the options '`IP: firewalling`' and '`IP: Transparent proxying`'. Now we just have to save the new configuration, compile the new kernel, install it, reconfigure LILO if necessary, and restart the system.

C.6.2 Configuration Options in /etc/squid.conf

Let us take a look at the options that we need to activate in the `/etc/squid.conf` file to get the transparent proxy up and running.

The options are:

- **`httpd_accel_host`** virtual
- **`httpd_accel_port`** 80 # the port number where the actual HTTP server is located
- **`httpd_accel_with_proxy`** on
- **`httpd_accel_uses_host_header`** on

C.6.3 Firewall Configuration with SuSEfirewall

Now we have to redirect all incoming requests via the firewall with help of a port-forwarding rule to the Squid port.

To do this, we will use a tool provided by SuSE: SuSEfirewall. Its configuration file can be found in `/etc/rc.config.d/firewall.rc.config`. The configuration file, in turn, consists of well documented entries. Even if we only want to set a transparent proxy, we will have to configure a couple of firewall options. In our example:

- Device pointing to the Internet: **`FW_DEV_WORLD`**="eth1"
- Device pointing to the network: **`FW_DEV_INT`**="eth0"

Ports and services (see /etc/exports) on the firewall being accessed from untrusted networks as Internet. This example only provides web services to the outside:

FW_SERVICES_EXTERNAL_TCP="www"

Ports/services (see /etc/exports) on the firewall being accessed from the secure network, both TCP and UDP services.

FW_SERVICES_INTERNAL_TCP="domain www 3128"

FW_SERVICES_INTERNAL_UDP="domain"

We are accessing web services and Squid (whose default port is 3128).

The service "domain" specified before stands for DNS or Domain Name Server. It is normal to use this service, otherwise we can simply take it out of the above entries and set the following option to **no**:

FW_SERVICE_DNS="yes"

The most important option is number 15:

```
#
# 15.)
# Which accesses to services should be redirected to a localport
# on the firewall machine?
#
# This can be used to force all internal users to surf via your
# squid proxy, or transparently redirect incoming webtraffic to
# a secure webserver.
#
# Choice: leave empty or use the following explained syntax of
# redirecting rules, separated by a space.
# A redirecting rule consists of 1) source IP/net,
# 2) destination IP/net, 3) original destination port and
# 4) local port to redirect the traffic to, separated by a colon,
# e.g. "10.0.0.0/8,0/0,80,3128 0/0,172.20.1.1,80,8080"
#
```

The comments above show the syntax to be followed. First of all, the IP address and the netmask of the "internal networks" are accessing the proxy firewall. Secondly, the IP address and the netmask to which these clients are "sending" their requests. In the case of web browsers, we will specify the networks **0/0**, a wildcard that means "to everywhere". After that, the "original" port to which these requests are sent and, finally, the port to which all these requests are "redirected".

As Squid supports more protocols than HTTP, we can also redirect requests from other ports to our proxy. For example, we can also redirect services for FTP (port 21), HTTPS or SSL (port 443), and HTTP or web (port 80) to our Squid port. In the example we use the default port 3128.

In case there are more networks or services to add, they only need to be separated by a single blank character in the corresponding entry.

Following our example, we will give proxy access in our network only to web and FTP protocols.

FW_REDIRECT_TCP="192.168.0.0/16,0/0,80,3128 192.168.0.0/16,0/0,21,3128"

FW_REDIRECT_UDP="192.168.0.0/16,0/0,80,3128 192.168.0.0/16,0/0,21,3128"

In order to start the firewall and the new configuration with it, we have to change an entry in the `/etc/rc.config` file. The entry **`START_FW`** must be set to "yes":

`START_FW`="yes"

Note

After the configuration file `/etc/rc.config` has been edited, you will have to execute the SuSEconfig script, all by hand, in order to make the changes effective in your system. The safest way of doing this is with help of YaST '`System administration`' -> '`Change configuration file`' which will run SuSEconfig automatically.

Start Squid as shown in the Section C.4 on page 125. To check if everything is working properly, take a look at the Squid logs in `/var/squid/logs/access.log`

In order to verify that all ports are correctly configured, we can perform a port scan on the machine from any computer outside our networks. Only the web services port (80) should be opened. The way to do the portscan is with **`nmap`**:

`nmap -O IP_address`

C.7 Squid and Other Programs

In the following section, we will see how other applications interact with Squid. `cachemgr.cgi` enables the system administrator to check the amount of memory needed for caching objects, squidgrd filters web pages, and Calamaris is a report generator for Squid.

C.7.1 cachemgr.cgi

The cache manager (cachemgr.cgi) is a CGI utility for displaying statistics about the memory usage of a running Squid process. It is also a more convenient way to manage the cache and view statistics without logging the server.

Setup

First of all, we need a running web server on our system. To check if Apache is already running, type as `'root'`: **`rcapache status`**.

If a message like this appears:

```
Checking for service httpd: OK
Server uptime: 1 day 18 hours 29 minutes 39 seconds
```

Apache is running on our machine. Otherwise type: **`rcapache start`** to start Apache with the SuSE Linux default settings.

The last step to set it up is to copy the file `cachemgr.cgi` to the Apache directory `cgi-bin`:

```
cp /usr/share/doc/packages/squid/scripts/cachemgr.cgi /var/www/cgi-bin
```

Cache Manager ACL's in /etc/squid.conf

There are some default settings in the original file required for the cache manager:

```
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
```

With the following rules:

```
http_access allow manager localhost
http_access deny manager
```

The first ACL is the most important, as the cache manager tries to communicate with Squid over the cache_object protocol.

The following rules assume that the web server and Squid are running on the same machine. If the communication between the cache manager and Squid originates at the web server on another computer, we will have to include an extra ACL as in figure C.7.1.

```
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 of webserver
```

File contents C.7.1: Adding extra ACL

Then add the rules as in figure C.7.2 on the next page.

We can also configure a password for the manager if we want to have access to more options such as closing the cache remotely or viewing more information about the cache. We then have to configure the entry **`cachemgr_passwd`** with a password for the manager and the list of options that we want to be able to view. This list appears as a part of the entry comments in `/etc/squid.conf`.

Remember to restart Squid with the **`-k reconfigure`** option every time the configuration file is changed.

Viewing the Statistics

Go to the corresponding web site:
`http://webserver.example.org/cgi-bin/cachemgr.cgi`

Press '`continue`' and browse through the different statistics. More details on each entry shown by the cache manager is in the Squid FAQ `http://www.squid-cache.org/Doc/FAQ/FAQ-9.html`

```
http_access allow manager localhost
http_access allow manager webserver
http_access deny manager
```

File contents C.7.2: Access Rules

C.7.2 SquidGuard

This chapter is not intended to go through an extensive configuration of SquidGuard, only to introduce it and give some advice on using it. For more in-depth configuration issues, please refer to the SquidGuard web site at: `http://www.squidguard.org`

SquidGuard is a free (GPL), flexible and ultra fast filter, redirector and “access controller plugin” for Squid. It lets you define multiple access rules with different restrictions for different user groups on a Squid cache. SquidGuard uses Squid’s standard redirector interface.

SquidGuard can be used for the following:

- limit the web access for some users to a list of accepted or well-known web servers or URLs.
- block access to some listed or blacklisted web servers or URLs for some users.
- block access to URLs matching a list of regular expressions or words for some users.
- redirect blocked URLs to an “intelligent” CGI-based info page.
- redirect unregistered users to a registration form.
- redirect banners to an empty GIF.
- have different access rules based on time of day, day of the week, date, etc.
- have different rules for different user groups.
- and much more..

Neither SquidGuard or Squid can be used to:

- Edit, filter or censor text inside documents
- Edit, filter or censor HTML-embedded script languages such as JavaScript or VBscript

Using SquidGuard

Install the squidgrd from the series `n`. Edit a minimal configuration file `/etc/squidguard.conf`. There are plenty of configuration examples in `http://www.squidguard.org/config/`. You can experiment later with more complicated configuration settings.

The following step is to create a dummy “access denied” page, or a more or less intelligent CGI page to redirect Squid in case the client requests a blacklisted web site. Again, using Apache is strongly recommended.

Now we have to tell Squid to use SquidGuard. We will use the following entries in the `/etc/squid.conf` file:

```
redirect_program /usr/bin/squidGuard
```

There is another option called **`redirect_children`** configuring how many different “redirect” (in this case SquidGuard) processes are running on the machine. SquidGuard is fast enough to cope with lots of requests (SquidGuard is quite fast: 100,000 requests within 10 seconds on a 500MHz Pentium with 5900 domains, 7880 URLs, 13780 in sum). Therefore it is not recommended to set more than 4 processes because this may lead to an unnecesary increase of memory for the allocation of these processes.

```
redirect_children 4
```

And last of all, send a HUP signal to Squid to have it read the new configuration:

```
squid -k reconfigure
```

Now you can test your settings with a browser.

C.7.3 Cache Report Generation with Calamaris

Calamaris is a Perl script used to generate reports of cache activity in ASCII or HTML format. It works with native Squid access log files. The Calamaris Home page is located at `http://Calamaris.Cord.de/`

The use of the program is quite easy. Log in as `‘root’`, and then:

```
cat access.log.files | calamaris [options] > reportfile
```

It is important when piping more than one log file that the log files are chronologically ordered, that is, older files first.

The various options:

-a normally used for the output of available reports

-w an HTML report

-l a message or logo in the header of the report

Further information on the various options can be found in the manual page: **`man calamaris`**

A typical example:

```
cat access.log.2 access.log.1 access.log | calamaris -a -w >
/usr/local/httpd/htdocs/Squid/squidreport.html
```

The report is stored in the directory of the web server. Again, Apache is required to view the reports.

Another powerful cache report generator tool is SARG (Squid Analysis Report Generator).

Further information on this can be found in the relevant Internet pages at: `http://web.onda.com.br/orso/`

C.8 More Information on Squid

Visit the homepage of Squid: `http://www.squid-cache.org/`. Here, you will find the Squid User Guide and a very extensive collection of FAQs on Squid.

The Mini-Howto regarding transparent Proxies in the package `howtoen`, under `/usr/share/doc/howto/en/mini/TransparentProxy.gz`

In addition, mailing lists are available for Squid at:
`squid-users@squid-cache.org`.

The archive for this is located at:
`http://www.squid-cache.org/mail-archive/squid-users/`

D Network Security

D.1 Masquerading and Firewalls

Owing to its outstanding network capabilities, Linux is becoming ever more widespread as a router operating system for dialup or dedicated lines. For the purposes of this chapter, by "router" we refer to a host which has more than one network interface and transmits any packets not destined for one of its own network interfaces to another host communicating with it (often called gateway). The packet filtering mechanism provided by the Linux kernel allows for a precise control over which packets of the overall traffic are allowed through and which packets are stopped.

In general, defining the exact rules for such a packet filter requires at least some experience on the part of the administrator. For the less experienced user, SuSE Linux includes two separate packages which are intended to make it easier to set up these rules: the longer-established package SuSEfirewall and the the newer package Personal-firewall. The main difference between them lies in the degree to which they are configurable and, as a result of this, there are also differences with regard to flexibility and intended purpose.

SuSEfirewall is highly configurable, making it a good choice for a more complex packet filtering setup. In contrast, Personal-firewall can be configured by setting just one variable, and is aimed at blocking any attempt to make an inbound connection with the corresponding host. Both packet filter solutions allow you to add filtering rules to enable masquerading (also called NAT = Network Address Translation), i. e. using a Linux machine as a router to link a local network through a dialup or dedicated connection, where only one IP address is visible to the outside world. In other words, masquerading is achieved by implementing rules for packet filtering.

Caution

This chapter only describes standard procedures which should work well in most situations. However, there's no guarantee that this book or other materials provided by us are free of mistakes which might have escaped our attention. Please don't make the authors of this book responsible if some evil cracker should gain access to your system, despite all your well-crafted security measures. On the other hand, we do appreciate your criticism and comments. Even though you might not receive a direct answer from us, rest assured that suggestions for improvement will be taken seriously.

D.1.1 Masquerading Basics

Masquerading is a special Linux case of NAT, or Network Address Translation. There's nothing terribly complicated about the underlying principle of this: Your router has more than one network interface, typically a network card and a modem (or an ISDN interface). While one of these interfaces will link you to the outside world, the remaining ones (or the only remaining one, for that matter) are used to connect this router with the other hosts in your network. Now let's have a look at an example where dialout connections are handled via ISDN, i. e. the network interface is `ippp0`. Several hosts of your local network are connected to the network card of your Linux router, which in this case is assumed to be `eth0`. The network address of your internal network is, say, `192.168.0.0`, while the router's address is `192.168.0.1`, and the hosts to be linked up have addresses like `192.168.0.2`, `192.168.0.3`, etc. These hosts will send any packets not destined for the local network to the address `192.168.0.1` which is the network interface of your router and thus functions as the network's default router or default gateway.

Note

Please make sure that both the broadcast addresses and the network masks are the same for all hosts when configuring your network!

With this setup, as soon as one of the hosts sends a packet destined for an Internet address, this packet ends up at the router machine because it's been set up as the network's default router. However, before anything can happen beyond that, the router needs to be configured to actually forward such packets – which SuSE Linux won't do by default for security reasons! Therefore, you need to set the variable `IP_FORWARD`, which is defined in the file `/etc/rc.config`, to `IP_FORWARD=yes`. The forwarding mechanism is enabled after rebooting or issuing this command:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

This is where the masquerading begins. Given that the router, as seen from the outside, has only one IP address (to stick with our example, after dialing out this address will belong to the ISDN interface), the source address of the packet must be replaced with the router's own address before sending it out over what is called the external network interface. If the router didn't replace the source address, the receiving end would have no means to reply with the packets that your host awaits. This is especially true if you are using the `192.168.x.x` address range. Though the latter represents a valid set of IP addresses, they're not forwarded at all by any of the Internet's routers.

With the exchange of source addresses, the target host at the other end of the link is told that it is supposed to talk to your router, but it won't see the host in your internal network. Your internal host is hidden behind the router, which is why the technique is called "masquerading".

In most cases sending some packets implies we expect to get something back in reply. Again because of the address translation, the target address for these reply packets will be our router. Now the router's task is to recognize the packets on their way back, translating the target address such that the host in the internal or local network is made to "believe" that the packets have been directly sent to it. The task of recognizing the packets belonging to a connection handled by a masquerading router is managed with the help of a table which is located right in the kernel of your router as long as this connection exists. By using the ipchains and the iptables command, the superuser (`'root'`) can even view these tables. Please read the documentation of these commands for detailed instructions on how to use them. In any case, it's worth mentioning that individual masqueraded connections are not only identified by their source and target addresses, but also by the port numbers and the protocols involved. Theoretically at least, this would enable your router to simultaneously "hide" many thousand connections per internal host.

With the routing of inbound traffic depending on the masquerading table there's no way to open a connection to some internal host from the outside. For such a connection, there would be no entry in the table because the entry itself is only created if an internal host opens a connection with the outside. In addition, any established connection gets assigned a status entry in the table, and this entry cannot be used by another connection, rather this second connection would have to rely on another status record.

As a consequence of all this, you might experience some problems with a number of applications: Programs use protocols to talk to each other and some of these try to open additional connections or to send packets from the server to their client which cannot be recognized as being valid by a simple packet filter. Examples of such protocols are FTP (only if in PORT mode; Netscape as well as the standard ftp program and many others use the PASV mode, and this passive mode is much less problematic as far as packet filtering and masquerading is concerned), ICQ, cucme, IRC (DCC, CTCP), Quake and others. The FTP protocol, for instance, opens a controlling connection and in addition another one for each file to be transferred (called the data connection). In PORT mode, the server opens a connection with the client but, in PASV (passive) mode, it's the client which establishes a connection. As we've said before, our setup allows for connections to be opened exclusively from the internal side, which explains the trouble that FTP will cause if used in PORT mode.

The Linux kernel handles masquerading or NAT with the help of packet filtering rules, so the next section is about firewalls.

D.1.2 Firewalling Basics

"Firewall" is probably the most widely used term to describe a mechanism to link two networks (however small they might be) while at the same time imposing a rather stringent policy on the data traffic between them. There are various types of firewalls which mostly differ with regard to the logical, abstract level on which traffic is analysed and controlled. Strictly speaking, the mechanism that we will describe in this section is called a "packet filter". A packet filter is just one way to set up a firewall in the broader sense. Like any type of firewall, a packet filter alone does not guarantee full protection from all security risks. What a packet filter does is implement a set of rules related to protocols, ports and IP addresses, in order to decide whether data may pass through or not. On the basis of this, one can block any packets which, according to their addresses were not supposed to be traveling on the network in the first place. For instance, you might be interested in blocking any packets related to the telnet service of your hosts on port 23. On the other hand, if you want people to have access to your web server, you'll need to enable the corresponding port. Note that a packet filter won't scan the contents of any packets as long as they have legitimate addresses (e. g. they're directed to your web server). Thus, packets could be sent which contain data to attack your CGI server, but the packet filter would let them through.

A more effective (but more complex) mechanism could be set up by combining several types of systems, e. g. a packet filter interacting with an application gateway/proxy. In this case, the packet filter rejects any packets which are not destined for our declared target, i. e. only packets directed to the application gateway are allowed through. This gateway or proxy now pretends to be the actual client of the server that we're contacting. In a sense, such a proxy could be considered as a masquerading machine for the protocol used by our application. One example for such a proxy is Squid, an HTTP proxy server. If you were to use Squid, you'd have to tell your browser to communicate via the proxy, such that any HTTP pages requested would be served from the proxy cache rather than directly from the Internet. As another example, the SuSE proxy suite (the package proxy-suite in series sec) includes a proxy for the FTP protocol.

In the following section, we're going to focus on the two packet filters that come with SuSE Linux. For more information and links on the topic, read the Firewall-HOWTO document included in package `howtoen`, series `doc`. If you have this package installed, you can read the HOWTO with the command **`less /usr/share/doc/howto/en/Firewall-HOWTO.gz`**, provided that you have installed the package howtoen.

D.1.3 Personal-firewall

As mentioned above, SuSE Linux includes two different packages to set up filtering rules: Personal-firewall and SuSEfirewall. The main difference between the two is their configurability. Personal-firewall should require no configuration and little maintenance; it is intended to give you the possibility of connecting to the Internet while not allowing any incoming connections, meaning that the host (or the network interface) protected by it will not offer any services to other hosts out there. Accordingly, the package personal-firewall should be an adequate solution for most home-based workstations.

Note

As we've already mentioned in the section about masquerading, the fact that incoming connection requests are rejected may lead to a number of problems with protocols trying to open a second connection to the client. In that respect, the effect that Personal-firewall can have on your applications is similar to the masquerading function.

Enabling personal-firewall is as easy as setting this one variable in the file `/etc/rc.config.d/security.rc.config`:

```
REJECT_ALL_INCOMING_CONNECTIONS
```

Please read the comments in this file to insert the network interface or interfaces (separated by spaces) which you want to be protected from incoming connections. Apart from the names of these interfaces themselves, such as `eth0`, `eth1`, or `ippp0`, you may also insert the following keywords:

`no`	Use this to disable personal-firewall. You can achieve the same by leaving the value for the variable REJECT_ALL_INCOMING_CONNECTIONS blank.
`yes`	This makes personal-firewall act on all interfaces except `lo`. `lo` is the loopback interface, localhost, i. e. connections to localhost are allowed.
`modem`	This refers to all the modems that are connected, as a short form for interface names beginning with `ppp`, i. e. `ppp0`, `ppp1`, ...
`masq`	Packets received by the host which are not destined for one of its own interfaces are to be masqueraded accordingly when forwarding them.

Table D.1: Possible keywords

Caution

Enabling the masquerading function makes sense only if IP forwarding (commonly also called routing) has been enabled, too. To achieve this, set the variable **`IP_FORWARD`** in `/etc/rc.config` to **`IP_FORWARD=yes`**. IP forwarding will be enabled after the next reboot, or after entering the command: **`echo 1 > /proc/sys/net/ipv4/ip_forward`**

So if you have entered:

```
REJECT_ALL_INCOMING_CONNECTIONS="ippp0 modem"
```

all connection requests arriving at your ISDN interface, or via your modem, are dropped. Assuming that you have configured a network card which is known to your system as `eth0`, that interface will not be affected by the firewall at all. Also, the masquerading function is disabled.

Note

ADSL and other variants of DSL are considered modems here because they get assigned an interface name like `ppp0` or `ppp1`.

Here's another example:

```
REJECT_ALL_INCOMING_CONNECTIONS="modem masq"
```

This would have the following effects: Interfaces beginning with `ppp` cannot accept connections anymore. Provided that IP forwarding is enabled (see the note above!), packets going through network interfaces that don't begin with `ppp` are "hidden" behind the IP address of your external interface. However, note that this does not necessarily determine the actual address used for masquerading! Theoretically, you could be online via both an ISDN card and a modem, and while incoming connections directed to the modem are rejected, those directed to the ISDN interface would be allowed through. With masquerading, the source address is always the address of the interface through which the packet will leave your router.

And yet another example: You have configured a network card to have the interface name `eth0`; in addition you have installed on your system an ISDN card which corresponds to `ippp0`. Now you enter these keywords:

```
REJECT_ALL_INCOMING_CONNECTIONS="ippp0 masq"
```

Again we assume that you've already enabled IP forwarding. Now if you have configured the hosts of your local network such that they use the IP address of the network card as their default router, any outgoing traffic from your local network will be masqueraded by the router. Inbound connection requests arriving at your ISDN interface are ignored.

Our last example:

```
REJECT_ALL_INCOMING_CONNECTIONS="masq"
```

Here, none of the connection requests arriving at the installed network interfaces are discarded, but connections forwarded from other hosts through your router are masqueraded. This is on the condition that the other hosts are configured to use an available network interface of your router as their default gateway. But be warned that any host that is actually able to send packets to you could hide behind your router, i. e. if its packets are not destined for one of your router's IP addresses.

In the following section we will describe the configuration of SuSEfirewall, which is a rather more challenging task and requires a certain degree of experience and understanding. Please note that neither the configuration of firewall packages nor the setup of masquerading are covered by the free installation support that SuSE provides.

D.1.4 SuSEfirewall

You can find documentation on SuSEfirewall in `/usr/share/doc/packages/SuSEfirewall`. Also, the theoretical background is also covered in this manual.

The configuration of SuSEfirewall is stored in `/etc/rc.config.d/firewall.rc.config` and is commented in English. The following paragraphs are a step-by-step description to successfully complete the configuration. For each configuration item, you'll find a note whether it is relevant for firewalling or masquerading. If you stumble across any comments in the configuration file that are related to what is called DMZ (or "demilitarized zone"), please note that this is not covered here.

If your requirements are strictly limited to masquerading, fill out the items marked with *masquerading* only.

- **`START_FW`** (firewall, masquerading): Set this variable to `yes` in `/etc/rc.config`, in order to ensure the script is started. This enables the firewall and/or masquerading.
- **`FW_DEV_WORLD`** (firewall, masquerading): For example `eth0`. This is the device to link you up with the Internet. In the case of ISDN, for instance, you'd have to insert `ippp0` here.
- **`FW_DEV_INT`** (firewall, masquerading): The device linking you up with the internal, "private" network. Leave this blank if there's no internal network, e. g. if the firewall is supposed to protect only this host.
- **`FW_ROUTE`** (firewall, masquerading): If you need the masquerading function, you have to insert `yes` here. For a firewall without masquerading, this should be done only if access to the internal network is required – which in turn would only work if your internal hosts use officially registered IP's. Normally, though, you should *not* allow access to your internal network from the outside. On the other hand, if you insert `yes` because of masquerading, your internal hosts are still invisible to the outside due to their private network addresses (e. g. `192.168.x.x`) which are ignored by Internet routers.
- **`FW_MASQUERADE`** (masquerading): Set this to `yes` if you need the masquerading function. Note that it's safer to have a proxy server between the hosts of the internal network and the Internet.
- **`FW_MASQ_NETS`** (masquerading): Insert the hosts and/or networks to be masqueraded, leaving a space between the individual entries. For example:

  ```
  FW_MASQ_NETS="192.168.0.0/24 192.168.10.1"
  ```

- **`FW_PROTECT_FROM_INTERNAL`** (firewall): Set this to `yes` if you want to protect your firewall host from attacks originating in your internal network as well. If you do so, services are only available to the internal network if explicitly enabled. Also see **`FW_SERVICES_INTERNAL_TCP`** and **`FW_SERVICES_INTERNAL_UDP`**.
- **`FW_AUTOPROTECT_GLOBAL_SERVICES`** (firewall): This should normally be left as it is, i. e. set to `yes`.
- **`FW_SERVICES_EXTERNAL_TCP`** (firewall): Enter the services to be made available, e. g. `"www smtp ftp domain 443"`. You'd normally leave this blank for a workstation at home that is not supposed to offer any services.
- **`FW_SERVICES_EXTERNAL_UDP`** (firewall): Leave this blank if you don't happen to run a name service which you want to make available to the outside. Otherwise, insert the ports to be used.

- **FW_SERVICES_INTERNAL_TCP** (firewall): This defines the services made available to the internal network. The notation is the same as for **FW_SERVICES_EXTERNAL_TCP**, but refers to the *internal* network in this case.

- **FW_SERVICES_INTERNAL_UDP** (firewall): See above.

- **FW_TRUSTED_NETS** (firewall): Here we specify the hosts that we can *really* trust ("trusted hosts"). Note, however, that these need to be secured by the firewall just like any others.

  Here's an example: `"172.20.0.0/16 172.30.4.2"` means that all hosts which have an IP address beginning with `172.20.x.x`, plus the host with the IP address `172.30.4.2`, are allowed to pass through the firewall.

- **FW_SERVICES_TRUSTED_TCP** (firewall): Here you can specify the port addresses which may be used by "trusted hosts". For instance, if you were to grant them access to all services, you'd enter `1:65535`. Usually, though, it will be sufficient to enter `ssh` as the only service.

- **FW_SERVICES_TRUSTED_UDP** (firewall): Just like above, but for UDP ports.

- **FW_ALLOW_INCOMING_HIGHPORTS_TCP** (firewall): Set this to `ftp-data` if you intend to use normal (active) FTP services.

- **FW_ALLOW_INCOMING_HIGHPORTS_UDP** (firewall): Set this to `dns` to be able to use the name servers registered in `/etc/resolv.conf`. If you enter `yes` here, all high ports will be enabled.

- **FW_SERVICE_DNS** (firewall): Enter `yes` here if you're running a name server that is supposed to be available to external hosts. At the same time, you'll need to enable port 53 under **FW_TCP_SERVICES_***.

- **FW_SERVICE_DHCLIENT** (firewall): Enter `yes` here if you're using dhclient to get your IP address.

- **FW_LOG_***: Here you can specify the firewall's logging activity. For normal operation, it'll be sufficient to set **FW_LOG_DENY_CRIT** to `yes`.

- **FW_STOP_KEEP_ROUTING_STATE** (firewall): Insert `yes` here if you have configured your dialout procedure to work automatically via diald or ISDN (dial on demand).

Now that you're done configuring SuSEfirewall, please don't forget to test your setup (e. g. with **telnet** from an external host). Have a look at `/var/log/messages`, where you should see something like this:

```
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)
```

D.2 SSH – Secure Shell, the Safe Alternative

In this age of the omnipresent network, accessing a remote system is a matter of fact. If we are using it to receive mail, mantain a server or to append an article to a web site of an editorial system – an authentication of the users carrying out such actions always has to take place.

These days, most users are well aware that their username and password is only intendend for their own use. Obligation to personal data is usually guaranteed between the employer, computer center or service provider.

However, the continuing practice of authenticating and transferring data in clear text form is a frightening phenomenon. Most directly affected are the commonly used services **`Post Office Protocol`** **`(POP)`** for retrieving mail and **`telnet`** for logging on to remote systems. Using these methods, user information and data considered sensitive, such as the contents of a letter or a chat via the talk command, travel openly and unsecured over the network. For one thing, this encroaches on the user's privacy and, for another, it leaves such access methods open to misuse. Such access is often used, especialy to attack other systems from there, or to obtain administrator or root permissions on this system.

The vulnerability of a system becomes especially apparent if `'root'` tasks of this nature are carried out on a remote network. Would you announce where you hide your extra apartment key over the radio?

Any device involved in data transfer or operating on the local network such as firewall, router, switch, mailserver, workstation, etc., can also access the data. In contrast to someone else opening and sabotaging your mail at the post office, for example, there is no way a recipient can be aware of the copying or altering of his or her digital data.

There are laws that basically prohibit such behavior, indeed it is liable to prosecution, but one should still not rely on the relatively negligible chances of such operations being uncovered, as there are numerous ways data can be hacked.

In respect to these risks, dropping the network connection remains the only option, but it is neither feasible nor desirable in most cases. Just as shutting off the irreplacable services likewise vulnerable to eavesdropping is not a solution.

The ssh software provides the only preferrable alternative. Complete authentication, usually username and password as well as the communication is encrypted here. Even here, snatching transferrable data is possible, but the contents can at least not be deciphered by intruders without the missing key. This enables secure communication via unsecured networks such as the Internet.

SuSE Linux includes the ssh openssh packages in the sec series. Basically, both packages provide the same functionality. When deciding between one or the other product, you should consider that, on the one hand, ssh may not be implemented in the commercial arena due to its availability at no cost under the stipulation of an Open Source license. You can find more information on this under `/usr/share/doc/packages/ssh/COPYING` after having installed the ssh package. On the other hand, the openssh package is also provided which has been developed for several years. It is under the auspices of a less restrictive license. SuSE Linux uses it in all of its package default selections. To this end, the following sections will refer to OpenSSH.

D.2.1 The OpenSSH Package

As soon as you have installed the openssh package, the following programs will be available to you: ssh, scp and sftp as alternatives to telnet, rlogin, rsh, rcp and ftp.

D.2.2 The ssh Program

With the ssh program, you can log on to remote systems and work interactively there. It is thus a replacement for both telnet and rlogin. Due to its similarity to rlogin, the symbolic name slogin similarly corresponds to ssh. For example, you can log in to the host `helios` by entering

```
hannah@earth:~> ssh sun
```

After the command has run its course, you will be asked for a password on the `helios` system with

```
hannah@sun password:
```

Following successful authentication, you can work from the command line there, e. g. with **`ls`**, or interactively with the SuSE administration program YaST.

If the local username is different from the remote username, you can log in using a different login name via

```
hannah@earth:~> ssh -l augustine sun
```

or

```
hannah@earth:~> ssh augustine@sun
```

Furthermore, ssh offers the familiar option from rsh of running commands on another system. In the following example, we will run the command **`uptime`** is run on the host `helios` and a directory with the name `tmp` is created. The program output is displayed on the local terminal of the host `earth`.

```
hannah@earth:~> ssh sun ’uptime; mkdir tmp’
hannah@sun password:
  1:21am  up  2:17,  9 users,  load average: 0.15, 0.04, 0.02
```

Apostrophes are necessary here to summarize the commands. Only then can the second command be run on the host sun in.

D.2.3 scp – Secure Copy

Using scp, copy the files to a remote machine. scp is the most secure and encrypted substitute for rcp. For example,

```
hannah@earth:~> scp MyLetter.tex sun:
```

copies the file `MyLetter.tex` from the machine `earth` to the machine sun. If the usernames involved on `earth` and sun differ, scp will resort to the **`username@machine`** format already mentioned in reference to the ssh command. An **`-l`** option does not exist here.

After it asks for the password, scp will start the data transfer and will show a series of stars, gradually marking the progress from left to right. In addition, the estimated time of arrival will be shown on the right margin. All output can be suppressed by giving the option **`-q`**.

scp provides, along with copying individual files, a recursive copying feature for entire directories.

```
hannah@earth:~> scp -r src/ sun:backup/
```

copies the entire contents of the directory `src/` including all sub-directories to the machine sun. Because the directory name `backup` is specified (after the hostname sun:), data is stored in the sub-directory `backup` on the machine sun. If this sub-directory does not exist yet, it will automatically be created by this command.

Via the option **`-p`**, scp can obtain a time stamp of the files. **`-C`** compresses the data transfer. This minimizes the data volume being transferred, but creates more demand on the processor. The latter concern is negligible in light of modern processor capabilities.

D.2.4 sftp - Secure File Transfer

Alternatively, sftp can be used for secure file transfer. During the session, sftp provides many of the same commands familiar from ftp. This may be an advantage over scp, especially when transferring data for which the filenames are unknown.

D.2.5 The SSH Daemon (sshd) – Server-Side

To implement ssh and scp, the ssh client programs, an ssh daemon, a server, have to be running in the background. This waits for its connections on
`TCP/IP port 22`.

The ssh daemon is a component of the ssh package and is automatically started on a SuSE Linux system in `runlevel 3` and `5` . The variable `START_SSHD` is set to `yes` in `/etc/rc.config`.

The daemon generates two key pairs when starting for the first time. The key pairs consist of a private and a public key. Therefore, this procedure is referred to as public key-based. To guarantee the security of the communication via ssh, only the system administrator can see the private key files. The file permissions are restrictively defined by the default setting. The private keys are only required locally by the ssh daemon and cannot be given to anyone else.

On the other hand, the public key components (possibly a file recognizable by the name extension `.pub`) are sent to communication partners and are readable for all users.

A connection is introduced by the ssh client. The waiting ssh daemon and the requesting ssh client exchange identification data comparing the protocol and software versions and to prevent connection to the wrong port. Since a child process of the original ssh daemon is the one that answers, several ssh connections can be made simultaneously.

The server will then send its public **host key** and a **server key**, regenerated by the ssh daemon every hour. Both allow the ssh client to encrypt a currently available session key and then to send it to the ssh server. The ssh client also supplies the server with the preferred encryption method or cipher.

The private host and server keys absolutely necessary for decoding the session key cannot be derived from the public parts. Only the ssh daemon contacted can decipher the session key using its own keys (see also `/usr/share/doc/packages/openssh/RFC.nroff`).

This initial connection phase can be followed closely using the ssh client program's error search option **-v**. It will end with the confirmation output of the ssh daemon, "Received encrypted confirmation.".

By way of client-side storage of all public host keys following initial contact in `~/.ssh/known_hosts`, such "man-in-the-middle" access attempts can be prevented. ssh servers which try to fraudulently use names and IP addresses of others will be exposed by a clear indicator. Either they will be noticed due to a host key which differs from `~/.ssh/known_hosts`, or they will not be able to decipher the session key in the absence of an appropriate private counterpart.

It is recommended that you securely archive the private and public keys stored in `/etc/ssh/` externally. In this way, key modifications can be detected and the old ones saved again after re-installing them. The latter action spares users from the unsettling warning. If it is verified that, despite the warning, it is indeed the correct ssh server, the existing entry regarding this system will have to be removed from `~/.ssh/known_hosts`.

D.2.6 SSH Authentication Mechanisms

Now, the actual authentication will take place which, in its simplest form, consists of entering a password as mentioned above.

The aim of ssh was to introduce a secure software that is also easy to use. As with the rsh and rlogin programs, soon to be replaced, ssh must also be able to provide an authentication method which is simple to use on a daily basis.

ssh accomplishes this by way of another key pair generated by the user. The ssh package also provides a help program, ssh-keygen, for this. After entering

```
hannah@sun:~> ssh-keygen
Generating RSA keys:
```

the key pair will be generated after a minute or two. You will be prompted for the base filename where you want to store the keys:

```
Enter file in which to save the key (/home/hannah/.ssh/identity):
```

Confirm the default setting so that you end up with this request for a passphrase:

```
Enter passphrase (empty for no passphrase):
```

Even if the software suggests an empty passphrase, a text from ten to thirty characters is recommended for the procedure described here. Use as short and simple a phrase as possible. After you have entered this successfully, you will be prompted to confirm this by entering it again. Subsequently, you will see the output of the location where the private and public keys are stored, in our example, the files `identity` and `identity.pub`, .

```
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 hannah@sun
```

When the private key (identity) is generated and is located on a system you are not administering or when retrieving your user directory over the NFS, it is especially important to use a passphrase. Use **`ssh-keygen -p`** to change your old passphrase.

Copy the public key component (identity.pub) to the remote machine and save it there at the location `~/.ssh/authorized_keys`. You will be asked to authenticate yourself with your passphrase the next time you attempt to make the connection. If this does not occur, verify the location and contents of the aforementioned files.

In the long run, this procedure is more troublesome than giving your password each time. To handle this, the ssh package provides another help program, the ssh-agent, which retains the private keys for the duration of an “X session”. The entire X session will then be started as a child process of ssh-agents. The easiest way to do this is to set the variable `usessh` at the beginning of the `.xsession` file to `yes` and to log in via a display manager such as KDM or XDM. Alternatively, you can enter ssh-agent startx.

As soon as you have started your X session, release your private key via ssh-add. Unless ssh-add can access a terminal, e. g. is started via a menu or is re-routed by **`</dev/null`**, a graphical entry screen will appear, `x11-ssh-askpass`.

Now you can use ssh or scp as you normally would. Given that you have assigned yourself a private key as described above, you should no longer be prompted for your password.

Be sure to exit your X session or to lock your screen via a password lock program such as xlock when leaving your machine.

D.2.7 X, Authentication and other Forwarding Mechanisms

Beyond the previously described security-related improvements, ssh also simplifies the usage of remote X applications. If you call up **`ssh`** with the option **`-X`**, the DISPLAY variable will automatically be set on the remote machine and all X output will be piped to the remote machine over the existing ssh connection. At the same time, this convenient function prevents the authorized intrusion discussed above when X applications are started remotely and locally viewed.

By adding the option **`-A`**, the ssh-agent authentication mechanism will be carried over to the next machine.This way, you can view these applications from one machine to the other without having to enter a password. However, only if you have previously assigned the public key to the destination hosts in question and have properly saved them there.

To be safe, both mechanisms have been de-activated in the default settngs but can be permanently activated at any time in the system-wide configuration file `/etc/ssh/sshd_config` or the user’s `~/.ssh/config`.

Similar to X forwarding, ssh can be also used to pipe any TCP/IP connection. As an example, SMTP and POP3 port pipes:

```
root@earth:~ # ssh -L 25:sun:25 sun
```

Here, each connection is piped to "earth port 25", SMTP to the SMTP port on helios via an encrypted channel. This is especially usefule for those using SMTP servers without SMTP-AUTH or POP-before-SMTP features. E-mail can be transferred for delievery via the "home" mail server in this manner from any arbitrary location connected to a network.

In a similar manner, the following command forwards all port 110 and POP3 requests on earth to the POP3 port of helios

```
root@earth:~ # ssh -L 110:sun:110 sun
```

Both examples must be carried out by user 'root' since the connection is made to privileged local ports. E-mail is sent and retrieved, as is customary, by normal users in an existing ssh connection. The SMTP and POP3 host must be configured on the localhost for this reason.

Additional information can be found in the manual pages for each of the programs described above and also in the files under /usr/share/doc/packages/openssh.

D.3 Security as a Matter of Confidence

D.3.1 Basic Considerations

One of the main characteristics of a Linux/UNIX system is its ability to handle several users at a time and to allow these users to perform several tasks at once (multi-user, multi-tasking), all on the same computer. Moreover, we expect the operating system to be network transparent, i. e. often we wouldn't even want to know whether the data or applications that we're using are provided locally from our machine or made available from somewhere else.

This particular feature of having several users with multiple tasks running on one system leads to the obvious need for a mechanism to keep these users and their data separate from each other. Now this is undeniably an area where many varied considerations have to be made and some of them may involve certain emotional aspects. Therefore, it's really important to give some serious thought to security and privacy issues.

The term "data security" isn't really all that new; in fact, it was already in use before computers could be linked through networks. Just like today, the most important concern was the ability to keep data available in spite of a lost or otherwise damaged data medium (a hard disk in most cases), probably more than the possibility that such a defect could bring down the wider infrastructure for some time. This chapter of the SuSE manual is primarily focused on confidentiality issues and on ways to protect the privacy of users, but it cannot be stressed enough that a comprehensive security concept should always include procedures to have a regularly updated, workable and tested backup in place. Without this, you could have a very hard time getting your data back – and not only in the case of some hardware defect, but also if the suspicion creeps in that someone has gained unauthorized access and tampered with your files.

D.3.2 Local Security and Network Security

If we step back one moment, it seems only logical that data stored on a computer can only be accessed if it has actually been made available somehow. Assuming that we don't want to keep our data under lock and key forever, we can give access to it in several ways:

- Some person is sitting in front of a computer and is talking to the user of the data on the phone
- directly from the console of a computer (physical access)
- over a serial line
- using a network link

Obviously all these cases are similar in that, as a user, you'd have to authenticate before being able to access the resources or data in question. In other words, there should be some access rule in place which requires you to provide proof of your identity to gain access to the data or computing resources requested. Now a web server might be less restrictive in this respect, but you still wouldn't want it to disclose all your personal data to any surfer out there. On a SuSE system, a few tweaks are probably sufficient to make it boot right into your desktop without even asking for a password – only in most cases that wouldn't be such a good idea. Because in a way, with the login procedure, you “extend” your personality and its actions onto the computer that you're controlling and, with a few exceptions, you wouldn't want any other person to do so on your behalf.

In the list above, the first case is the one where the highest amount of human interaction is involved, like when you're contacting a bank employee and need to prove that you're the person owning that bank account. So you'll be asked to provide a signature, a PIN or a password to prove that you're the person you claim to be. In some cases (which are not really typical for computers, operating systems and networks) it might be possible to elicit some intelligence from an informed person just by mentioning known bits and pieces here and there in a canny way or to worm one's way into the confidence of that person by applying clever rhethoric. The victim could be led to gradually reveal more and more information, maybe without even becoming aware of it. Some people are rather unmindful of what they say or act unconsciously in the way they give answers, so that even a question which they believe was left unanswered might provide enough information to proceed with an even more precise question. Piece after piece gets added to the puzzle until the picture is nearly complete (“No, Mr. Smith is on vacation right now, it's at least three weeks before he'll be back in. He's not my boss anyway, you know he's up there in the fourth story while I'm here in the third!”). In hacker circles, this is called “social engineering”. You can only guard against this by educating people and by dealing with language and information in a conscious way. Before breaking into computer systems, attackers often try to target receptionists, service people working with the company or even family members and, in many cases, such an attack based on social engineering will only be discovered at a much later time.

A person wanting to obtain (unauthorized) access to your data could also use the conventional, very traditional way and try to get at your hardware directly. Therefore, the machine should be protected against any tampering so that no one can remove, replace or cripple its components. This also applies to the system as a whole (including the backups!) and even any network cable or the power cord. Likewise, it might be necessary to secure the boot procedure, as there are some well-known key combinations provoking special reactions on bootup. You can protect yourself against this by setting passwords for the BIOS and the bootloader.

Serial ports connecting with terminals are still being used in many places but are rarely installed with new systems anymore. With regard to data access, serial terminals are a special case, i. e. unlike network interfaces they don't rely on a network protocol to communicate with the host. A simple cable (or maybe an infrared port) is used to send plain characters back and forth between the devices. The cable itself is the weakest point of such a system: With an old printer connected to it, it's really easy to record anything that runs over the wires. What can be achieved with a printer can also be done in other ways, depending on the effort that goes into the attack.

Networks make it easier for us to access data remotely, but they do this with the help of network protocols which are often rather complex. This might seem paradoxical at first, but is really indispensable if you want to work from any location so wish to remotely control a computer or to retrieve data from it. It is necessary to have abstract, modular designs with layers that are more or less separate from each other. We rely on such designs in many daily computing situations. Modularity means that your text processor, for instance, doesn't need to know about the kind of hard disk that you're using, or that your e-mail program doesn't have to care whether you have a modem or an ethernet card. The components of your operating system (Linux in our case) care about the details of a certain set of functions and make these available to the system through a predefined interface. With this modularity, a text processor or a mail user agent (MUA) can function on a variety of hardware platforms, while, at least in principle, you could also use these programs from any place in the world.

>From the data perspective and as a consequence of the way in which files are handled and accessed, there is no difference between opening a file from a command line or telling a web server (e. g. Apache) to do so for you so that the file is sent out over the network and displayed by a browser. In both cases, a user with the proper permissions has opened the file and seen at least parts of its contents. There are other possibilities: to read the file, you could also log in via a network (e. g. using a telnet program or with a secure shell client – which is actually a much better option as ssh encrypts all network traffic). To do so, however, you'd have to take several hurdles: first, you are required to log in at the host over the network, i. e. to authenticate yourself (provide proof of your identity). Once you are in, you face the file permissions restricting your actions on that system.

Given that opening a file locally on a host requires other access rules than opening a network connection with a server on a different host, we need a distinction between local security and network security. The line between the two can be drawn where data have to be put into packets, in order to be sent somewhere else before they can be used.

Local Security

As we've mentioned above, local security starts with the physical environment in the location where the computer is running. Let's assume that your machine is set up in a place where security is in line with your expectations and needs. Now if you put yourself in the place of an attacker for a moment, it's easy to see that as long as we're talking about "local security", the task is to keep users separate from each other, so that no user can assume the permissions of another one. This is a general rule to be observed, but it's especially true for the `'root'` account, the supreme power on any system. Users that become `'root'` have the ability to take over any other local user account without knowing the password, and thus read any locally stored file.

For an attacker who obtained access to local resources from the command line, there's certainly no shortage of things that could be done to compromise the system:

Passwords

On a Linux system, it is, of course, not the case that passwords (and you *do* use passwords on your machine, don't you?) are stored as plain text to just compare the text string with what has been entered at the prompt. If things were like that, all accounts on your system would be compromised as soon as someone grabs the corresponding file. Rather, the system encrypts your password; each time it is entered, it gets encrypted again, and is compared with the previously encrypted string. Naturally, this will only work if the encrypted password cannot be reverse-computed into the original text string. This is actually achieved by a special kind of algorithm which is also called "trapdoor algorithm" because it only works in one direction. An attacker who has obtained the encrypted string won't be able to get at your password by simply applying the same algorithm once again. Instead, it would be necessary to test all the possible character combinations, until a combination is found which looks like your password when encrypted. As you can imagine, with passwords that are eight characters long, there are quite a number of possible combinations to calculate.

In the seventies, it was argued that this method would be more secure than others due to the relative slowness of the algorithm used, so that it took one or more more seconds to encrypt just one password. In the meantime, however, PC's have become powerful enough to do several hundred thousand or even million encryptions per second, which is why we have two additional requirements. First, encrypted passwords should not be visible to regular users (therefore, regular users don't have read permission for `/etc/shadow`). Second, even if passwords became visible because of some error, they shouldn't be easy to guess. Accordingly, it's not really useful to "translate" a password like "tantalise" into "t@nt@1ls3". Things like that where you just swap a few letters are a piece of cake for password cracking programs which use dictionaries to guess words. A better way would be to make up a word with no common meaning, something which only makes sense to you personally (but not in the way you set a cipher on the combination lock of your suitcase!), like the first letters of the words of a sentence. As an example, one could use a book title, such as "The Name of the Rose" by Umberto Eco. This would give the following well-formed password: "TNotRbUC9". By contrast, passwords like "beerbuddy" or "jasmine76" are easily guessed even by someone who has only some casual knowledge about you.

The boot procedure

You should configure your system such that it cannot be booted from a floppy or from CD, either by removing the drives entirely or by setting a BIOS password and configuring the BIOS to allow booting from a hard disk only.

Normally, a Linux system will bootstrap with the help of a boot loader allowing you to pass on additional options to the boot kernel. These options are crucial to your system's security: Not only does the kernel itself run with root permissions, but it is also the first authority to grant root permissions on system startup. You can prevent others from using such parameters on bootup by using the options "restricted" and "password=your_own_password" in `/etc/lilo.conf`. Don't forget to execute the command **`lilo`** after making any changes to `/etc/lilo.conf` and look for any unusual output that the command might produce. If you should forget this password, you'll have to know the BIOS password and boot from CD to read the entry in `/etc/lilo.conf` from a rescue system.

File Permissions

As a general rule, one should always work using the most restrictive privileges possible for a given task. For instance, it's definitely not necessary to be root to read or write one's e-mail. If the mail program (or MUA = mail user agent) that you're using has a bug, this bug could be used as an attacking mechanism which will act with exactly the permissions of the program when it was started. By following the above rule, you'll mimimize the risk of that happening.

The permissions of the well over 200,000 files included in a SuSE distribution are carefully chosen. A system administrator who installs additional software or other files should take great care when doing so, especially when setting the permission bits. Experienced and security-conscious system administrators always use the **`-l`** option with the command **`ls`** to get an extensive file list, which allows them to detect any wrong file permissions immediately. It should also be noted that an incorrect file attribute does not only mean that files could be overwritten or deleted. It also allows for a situation where files that have been changed could be executed with the permissions of `'root'` or, in the case of configuration files, that programs could use such files with the permissions of `'root'`. This increases the possibility that a potential attacker will be successful quite a bit. Attacks of this kind are called cuckoo's eggs, because the program (the egg) is hatched (executed) by a different user (bird), just like a cuckoo would trick other birds into hatching its eggs.

A SuSE system includes the files `permissions`, `permissions.easy`, `permissions.secure` and `permissions.paranoid`, which are all in the directory `/etc`. The purpose of these files is to define special permissions, such as world-writable directories or setuser ID bits (if the latter is set, the corresponding program won't run with the permissions of the user that has launched it, but with the permissions of the file owner, which will be `'root'` in most cases). An administrator may use the file `/etc/permissions.local` to add his own settings. The variable **`PERMISSION_SECURITY`**, which is set in `/etc/rc.config`, defines which of the above files is used by SuSE's configuration programs to set permissions accordingly. As a more convenient way to select the file, you can use the submenu '`Security`' in YaST1 or YaST2. To learn more about the topic, please read the comments in `/etc/permissions`, or consult the manual page of **`chmod`** (**`man chmod`**).

File race conditions

Let's assume that a program wants to create a file in a directory which is world-writable (such as `/tmp`). First the program checks whether the file already exists and, if that is not the case, it gets created. However, between checking and file creation, there is a very short moment which can be used by an attacker to create a symbolic link, i. e. a pointer to another file. The program may then be tricked into following the symbolic link, overwriting the target file with its own permissions. This is called a race because the interval during which the attacker can create a "symlink" is very short indeed. The race is only possible if the checking and file creation procedure is not made to be indivisible or atomic. If the race is allowed to take place at all, there's a chance that it may be won by the attacker, i. e. it's all a matter of probability.

Buffer overflows, format string bugs, signed and unsigned bugs

Special care must be taken whenever a program is supposed to process data which can or could be changed by a user at will, though this is more of an issue for the programmer of an application than for regular users. The programmer has to make sure that his application will interpret data in the correct way, without writing them into memory areas which are too small to hold them. Also, the program should hand over data in a consistent manner, using the interfaces defined for that purpose.

A buffer overflow can happen if the actual size of a memory buffer is not taken into account when writing to that buffer. There are cases where this data (as generated by the user) use up some more space than what is available in the buffer. As a result, data is written beyond the end of that buffer area, which under certain circumstances makes it possible that a program will execute program sequences influenced by the user (and not by the programmer), rather than just processing user data. A bug of this kind may have serious consequences, in particular if the program is being executed with special privileges (see the above section on this topic). "Format string bugs" work in a slightly different way, but again it's the user input which could lead the program astray.

In most cases, these programming errors are exploited with programs that are executed with special permissions, i. e. setuid and setgid programs – which also means that you can protect your data and your system from such bugs by removing the corresponding execution privileges from programs. Again, the best way is to apply a policy of using the lowest possible privileges (see the section on the topic of permissions).

Given that buffer overflows and format string bugs are bugs related to the handling of user data, they are not only exploitable if access has been given to a local account. Many of the bugs that have been reported can also be exploited over a network link. Accordingly, buffer overflows and format string bugs should be classified as being relevant for both local and network security.

Viruses

Contrary to what some people will tell you, there *are* viruses that run on Linux. However, the viruses that are known were released by their authors as "proof of concept", meaning that they were written in order to prove that the technique works as intended. On the other hand, none of these viruses have been spotted "in the wild" so far.

Viruses wouldn't be able to survive and spread without a host on which they can live. In our case, the host would be a program or an important storage area of the system, such as the master boot record, which needs to be writable for the program code of the virus. Owing to its multiuser capability, Linux can restrict write access to certain files, which is the case especially with system files. Therefore, if you did your normal work with `'root'` permissions, you'd increase the chance of the system being infected by a virus. By contrast, if you follow the principle of using the lowest possible privileges as mentioned above, chances are that you'll have a hard time getting any virus at all. Apart from that, you should never rush into executing a program that you've gotten from some Internet site which you don't really know. SuSE's RPM packages carry a cryptographic signature, as a digital label that the necessary care was taken to build them. Viruses are a typical symptom that the administrator (or the user, for that matter) lacks the required security awareness, thus putting at risk even a system that should be highly secure by its very design.

Viruses should not be confused with worms which belong to the world of networks entirely and don't need a host for spreading.

Network Security

As long as we were focused on local security, our main goal was to keep users apart *within* a system, and in particular the user `'root'`. By contrast, network security means that the system as whole should be protected from an attack originating in the network. Though the typical login procedure requires entering a user name and a password, this kind of user authentication is mostly related to local security. However, in the special case of logging in over a network, we need to consider both the actions happening before the authentication proper (network security) and anything that happens afterwards (local security).

X Window system (X11 authentication)

As we've mentioned at the beginning, network transparency is one of the central characteristics of a UNIX system. X11, the windowing system of UNIX operating systems, can make use of this feature in an impressive way. With X11, it's basically no problem to log in at a remote host and start a graphical program which will then be sent over the network to be displayed on your computer. The protocol to communicate between the X application and the X server (which is the local process that draws the windows with the help of your video card) is relatively lightweight as far as bandwidth usage is concerned. This is because the protocol was designed in the eighties when network bandwidth was still a scarce resource.

Now if we want an X client to be displayed remotely using our X server, the latter is supposed to protect the resource managed by it (i. e. the display) from unauthorized access. In more concrete terms, certain permissions must be given to the client program. With the X Window System, there are two ways to do so, called host-based access control and cookie-based access control. The former relies on the IP address of the host where the client is supposed to run; the program to control this is `xhost`. What `xhost` does is to enter the IP address of a legitimate client into a tiny database belonging to the X server. Note, however, that relying on IP addresses for authentication is not exactly very secure. For instance, if there were a second user working on the host sending the client program, that user would have access to the X server as well – just like someone stealing the IP address. Because of these shortcomings, we won't describe this authentication method in more detail here, but you can learn about the way it functions if you read the manpage of **`xhost`** (which includes a similar warning).

In the case of cookie-based access control, a character string is generated which is only known to the X server and to the legitimate user, just like an ID card of some kind. This cookie (the word goes back not to ordinary cookies but to Chinese fortune cookies which contain an epigram) is stored upon login in the file `.Xauthority` in the user's home directory and is available to any X Window client wanting to use the X server to display a window. The file `.Xauthority` can be examined by the user with the tool `xauth`. If you were to rename `.Xauthority` or if you deleted the file from your home directory by accident, you would not be able to open any new windows or X clients. You can read more about X Window's security mechanisms in the manpage of `Xsecurity` (**`man Xsecurity`**).

Apart from that, `ssh` (secure shell) can be used to completely encrypt a network connection and forward it to an X server transparently (i. e. without the encryption mechanism being perceived by the user). This is also called "X forwarding". X Forwarding is achieved by simulating an X server on the server side and setting a DISPLAY variable for the shell on the remote host. Before being displayed, the client opens a connection with sshd (secure shell daemon, the server side program) which then gets the connections through to the real X server. If your setup requires that X clients are displayed remotely, you should consider using `ssh` and have a closer look at it. The manpage of `ssh` will tell you more about the functionality of this program.

Caution

If you don't consider the host where you log in to be a secure host, you shouldn't use X forwarding. With X forwarding enabled, an attacker could authenticate via your ssh connection to intrude on your X server and sniff your keyboard input, for instance.

Further details about `ssh` can be found in section D.2 on page 147 of this book.

Buffer overflows and format string bugs

As we've said in the section on local security, buffer overflows and format string bugs should be classified as issues concerning both local and network security. Just like with the local variants of such bugs, buffer overflows in network programs, when successfully exploited, are mostly used to obtain `'root'` permissions. Even if that is not the case, an attacker could use the bug to gain access to an unprivileged local account to exploit any other (local) vulnerabilities which might exist on the system.

Buffer overflows and format string bugs exploitable over a network link are certainly the most frequent form of remote attacks in general. Exploits for these (i. e. programs to exploit these newly found security holes) are often posted on the security mailing lists; they can be used to target the vulnerability without knowing about the details of the code. Over the years, experience has shown that the availability of exploit codes has contributed to more secure operating systems, which is obviously due to the fact that operating system makers were forced to fix the problems in their software. With free software, anyone has access to the source code (SuSE Linux comes with the source of all the programs that make it available) and anyone who finds a vulnerability and its exploit code can submit a patch to fix the corresponding bug.

DoS - Denial of Service

The purpose of this kind of attack is to force down a server program (or even an entire system), something which could be achieved by various means: overloading the server, keeping it busy with garbage packets, or exploiting a remote buffer overflow (i. e. a buffer overflow which cannot be exploited to run programs on the remote site).

Often a DoS attack is done with the sole purpose of making the service disappear. However, once a given service has become unavailable, communications could become vulnerable to so-called man-in-the-middle attacks (sniffing, TCP connection hijacking, spoofing) and DNS poisoning, which are explained below.

Man in the middle: sniffing, tcp connection hijacking, spoofing

In general any remote attack performed by an attacker who puts himself between the communicating hosts is called a "man-in-the-middle attack". What almost all types of man-in-the-middle attacks have in common is that the victim won't normally become aware that there's something going on. There are many possible variants: For instance, the attacker could pick up a connection request and forward that to the target machine himself. Now the victim has unwittingly established a connection with the wrong host, because the other end is posing as the legitimate destination machine. The simplest form of a man-in-the-middle attack is called "sniffer", i. e. the attacker is "just" listening to the network traffic passing by. As a more complex attack, the "man in the middle" could try to take over an already established connection (hijacking). To do so, the attacker would have to analyze the packets for some time to be able to predict the TCP sequence numbers belonging to the connection. When the attacker finally seizes the role of the target host, the victim will notice because he or she gets an error message saying the connection was terminated due to a failure.

What often makes things easier for attackers is the fact that there are protocols which are not secured against hijacking through encryption and which don't perform an authentication procedure upon establishing the connection. Finally, we want to mention "spoofing", which is an attack where packets are modified to contain counterfeit source data, i. e. mostly the IP address. Most active forms of attack rely on sending out such fake packets – something which on a Linux machine can only be done by the superuser (`'root'`).

Many of the attacks mentioned are carried out in combination with a DoS. If an attacker sees an opportunity to abruptly bring down a certain host (even if only for a short time), that will make it easier for him to push the active attack, because the host won't be able to interfere with the attack for some time.

DNS poisoning

DNS poisoning means that the attacker corrupts the cache of a DNS server by replying to it with spoofed DNS reply packets, trying to get the server to send certain data to a victim who is requesting information from that server. To foist such false information onto the server in a credible way, normally the attacker must have received and analysed some packets from it. Given that many servers are configured to maintain a trust relationship with other hosts (which is based on IP addresses or hostnames), such an attack may be successful in a relatively short time, but, on the other hand, it also requires quite an effort. In any case, the attacker will need a good understanding of the actual structure of the trust relationships between hosts. From the attacker's viewpoint, targeting a DNS server with a well-timed DoS attack to spoil its information is often the first thing to consider.

Again, you can protect yourself by using encrypted connections which are able to verify the identity of the hosts to which you want to connect.

Worms

Worms are often confused with viruses, but there's a clear difference between the two. Unlike viruses, worms don't need to infect a host program to live. Rather, they are specialized to spread as quickly as possible on network structures. The worms that appeared in the past, such as Ramen, Lion or Adore, make use of well-known security holes in server programs like bind8 or `lprNG`. Protecting against worms is relatively easy. Given that some time will elapse between the discovery of a security hole and the moment the worm hits your server, there is a good chance that an updated version of the program affected will be available on time. Of course, that is only useful if the administrator actually installs the security updates on the systems in question.

D.3.3 Some General Security Tips and Tricks

Information: To handle security competently, it's important to keep up with new developments and to stay informed about the latest security issues. One very good way to protect your systems against problems of all kinds is to get and install the updated packages recommended by security announcements as quickly as possible. SuSE security announcements are published on a mailing list to which you can subscribe by following the link `http://www.suse.`

de/security. The list, which has the address suse-security-announce@suse.de, is a first-hand source of information with regard to updated packages and includes members of SuSE's security team among its active contributors.

The mailing list suse-security@suse.de is a good place to discuss any security issues of interest; you can subscribe to it under the URL as given above for suse-security-announce@suse.de.

bugtraq@securityfocus.com is one of the best-known security mailing lists worldwide. We really recommend that you read this list which receives between 15 and 20 postings per day. More information can be found at http://www.securityfocus.com.

The following is a list of rules which you may find useful to deal with basic security concerns:

- According to the rule of using the most restricive set of permissions possible for every job, avoid doing your regular jobs as 'root'. This reduces the risk of getting a cuckoo's egg or a virus and protects you from your own mistakes.
- If possible, always try to use encrypted connections to work on a remote machine. Use "ssh" (secure shell) to replace telnet, ftp, rsh and rlogin.
- Avoid using authentication methods based on IP addresses alone.
- Try to keep the most important network-related packages up-to-date and subscribe to the corresponding mailing lists to receive announcements on new versions of such programs (e. g. BIND, sendmail, ssh). The same should apply to software which is relevant to local security.
- Change the /etc/permissions file which best fits your needs to optimize the permissions of files which are crucial to your system's security. If you remove the setuid bit from a program, it might well be that it cannot do its job anymore in the way it is supposed to. On the other hand, consider that in most cases the program will also have ceased to be a potential security risk. You might take a similar approach with world-writable directories and files.
- Disable any network services that you don't absolutely require for your server to work properly. This will make your system safer, plus it prevents your users from getting used to a service that you had never intended to be available in the first place (the so-called legacy problem). Open ports (those having the socket state LISTEN) can be found with the program netstat. As for the options, we suggest that you use **netstat -ap** or **netstat -anp**. The **-p** option allows you to see which process is occupying a port under which name.

  Compare the netstat results with those of a thorough portscan done from outside your host. An excellent program for this job is nmap, which does not only check out the ports of your machine, but also draws some conclusions as to which services are waiting behind them. However, portscanning may be interpreted as an aggressive act, so don't do this on a host without the explicit approval of the administrator. Finally, remember that it's important that you not only scan TCP ports but also UDP ports (options **-sS** and **-sU**).

- To monitor the integrity of the files of your system in a reliable way, you should use the program `tripwire`. Encrypt the database created by `tripwire` to prevent it from being tampered with. Furthermore, you should keep a backup of this database available outside your machine, i. e. stored on an external data medium not connected to it by some network link.

- Please take proper care when installing any third-party software. There have been cases where a cracker had built a trojan horse into the tar archive of a security software package – which was fortunately discovered very quickly. If you install a binary package, you should have no doubts about the site from which you have downloaded it.

  Note that SuSE's RPM packages are gpg signed. The key used by SuSE for signing reads as follows:

  ID:9C800ACA 2000-10-19 SuSE Package Signing Key <build@suse.de>

  Key fingerprint = 79C1 79B2 E1C8 20C1 890F 9994 A84E DAE8 9C80 0ACA

  The command **`rpm -checksig package.rpm`** shows you whether the checksum and the signature of a (not yet installed!) package are correct. Beginning with version 7.1 of SuSE Linux, you can also find the key on the first CD of the distribution and on most keyservers worldwide.

- Check your backups of user and system files regularly. Consider that if you don't test whether the backup will work or not, it might actually be worthless.

- Check your log files, if possible, with a small script that you write to search for suspicious entries. Admittedly, this is not exactly a trivial task. In the end, only you can know which entries are unusual and which are not.

- Use `tcp_wrappers` to restrict access to the individual services running on your machine, so that you have explicit control over which IP addresses can connect to a service. For further information on `tcp\_wrappers`, consult the manual page of `tcpd(8)` and `hosts\_access` (**`man tcpd`**, **`man hosts_access`**).

- You can use SuSEFirewall to enhance the security provided by `tcpd` (tcp_wrapper). However, if you don't intend to provide any services from your machine, you should probably install Personal-firewall instead. Configuring Personal-firewall is as simple as providing the name of the network interface on which inbound traffic should be rejected. You can find more information on this in the files `/sbin/SuSEpersonal-firewall` and `/etc/rc.config.d/security.rc.config`, as well as in section D.1 on page 139.

- Design your security measures to be redundant: a message that you see twice is much better than not getting any message at all. You would feel the same way when it comes to the way your co-workers communicate with you.

D.3.4 Using the Central Security Reporting Address

If you find a security related problem (please check the available update packages first), don't hesitate to write an e-mail to `security@suse.de`. Please include a detailed description of the problem and the version number of the package concerned. SuSE will try to send a reply as soon as possible. You are encouraged to pgp encrypt your e-mail messages. SuSE's pgp key is as follows:

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

This key is also available for download from: `http://www.suse.de/security`

E YaST and SuSE Linux License Terms

YaST Copyright (c) 1995 - 98 SuSE GmbH, Furth (Germany)

The object of this license is the YaST (Yet another Setup Tool) program, the name “YaST”, SuSE Linux the Linux Distribution of SuSE GmbH *and all other programs of SuSE GmbH under this license*, all programs derived from YaST or *another program under this license* and all works or names derived in full or in part thereof together with the use, application, archiving, reproduction and passing on of a program under this license, all programs derived from *a program under this license* and all works derived in full or in part thereof. The YaST program, any other program under this license and all sources are the intellectual property of SuSE GmbH within the meaning of the Copyright Law. The name YaST is a registered trademark of SuSE GmbH. In the following SuSE GmbH is the licensor and every user or processor of YaST *or any other program under this license*, works derived in full or in part thereof, together with every person who reproduces, distributes or archives YaST, SuSE Linux *or any other program under this license*, is the licensee of SuSE GmbH.

The following license terms are recognised as a result of the processing, use, application, archiving reproduction and dissemination *of the programs under this license*.

Only this license gives the Licensee the right to use reproduce, to distribute or to amend YaST, *any other program under this license* or works derived thereof. These actions are forbidden by the copyright act, if this license is not recognised. If this license is recognised and complied with in full, it is also valid even without the written consent of the Licensee.

1. Usage

   YaST, SuSE Linux *and any other program under this license* may be used for personal and commercial purposes if the copyright and license terms of the installed packages and programes are observed. The use of YaST *or any other program under this license*, even if a modified version is used, does NOT exempt in particular the Licensee from the duty to take due care with regard to the license terms of the packages or programs installed through YaST, *any other program under this license* or works based on it.

2. Processing

   All programs derived from YaST, *any other program under this license* and all works derived thereof, in full or parts thereof are to be filled on the opening screen with the clear information “Modified Version”. Moreover the operator give his name on the opening screen, stating that SuSE GmbH is not providing any support for the “Modified Version” and is excluded from any liability whatsoever. Every amendment to the sources which are not conducted by SuSE GmbH are deemed to be a “Modified Version”. The Licensee is entitled to change his copy from the sources of YaST *or any other program under this license*, whereby a work based on *one of these programs* is created, provided that the following conditions are satisfied.

a) Every amendment must have a note in the source with date and operator. The amended sources must be made available for the user in accordance with section 3) together with the unamended license.

b) The Licensee is obliged to make all work distributed by him which is derived as a whole or *in part from a program under this license* or parts thereof to third parties as a whole under the terms of this license without royalties.

c) The amendment of this license by a Licensee, even in part, is forbidden.

SuSE GmbH reserves the right to accept parts or all amendments of a modified version of any program under this license into the official version of the *concerned program* free of charge. The Licensee has no bearing on this.

3. Dissemination

   It is forbidden to reproduce or distribute data carriers which have been reproduced without authorisation for payment without the prior written consent of SuSE GmbH or SuSE Linux. Distribution *of programs under this license*, their sources, whether amended or unamended in full or in part thereof, and the works derived thereof for a charge require the prior written consent of SuSE GmbH.

   All programs derived *from programs under this license*, and all works derived thereof as a whole or parts thereof may only be disseminated with the amended sources and this license in accordance with 2b). Making YaST, or *any other program under this license* or works derived thereof available free of charge together with SuSE Linux on FTP Servers and mailboxes is permitted if the licenses on the software are observed.

4. Guarantee

   No guarantee whatsover is given for YaST, *any other program under this license or for works derived thereof* and SuSE Linux. The SuSE GmbH guarantee only covers fault-free data carriers.

   SuSE GmbH will provide *every program under this license* and SuSE Linux *AS IT IS* without any guarantee whatever that it is fit for a specific purpose or use. In particular SuSE is not liable for lost profit, savings not made, or damages from the claims lodged by third parties against the Licensee. SuSE GmbH is not liable either for other direct or indirect consequential losses, in particular not for the loss or production of recorded data.

   The observance of the respective licenses and copyrights of the installed software is incumbent solely upon the user of *the relevant program* and SuSE Linux.

5. Rights

   No other rights to YaST, *any other program under this license* or to SuSE Linux are granted other than negotiated in this license. An infringement against this license automatically terminates the rights of the Licensee. However the right of third parties who have received copies or rights under this license from the Licensee, are not terminated as long as all parts of his license are recognised and observed. If the Licensee is subject to conditions, or obligations as a result of a court judgement, patent terms, license terms, or another reason, and these conditions or obligations contradict this license as a whole or in part, the Licensee shall only be exempted in full or in part from this license and its terms with the express prior written consent of SuSE, SuSE is entitled to withhold its consent without giving reasons.

6. Additional restrictions

   If the distribution or use of YaST, *any other program under this license* and SuSE Linux or parts of SuSE Linux is restricted in a state either by patents or by interfaces protected by copyright, SuSE GmbH can specify an explicit geographic restriction of the distribution of the *concerned program* or parts of SuSE Linux, in which these states are fully or partially excluded from distribution. In such a case this license includes the whole or partial restriction as if it was written down in this license.

Index

symbols
SuSE Linux Solutions AG . . . 98
SuSE Linux Firewall on CD . 63

A
ACLs
  arranging 55
  defining 54
adding users
  problems 29
Apache 133, 136
AUTOPROTECT 67
awk 66

B
bibliography 101
BIND 28, 77, 103, 162
bind8 . . . . . . 40, 41, 64, 111, 161
boot parameters 83
business support 98

C
cachemgr 121
cachemgr.cgi 125
Calamaris 121, 133
chroot 79
communication analysis 91
communication matrix 91
compartment 79
configuration
  boot manager 23
  documentation 60
  editing 59
  saving 58
  SuSE Firewall Script 65
  testing 59
configuration disk 80
configuration files 26, 80
content filter 56, 78
CPU 125
cracklib 35

D
dhclient 146
DHCP 115
`dhcpcd` 116
dhcpd 116
`DHCPD_INTERFACE` 117
diald 146
DNS 77
  configuration 40
domain 31
Domain Name System . *see* NIS

F
`fas-devel` 19
fasd 34
firewall 139
  base configuration 36
  monitoring 60
  new configuration 35
  services 64
  setup 85
  start-up 86
  testing 86
Firewall
  Setups 14
Firewall Administration System
    9, 17, 34
  startup 34
  user fwadmin 33, 34
ftp 51, 148, 149
ftp proxy 52, 78
ftp-proxy-suite 64

G
gateway 32
grep 66

H
harden_suse 20
host.conf 27
  alert 28
  multi 28
  nospoof 28
  order 27
  trim 28
HOSTNAME 31
hosts 26, 28
`howtoen` 137, 142
HTTP proxy 53, 58, 77
httpf 64, 78

I
ifconfig 66
inetd 32
intrusion detection 93
IP address 32
ipchains 41, 64–66, 76
ipchains(8) 63, 64
ipchains-save 76

K
K51firewall 68
KDM 151
kernel caps 79

L
`libcinfo` 28
LILO 23
Linux training 99
linuxrc 20
Live Filesystem 96

M
magic user 78
mail 77
mail relay 46
masquerading 139

N
name resolution
  NIS 28
name server 28, 31
name service cache daemon . 29
name service switch 28
netmask 32
netstat 66
network 26
  configuration files 26
  manual configuration 25

networks . . . 26
nmap . . . 59
nscd . . . 29, 31
nscd.conf . . . 29
nsswitch.conf . . . 28

O
openssh . . . 64, 79, 82, 147, 148

P
package
  `dhcpcd` . . . 116
  `fas-devel` . . . 19
  `howtoen` . . . 137, 142
  `libcinfo` . . . 28
packet filter . . . 76, 139
pam_auth . . . 130
parent proxy configuration . . 57
personal firewall . . . 142
pidentd . . . 130
postfix . . . 46, 64, 68, 69, 77, 81
professional services . . . 98
ps . . . 94

R
RAM . . . 124
rcp . . . 148
resolv.conf . . . 31
rlogin . . . 148, 150
root password . . . 23
rsh . . . 148, 150

S
SaX . . . 25
scp . . . 148, 149, 151
sec . . . 147
secumod . . . 79
secure shell . . . 147
security . . . 152
  firewall . . . 139
security policy . . . 90
sed . . . 66
sendmail . . . 32, 162
series
  `doc` . . . 28, 142
  `n` . . . 18, 116, 135
  `zfw` . . . 18
  `zfw1` . . . 18
services . . . 91, 99
sftp . . . 148, 149
slogin . . . 148
smtpd . . . 68
Squid . . . 64, 68, 77, 78, 82, 121–129, 131–137
squidgrd . . . 121, 133, 135
SquidGuard . 129, 130, 135, 136
ssh 36, 49, 68, 69, 79, 147–152, 162
  configuration . . . 48
  keys . . . 48
ssh-add . . . 151
ssh-agent . . . 151
ssh-agents . . . 151
ssh-keygen . . . 150
`START_DHCPD` . . . 118
startup scripts . . . 32
startx . . . 151
support . . . 97
  commercial . . . 98
  professional services . . . 98
  services . . . 98, 99
SuSE
  services . . . 99
SuSE . . . 99
SuSE Firewall Adminhost 82, 85
SuSE firewall module . . . 41
SuSE Firewall Script . . . 65
SuSE Linux Firewall Adminhost 50
  new installation . . . 20
SuSE Linux Firewall on CD . . 9, 13
SuSE Linux Live CD for Firewall 64, 85
suse-ftp-proxy . . . 78
SuSEconfig . . . 26, 81, 133
SuSEfirewall . . . 131
syslog
  configuration . . . 50
syslog-ng . . . 18, 61

T
telnet . . . 148
tinyproxy . . . 64, 78
traceroute . . . 67, 68
training . . . 99
transproxy . . . 64, 78
troubleshooting . . . 89, 93

U
update . . . 17, 92
  completion . . . 19
  with YaST1 . . . 18
  with YaST2 . . . 18

V
Variable
  `DHCPD_INTERFACE` . . . 117
  `START_DHCPD` . . . 118

X
XDM . . . 151
XFree86 . . . 2
xlock . . . 151
xterm . . . 61

Y
YaST . 2, 17, 25, 26, 31, 32, 148, 165
YaST1 . . . 111, 157
YaST2 . . . 17, 19–27, 31, 157
  hard disk . . . 22
  keyboard . . . 21
  language selection . . . 20
  mouse . . . 21
  network . . . 25
  start installation . . . 24
  time zone . . . 21