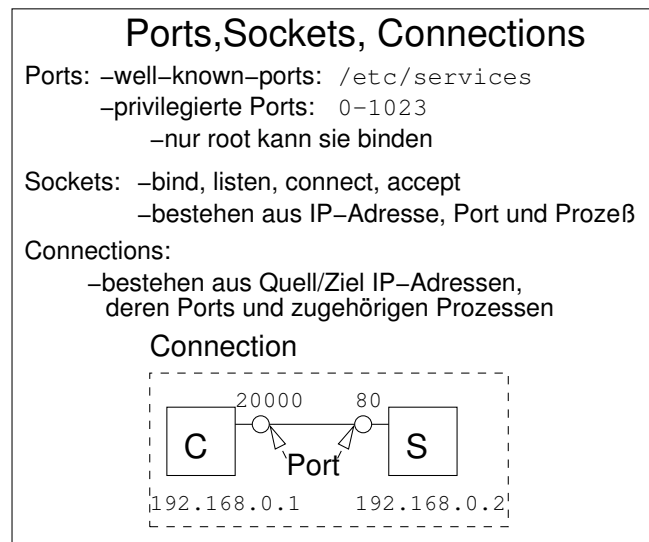

7 TCP/IP-Dienste konfigurieren

In diesem Kapitel lernen Sie

- ④ ▶ die Begriffe Ports, Sockets und Connections kennen (LPI 1: 109.1).
- ④ ▶ den Zusammenhang der Ports von TCP/IP-Diensten mit der Datei `/etc/services` kennen (LPI 1: 109.1).
- ④ ▶ den Internet-Superserver **xinetd** kennen (LPI 1: 110.2).
- ④ ▶ die Funktionsweise des TCP-Wrappers kennen (LPI 1: 110.1).

7.1 Ports, Sockets und die Datei `/etc/services`



Wie wir schon aus den vorangegangenen Abschnitten wissen, können also sehr viele Dienste gleichzeitig auf einem Server laufen. Wenn nun ein Client sich mit dem Server verbindet, wie findet dann die Unterscheidung anhand des Dienstes statt, wenn man bedenkt, dass der Server im Normalfall nur eine einzige IP-Adresse besitzt?

Aus diesem Grunde haben die Protokolle TCP und UDP ein Feld im Protokoll-Header eingebaut, das die so genannte *Portnummer* enthält. Durch diese kann man die Dienste, die auf einem Rechner laufen, unterscheiden. Es gibt insgesamt 65535 Ports (= 16 Bit),


TCP/IP-Dienste konfigurieren

von denen die Ports 1–1023 nur von Prozessen gebunden (= belegt) werden können, deren Eigentümer der Benutzer `root` ist („*privileged ports*“).

Ein *Socket* ist durch IP-Adresse und Port definiert. Ein *Socket* gehört zu einem Prozess, wobei ein Prozess durchaus mehrere *Sockets* öffnen kann.

Die Verbindung von Client-IP-Adresse und Client-Port sowie Server-IP-Adresse und Server-Port nennt man eine *Connection*. Sobald zwischen zwei Rechnern eine *Connection* besteht, können Server und Client sehr einfach kommunizieren. Eine *Connection* wird in drei Schritten aufgebaut:

1. Ein Dienst auf den Server *bindet* (siehe **man 2 bind**) sich an den entsprechenden Port und *lauscht* (siehe **man 2 listen**), ob Verbindungsanfragen eingehen (= offenes *Socket*). Dies geschieht gewöhnlich nur einmal, wenn der betreffende Dienst gestartet wird.
2. Ein Client belegt eine zufällig bestimmte, freie Portnummer, (die zusammen mit seiner IP-Adresse das Absender-*Socket* darstellt), stellt eine *Verbindungsanfrage* an den Port des Dienstes auf dem Server. (siehe **man 2 connect**)
3. Der Dienst *nimmt die Verbindung an* oder lehnt sie ab. (siehe **man 2 accept**)

Für bekannte Internet-Dienste sind standardmäßig bestimmte Ports definiert, wie zum Beispiel Port 80 für HTTP oder Port 25 für SMTP oder Port 23 für `telnet`. Diese Standardports nennt man auch *well known ports*. Diese werden von der Internet Assigned Numbers Authority, der *IANA*, festgelegt und sind in der Datei `/etc/services` (LPI 1: 109.1) nachzuschlagen: 

Ausschnitt aus `/etc/services`

```
tcpmux      1/tcp      # TCP port service multiplexer
echo        7/tcp
echo        7/udp

...

ftp-data    20/tcp
ftp         21/tcp
fsp         21/udp      fspd
ssh         22/tcp      # SSH Remote Login Protocol
ssh         22/udp      # SSH Remote Login Protocol
telnet      23/tcp
# 24 - private
```

7.2 Server mit eigenem Startskript

```
smtp          25/tcp      mail
# 26 - unassigned
www           80/tcp      http # WorldWideWeb HTTP
www           80/udp      # HyperText Transfer Protocol
pop-3        110/tcp     # POP version 3
pop-3        110/udp
...

```

Einige gebräuchliche TCP- und UDP-Ports

20 (TCP): FTP-Data, Datenkanal für FTP-Verbindungen

21 (TCP): FTP, Kommunikationssteuerung bei FTP-Verbindungen

22 (TCP/UDP): SSH, Secure Shell

23 (TCP): Telnet

25 (TCP): SMTP, Simple Mail Transfer Protocol

53 (TCP/UDP): DNS, Domain Name Service

80 (TCP/UDP): WWW/HTTP

110 (TCP/UDP): POP3, Post Office Protocol

119 (TCP): NNTP, Net News Transfer Protocol

139 (TCP/UDP): NetBIOS-SSN, Sitzungsdienst für Windows Netzwerke

143 (TCP/UDP): IMAP, Interim Mail Access Protocol

161 (UDP): SNMP, Simple Network Management Protocol

7.2 Server mit eigenem Startskript

TCP/IP-Dienste konfigurieren

■ *Hinweis:* Im folgenden Text wird für einen Server-Prozess, der im Hintergrund arbeitet, der Begriff *Dämon* verwendet.

Häufig benötigte Server werden bereits beim Booten des Systems gestartet und laufen damit unabhängig von allen anderen Programmen (*standalone*). Damit sind sie sehr schnell abrufbereit, belegen aber andererseits permanent etwas Platz im Hauptspeicher.

Gestartet werden diese Server über Startskripte im Verzeichnis `/etc/init.d/` (LPI 1: 110.2), genaugenommen über die entsprechenden Links in den Unterverzeichnissen der verschiedenen Runlevel in `/etc/rcX.d`, wobei *X* für die Nummer des Runlevels steht. Da bei einigen Startskripten die Reihenfolge wichtig ist, (z.B. baut der NFS-Server auf dem Portmapper auf) wird dem Namen des Links ein Buchstabe (S für Start und K für Kill) sowie eine zweistellige Priorität vorangestellt. Ⓛ

Beispiel für das Startskript und die entsprechenden Links des Mailservers **sendmail**:

```
nagold:~ # ls -l /etc/init.d/sendmail /etc/rc*.d/*sendmail
lrwxrwxrwx 1 ... 11 Feb 23 19:49 /etc/rc3.d/K12sendmail -> ../sendmail
lrwxrwxrwx 1 ... 11 Feb 23 19:49 /etc/rc3.d/S10sendmail -> ../sendmail
lrwxrwxrwx 1 ... 11 Feb 23 19:49 /etc/rc5.d/K12sendmail -> ../sendmail
lrwxrwxrwx 1 ... 11 Feb 23 19:49 /etc/rc5.d/S10sendmail -> ../sendmail
-rwxr--r-- 1 ... 01 Jan 19 11:15 /etc/init.d/sendmail
```

sendmail wird in diesem Beispiel im Runlevel 3 und 5 gestartet und beim Verlassen derselben wieder gestoppt. Mit der Priorität 10 liegt er ziemlich im Mittelfeld und startet, wie alle Server, nach der Initialisierung der Netzwerkkarte und des Routings.

Um einen Server zusätzlich oder nicht mehr zu starten, wird normalerweise der entsprechende Link gesetzt bzw. gelöscht. Wer beispielsweise den WWW-Server Apache nicht benötigt, entfernt die entsprechenden Links wie folgt:

```
# cd /etc/
# rm rc*.d/*apache
```

Soll hingegen der Nameserver im Runlevel 5 gestartet werden, dann werden die Links neu gesetzt.

```
# cd /etc/rc4.d
# ln -s ../init.d/named S10named
# ln -s ../init.d/named K12named
```

Natürlich kann man die entsprechenden symbolischen Links von Hand anlegen, einfacher ist es aber wie folgt (hier für unser Beispiel Apache, der in den Runleveln 2, 3 und 5 automatisch starten soll):

Bei SuSE mit dem folgendem Befehl:

7.2 Server mit eigenem Startskript

```
# chkconfig apache 235
```

Man kann aber auch den Runlevel-Editor von SuSE verwenden. Der findet sich bei YaST unter System→Runlevel-Editor.

Bei Fedora/RedHat lautet der Befehl:

```
# chkconfig --level 235 httpd on
```

Als grafisches Administrations-Tool für diese Aufgabe gibt es bei Fedora/RedHat das Programm **system-config-services**.

Unter Debian gibt es das Utility **update-rc.d**, mit dem die Symlinks in `/etc/rc*.d` gesetzt werden:

```
# update-rc.d apache2 defaults
```

Dabei bezeichnet „apache“ den Namen des Skripts in `/etc/init.d`. „defaults“ bedeutet Start in den Leveln 2, 3 und 5.

Wir werden später, bei dem Thema Sicherheit, auf die Startskripte und Variablen zurückkommen, weil nicht benötigte Server ein unnötiges Sicherheitsrisiko darstellen.