

Fischer/Walther/Schmidt/Werner: *Linux-Netzwerke*



Dr. Stefan Fischer, Dr. Ulrich Walther,
Stefan Schmidt, Christian Werner

Linux-Netzwerke

Aufbau, Administration, Sicherung
2., aktualisierte und erweiterte Auflage



Alle in diesem Buch enthaltenen Programme, Darstellungen und Informationen wurden nach bestem Wissen erstellt und mit Sorgfalt getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund ist das in dem vorliegenden Buch enthaltene Programm-Material mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autoren und Verlag übernehmen infolgedessen keine Verantwortung und werden keine daraus folgende Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieses Programm-Materials, oder Teilen davon, oder durch Rechtsverletzungen Dritter entsteht.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Buch berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann verwendet werden dürften.

Alle Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt und sind möglicherweise eingetragene Warenzeichen. Der Verlag richtet sich im Wesentlichen nach den Schreibweisen der Hersteller. Andere hier genannte Produkte können Warenzeichen des jeweiligen Herstellers sein.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Buches, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form (Druck, Fotokopie, Microfilm oder einem anderen Verfahren), auch nicht für Zwecke der Unterrichtsgestaltung, reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Bibliografische Information Der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.ddb.de> abrufbar.
ISBN 3-935922-17-5

© 2005 Nicolaus Millin Verlag GmbH, Poing (<http://www.millin.de>)

Umschlaggestaltung: Fritz Design GmbH, Erlangen

Gesamtlektorat: Nicolaus Millin

Fachlektorat: Peter Albrecht, Harald Bertram, Dieter Bloms, Anke Börnig, Sumit Bose, Franz Hassels, Fabian Herschel, Marc Heuse, Andreas Jaeger, Klaus Kämpf, Harald Müller-Ney, Jürgen Scheiderer, Sascha Wessels

Satz: L^AT_EX

Druck: Kösel, Krugzell

Printed in Germany on acid free paper.

Vorwort

Wir danken unseren fleißigen Kollegen Marc Bechler, Jens Brandt, Carsten Buschmann, Jörg Diederich, Verena Kahmann, Andreas Kleinschmidt, Torsten Klie und Dennis Pfisterer fürs Korrekturlesen und die zahlreichen Verbesserungsvorschläge.

Ganz besonderer Dank gilt unseren Kollegen Frank Strauß und Oliver Wellnitz, die uns immer mit zahlreichen fachlichen Ratschlägen und Anregungen zur Seite standen und damit einen wichtigen Beitrag zur Entstehung dieser zweiten Auflage geleistet haben.

Wir danken außerdem Herrn Arndt Buschmann für das zur Verfügung gestellte Text- und Bildmaterial zum Thema E-Mail.

Doch mindestens ebenso wichtig wie die fachliche Unterstützung waren der Rückhalt und das Verständnis, das uns unsere Freunde und Familien in der Zeit des Schreibens entgegengebracht haben – herzlichen Dank!

Stefan Fischer, Ulrich Walther, Stefan Schmidt und Christian Werner
Braunschweig, im Frühjahr 2005

Inhaltsverzeichnis

I Grundlagen	1
1 Einführung	3
1.1 Motivation	3
1.2 Aufbau des Buches	5
2 Geschichte, Entwicklung und Aufbau des Internet	9
2.1 Entstehungsgeschichte	9
2.2 Größenentwicklung	11
2.3 Aufbau des Internet	12
2.3.1 Gemeinsames Adressierungsschema	12
2.3.2 Client-Server-Modell	13
3 Das Internet-Referenzmodell	15
3.1 Schichtenmodelle für die Netzwirkommunikation	15
3.1.1 Architektur	15
3.1.2 Datenfluss im Schichtenmodell	16
3.1.3 Behandlung des Pakets während der Übertragung	17
3.2 Das ISO/OSI-Schichtenmodell	18
3.3 Schichtenmodell des Internet	21
3.3.1 Architektur	21
3.3.2 Netzzugangsschicht	21
3.3.3 Internetschicht	22
3.3.4 Transportschicht	23
3.3.5 Anwendungsschicht	24
4 Ein paar Worte zu Linux	27
4.1 Die Entstehung von Linux	28
4.2 Linux-Distributionen	29

II	Internet-Technik	31
5	Physikalische Netzwerktechnologien	33
5.1	Grundlagen	34
5.1.1	Netzwerkadapter und -interfaces	34
5.1.2	Adressierung in physikalischen Netzen	36
5.1.3	Aufbau eines Layer-2-Frames am Beispiel Ethernet	37
5.2	Ethernet	39
5.2.1	Technologische Grundlagen	39
5.2.2	Installation mit YaST	41
5.3	Wireless LAN	43
5.3.1	Technologische Grundlagen	43
5.3.2	Installation mit YaST	45
5.3.3	Verwendung von <code>iwconfig</code>	49
5.4	Andere Technologien	50
5.5	Netzanbindung über PPP	51
5.5.1	Modem	52
5.5.2	ISDN	56
5.5.3	DSL	59
6	Die TCP/IP-Protokolle	65
6.1	Internet Protocol Grundlagen	65
6.1.1	Aufbau der IP-Pakete (Datagramme)	66
6.1.2	Fragmentierung von Datagrammen	68
6.1.3	Adressierung im IP	69
6.1.4	IP-Subnetze und IP-Subnetzmasken	71
6.1.5	Klassenlose IP-Adressen	73
6.1.6	Zuweisung von IP-Adressen	74
6.1.7	Private IP-Adressen	75
6.1.8	Konfiguration von Netzwerkinterfaces	76
6.2	Routing und Paketauslieferung	78
6.2.1	Routing im Internet	78
6.2.2	Direkte Auslieferung von Paketen und ARP	79
6.2.3	Indirekte Auslieferung von Paketen	79
6.2.4	Routing-Tabellen	80
6.2.5	Host-, Netzwerk- und Default-Routen	81
6.2.6	Der Routing-Algorithmus des IP	82
6.2.7	Konfiguration von Routing-Tabellen in SuSE Linux	83
6.2.8	Weiterführendes	88
6.3	IP Masquerading und Network Address Translation (NAT)	89
6.3.1	Grundidee von Network Address Translation (NAT)	89
6.3.2	Beispiel für NAT	91
6.3.3	Grundidee von IP Masquerading/PAT	92

6.3.4	Beispiel für Masquerading	92
6.3.5	NAT und Masquerading mit SuSE Linux	94
6.4	Internet Control Message Protocol (ICMP)	94
6.5	Ausblick: IP Version 6 (IPv6)	95
6.5.1	Merkmale von IPv6	96
6.5.2	Das IPv6-Paket im Vergleich	98
6.5.3	Syntax von IPv6-Adressen	99
6.5.4	Wichtiges Konzept: Tunneling	100
6.6	Transmission Control Protocol (TCP)	101
6.6.1	Überblick	101
6.6.2	Zuverlässige Übertragung	102
6.6.3	Verbindungsorientierte Übertragung	102
6.6.4	TCP-Paketformat	102
6.6.5	Verbindungsaufbau	104
6.6.6	Portnummern	104
6.7	User Datagram Protocol (UDP)	106
7	Domain Name System (DNS)	109
7.1	Symbolische Namen und IP-Adressen	109
7.1.1	Motivation	109
7.1.2	Erste Ansätze zur Namensverwaltung	110
7.1.3	Aufbau der DNS-Namensbereiche	110
7.2	Aufbau der DNS-Datenbank	111
7.3	Aufbau von DNS-Datensätzen	113
7.4	Namensauflösung unter Linux	118
7.4.1	Basiskonfiguration des Resolvers	118
7.4.2	Arbeitsweise des Resolvers	120
7.4.3	nslookup	122
7.4.4	dig	123
7.5	Konfiguration eines BIND 9 Nameservers	126
7.6	Spezialfall: Dynamic DNS	130
8	DHCP	131
8.1	Arbeitsweise von DHCP	132
8.2	Einrichten des DHCP-Servers	135
8.3	Einrichten eines DHCP-Clients	137
9	Programmierung von Anwendungen im Internet	139
9.1	Client-Server im Internet	139
9.2	Ablauf einer Kommunikationsbeziehung	140
9.3	Lokalisierung des Servers	141
9.4	Vergabe von Ports	141
9.5	Client-Server-Anwendungen selbst programmieren	142

Inhaltsverzeichnis

9.5.1	Idee der Socket-Schnittstelle	142
9.5.2	Funktionen der Socket-Schnittstelle	143
9.5.3	Ablauf einer Client-Anwendung basierend auf der Socket-Schnittstelle	146
9.5.4	Ablauf eines Server-Programms	148
9.5.5	Zusammenspiel der Anwendungen	149
9.5.6	Sockets in Java	149
9.6	Zusammenfassung	153
10	Zugriff auf entfernte Ressourcen	155
10.1	Netzwerkweite Konfigurationsdateien mit NIS	156
10.1.1	Einrichten eines NIS-Servers	157
10.1.2	Einrichten eines NIS-Clients	161
10.1.3	Manuelle Benutzung von NIS-Datenbanken und Fehlerdiagnose	164
10.2	Zugriff auf Dateiarhive: File Transfer Protocol (FTP)	166
10.2.1	Aufgaben, Architektur und Eigenschaften	166
10.2.2	Anonymous FTP	168
10.2.3	FTP-Server	169
10.2.4	FTP-Clients	169
10.2.5	Eine FTP-Sitzung	170
10.2.6	Konfiguration von <code>vsftpd</code>	171
10.3	Einbindung entfernter Dateisysteme	174
10.3.1	Network File System (NFS)	174
10.3.2	Samba	179
10.4	Zugriff auf entfernte Terminals	185
10.4.1	Telnet und SSH	185
10.4.2	VNC	187
10.4.3	RDP	189
10.5	Drucken über das Netz: CUPS	190
10.5.1	Grundlagen	191
10.5.2	Administration über das Web-Interface	191
10.5.3	Testen des CUPS-Servers und Konfiguration von CUPS-Clients	192
11	Informations- und Kommunikationsdienste	195
11.1	E-Mail	196
11.1.1	Das Internet Message Format	197
11.1.2	Das Simple Mail Transfer Protocol (SMTP)	198
11.1.3	Das Post Office Protocol (POP)	202
11.1.4	Das Internet Message Access Protocol (IMAP)	205
11.1.5	Konfiguration des E-Mail-Systems mit YaST	207
11.1.6	E-Mail-Clients	209

11.1.7	Rundschreiben mit Komfort: Mailinglisten	213
11.2	Internet Relay Chat (IRC)	217
11.3	Network News	221
11.3.1	Diskussionsgruppen (<i>Newsgroups</i>) im Internet	221
11.3.2	News-Server	222
11.3.3	Funktionsweise von NNTP	222
11.3.4	News-Reader	223
11.3.5	News-Reader unter Linux	223
11.3.6	Einrichten eines News-Servers unter SuSE Linux	226
11.4	World Wide Web (WWW)	226
11.4.1	Einführung	226
11.4.2	Uniform Resource Locators (URLs)	226
11.4.3	HTML	228
11.4.4	Hypertext Transfer Protocol (HTTP)	228
11.4.5	Web-Browser	230
11.4.6	Apache-Web-Server	230

III Sicherheit

233

12	Gefahren, Risikoabschätzung und Sicherheitskonzepte	235
12.1	Was ist Sicherheit?	236
12.2	Konzeptionelle Probleme des Netzwerkbetriebs	237
12.3	Detaillierte Kommunikations- und Risikoanalyse	238
12.3.1	Kommunikationsbedarf	238
12.3.2	Wichtige Fragen bei der Risikoanalyse	238
12.3.3	Risiken durch Benutzer	240
12.3.4	Risiken durch Würmer, Viren und Trojanische Pferde	241
12.3.5	Sicherheitsrisiken der verwendeten Protokolle und Dienste	242
12.4	Sicherheitskonzepte	248
12.4.1	IT-Grundschutz nach BSI	248
12.4.2	Allgemeine Sicherheitskonzepte	250
12.4.2.1	Keine Sicherheit	250
12.4.2.2	Sicherheit durch Verschleiern	250
12.4.2.3	Sicherheit auf Rechnernebene	250
12.4.2.4	Sicherheit auf Netzebene	251
12.4.3	Sicherheitspolitik	251
12.4.3.1	Vorgehen zur Umsetzung eines Sicherheitskonzepts	251
12.4.3.2	Aufstellen der Sicherheitspolitik	252
12.4.3.3	Umsetzung	253
12.4.3.4	Kontrolle	255

12.5 Zusammenfassung	255
13 Angriffe	257
13.1 Angreifer — Herr der Pakete	257
13.2 Umlenken von Datenströmen	259
13.2.1 ARP Spoofing	259
13.2.2 DNS Cache Poisoning	261
13.3 Denial of Service-Angriffe	265
13.3.1 Smurf und Fraggle Angriffe	265
13.3.2 TCP SYN Flooding	266
13.3.3 Verteilte Denial of Service Angriffe	267
13.3.4 trin00	269
13.3.5 Tribe Flood Network 2000	270
13.4 Ausnutzen von Vertrauensbeziehungen	272
13.5 Würmer und Trojanische Pferde	274
13.5.1 Der Internet Worm	274
13.5.2 Der Loveletter Wurm	277
13.5.3 Trojanische Pferde und andere Hintertüren	278
13.6 Gegenmaßnahmen	279
13.7 Zusammenfassung und Ausblick	281
14 Sichere Kommunikationsprotokolle	283
14.1 IPSec	283
14.1.1 Sicherheitsdienste	284
14.1.2 Protokolle	285
14.1.3 IPSec in Linux	290
14.1.4 Manuelle Konfiguration in Linux	291
14.1.5 Automatische Konfiguration in Linux	296
14.1.6 Probleme beim Einsatz von IPSec	302
14.2 Secure Socket Layer (SSL)	302
14.2.1 Sicherheitsdienste	302
14.2.2 Protokolle	303
14.2.3 Zusammenfassung und Ausblick	306
14.2.4 SSL in Linux	306
14.3 Secure Shell (SSH)	307
14.3.1 Sicherheitsdienste	307
14.3.2 Protokolle	308
14.3.3 SSH Programme und Einsatzmöglichkeiten	309
14.3.4 Konfiguration und Einsatz von SSH in Linux	309
14.4 Zusammenfassung und Bewertung	313
15 Firewalls	315
15.1 Eigenschaften von Firewalls	315

15.1.1	Was ist eine Firewall?	316
15.1.2	Funktionsumfang von Firewalls	316
15.1.3	Typen von Firewall-Komponenten	317
15.2	Paketfilter	317
15.2.1	Architektur	318
15.2.2	Funktionsumfang von Paketfilter	318
15.2.3	Vor- und Nachteile von Paketfiltern	319
15.2.4	Zentrale Komponente eines Paketfilters: Filterregeln	319
15.3	Proxy Server (Application Gateways)	321
15.3.1	Architektur	321
15.3.2	Warum Proxy Server?	322
15.3.3	Funktionsweise eines Proxy Servers	322
15.3.4	Vor- und Nachteile von Proxys	323
15.4	Bastion Hosts	324
15.4.1	Grundlagen	324
15.4.2	Spezielle Bastion Hosts	324
15.4.3	Eigenschaften des Bastion Host	325
15.4.4	Vorgehen beim Einrichten eines Bastion Hosts	325
15.5	Firewall-Konfigurationen	327
15.5.1	Dual-Homed Host-Architektur	327
15.5.2	Screened-Host-Architektur	328
15.5.3	Screened Subnet-Architektur	329
15.5.4	Variationen	330
15.5.5	Interne Firewalls	331
15.6	Auswahl und Betrieb einer Firewall	332
15.7	Paketfilter unter Linux: iptables	333
15.7.1	Architektur	333
15.7.2	Aufrufkonventionen	334
15.8	SuSEFirewall2	343
15.8.1	Manuelle Konfiguration	344
15.8.2	Konfiguration mit YaST	347
15.9	Application Proxy Server unter Linux: Squid	348
15.9.1	Funktionalität	348
15.9.2	Konfiguration	348
15.9.3	Zugriffskontrolle	350
15.9.4	Squid als transparenter Proxy	354
15.10	Zusammenfassung	354
16	Virtual Private Networks (VPN)	355
16.1	Einführung	355
16.2	Motivation für den Einsatz von VPNs	356
16.3	VPN-Architektur	357
16.3.1	Typen von VPNs	357

Inhaltsverzeichnis

16.3.2	Konfigurationen		358
16.4	VPNs unter Linux mit IPSec		359
16.5	VPNs unter Linux mit PPTP		360
16.5.1	Installation und Konfiguration des Servers		360
16.5.2	Installation und Konfiguration des Clients		363
16.5.3	PPTP und Firewalls		366
17	Sicherheitsüberprüfung und Alarmanlagen		367
17.1	Programme zur Prüfung der Netzsicherheit		367
17.1.1	Funktionsweise von Netzwerkscannern		368
17.1.2	Fähigkeiten von Netzwerkscannern		368
17.2	Netzwerkscanner in Linux: Nessus		370
17.2.1	Installation von Nessus		371
17.2.2	Konfiguration		373
17.2.3	Testen von Rechnern		374
17.3	Intrusion Detection Systeme		378
17.3.1	Kategorien von Intrusion Detection Systemen		379
17.3.2	Techniken		379
17.4	Intrusion Detection System in Linux: Snort		380
17.4.1	Installation		380
17.4.2	Snort im Einsatz		381
17.4.3	Snort im NIDS Modus – Konfiguration		382
17.4.4	Regeln		383
17.4.5	Auswertung der Ergebnisse		385
IV	Beispiele		387
18	Einzelbenutzer über ISP ans Netz		389
18.1	Grundkonfiguration und Einrichten der Hardware		390
18.2	Einrichten des DSL-Zugangs		391
18.3	Personal Firewall		393
18.4	Der erste Verbindungsaufbau		393
19	Kleines Home-Network		397
19.1	Netzwerkplanung		398
19.2	Dial-on-Demand		399
19.3	Statische Netzwerkkonfiguration auf dem Router-PC		399
19.4	DHCP		400
19.5	File-Server		403
19.6	Firewall und Masquerading		404
19.7	Dynamisches DNS		406
20	Großes Corporate Network		409

20.1 Netzwerkplanung	410
20.2 Anschluß der Hardware und Einrichtung von Routern und VPNs	412
20.3 Firewall	417
20.4 DNS	419
20.5 Web-Server	422
20.6 Mail-Server	423
Literaturhinweise	427
 V Anhang	 429
A Abkürzungen und Akronyme	431
B Well-known Ports	435

Teil I

Grundlagen

Kapitel 1

Einführung

1.1 Motivation

Computernetze spielen in der Informationsverarbeitung und -übermittlung seit einiger Zeit eine immer größere Rolle. Heute kommt praktisch keine größere Organisation mehr ohne ihr eigenes Netzwerk aus, das die dezentral organisierten Rechner zu einem Verbund zusammenschließt. Hinzu kommt die ständig wachsende Bedeutung des Internets, dessen Faszination und Nutzen sich immer weniger Menschen entziehen können.

Nicht nur Unternehmen, sondern eine Vielzahl von Privatsleuten nutzt heute bereits das Netz der Netze, und oftmals haben nicht nur Freaks sogar ein eigenes kleines Netzwerk zuhause, über das die Familie miteinander oder mit Freunden weltweit per E-Mail oder Videokonferenz kommuniziert. Und immer interessanter werden selbstverständlich auch die Entertainment-Möglichkeiten: Video-Streaming von Fußballspielen im Heim- oder Büronetzwerk, gemeinsame Spiele vieler Teilnehmer über das Netz; der Fantasie sind hier fast keine Grenzen gesetzt.

Zur Organisation solcher Netze wird in immer höherem Maße die Technik der *Intranets* eingesetzt. Bei einem Intranet handelt es sich im wesentlichen um ein Netzwerk, das auf der Basis der Internet-Protokolle TCP/IP arbeitet. Außerdem ist ein Intranet gewöhnlich an das weltweite Internet angebunden, aber auf eine Art und Weise, die den Datenaustausch zwischen Internet und Intranet einschränkt. Man könnte die Definition des Begriffs Intranet etwa auf die folgende Formel reduzieren: Intranet = Internet + Sicherheit.

Der Aufbau eines organisationsweiten Netzes als Intranet hat eine Reihe von Vorteilen:

- Die TCP/IP-Protokollfamilie ist seit ca. 30 Jahren erprobt und sehr stabil. Viele der bekannten Sicherheitslücken sind inzwischen geschlossen, so dass

sich die Technik für sichere Kommunikation, wie sie in Intranets erforderlich ist, eignet.

- ❑ Da Internet und Intranet dieselben Protokolle verwenden, lassen sie sich problemlos miteinander verbinden. Die Kommunikation zwischen Partnern in unterschiedlichen Netzteilen funktioniert problemlos und sehr effizient, wenn sie gestattet ist. Es sind keine Protokollumsetzer (Gateways) erforderlich.
- ❑ Es gibt zahllose Anwendungen, die für das Internet entwickelt wurden und nun im Intranet eingesetzt werden können. Beispiele sind Web-Browser, Java-Anwendungen, Dateiübertrag (FTP), etc.
- ❑ TCP/IP-basierte Netze sind preiswert, da die Protokolle für alle gängigen Architekturen und Betriebssysteme zur Standardausstattung gehören.

Gleichzeitig mit dem Siegeszug des Internets begann der kometenhafte Aufstieg einer weiteren Technologie, nämlich des Betriebssystems Linux. Aus den Ursprüngen eines studentischen Projekts hat sich bis heute eine erstaunlich stabile und leistungsfähige Unix-Variante entwickelt, der nicht nur Akademiker eine strahlende Zukunft voraussagen. Im industriellen Bereich wird Linux zur Zeit als potentieller Hauptkonkurrent der Windows-Familie gesehen.

Linux hat gerade für kleinere Firmen und Organisationen, die sich mit der Frage des Aufbaus einer eigenen dezentralen Informationsinfrastruktur beschäftigen, einen gewissen Charme: einerseits kostet Linux fast nichts, da es unter dem Open-Source-Modell vertrieben wird, andererseits gibt es bei geringen Investitionen von unter 100 Euro schon sehr gut zusammengestellte Distributionen für den professionellen Bereich wie etwa die aktuelle (Frühjahr 2005) SuSE Linux Version 9.3, die nur noch wenige Wünsche offen lassen. Und auch der Support, bei kostenlosen Produkten oft ein Problem, ist typischerweise in Linux sehr gut: einerseits bietet das Internet über seine Linux-Newsgruppen Informationen zu allen Problemen und Fragestellungen, andererseits ist bei vielen Distributionen zumindest ein befristeter Support inklusive.

Als Unix-Variante hat Linux natürlich auch den kompletten TCP/IP-Protokoll-stack implementiert; bei guten Distributionen finden sich außerdem zahlreiche Anwendungen, mit denen eine Internet- bzw. Intranet-basierte Informationsverarbeitung aufgebaut werden kann. Damit ist Linux sicher eine sehr gute Wahl als Betriebssystem für die Rechner eines Intranets.

Dieses Buch möchte den Leser mit der Administration von Linux-Rechnern in einem TCP/IP-basierten Netzwerk vertraut machen. Dabei werden einige der hardwaretechnischen Fragen angeschnitten, insbesondere auch die Installation von Netzwerkkarten in einem PC, aber das Buch konzentriert sich doch im Wesentlichen auf Protokoll- und Softwareaspekte. Prinzipiell soll es dem Leser nach der Durcharbeitung des Buches möglich sein, ein eigenes Intranet mit Internet-

anbindung auf Basis von Linux zu realisieren. Dabei spielen insbesondere Sicherheitsaspekte eine große Rolle, die in der zweiten Hälfte des Buches detailliert besprochen werden. Um das Vorgehen bei den verschiedenen zu bewältigenden Aufgaben so anschaulich wie möglich zu machen, werden alle eingeführten Konzepte an verschiedenen Beispielen am Ende des Buches erläutert. Diese Beispiele versuchen vor allem, auf unterschiedliche Situationen und Netzwerkkonfigurationen einzugehen, so dass für jeden Leser etwas Passendes dabei sein sollte.

1.2 Aufbau des Buches

Das Buch besteht entsprechend der Zielsetzung aus vier Hauptteilen. Im ersten Teil werden die wichtigsten Grundlagen gelegt, die zum Verständnis der beiden anderen Teile notwendig bzw. einfach interessant sind. Nach dieser kurzen Einführung beschäftigt sich Kapitel 2 mit Geschichte, Entwicklung und Grundprinzipien des Internets. Aufgabe von Kapitel 3 ist die Einführung in den grundlegenden Aufbau von Netzwerken auf der Basis so genannter Referenzmodelle, die sich im Netzwerkbereich als Schichtenmodelle darstellen. Der Leser wird mit den einzelnen Schichten des Internet und deren Zusammenspiel vertraut gemacht. Kapitel 4 gibt dann einen kurzen Überblick über Geschichte und Aufbau des Betriebssystems Linux, das ja neben Computernetzen im Mittelpunkt dieses Buches steht.

Der zweite Teil beschäftigt sich allgemein mit der Internet-Technologie, wobei wir uns von unten nach oben durch das Schichtenmodell durcharbeiten. Entsprechend beginnen wir in Kapitel 5 mit den physikalischen Netzwerktechnologien. Dabei stehen die heute gebräuchlichsten Technologien für lokale Netze wie Ethernet und Wireless LAN im Vordergrund, aber auch die Anbindung an ein Weitverkehrsnetz wird besprochen. Kapitel 6 geht noch einmal kurz auf die Schichtenstruktur des Internetmodells ein, bevor es detailliert die TCP/IP-Protokolle der Schichten 3 und 4 erläutert. Neben den beiden tragenden Säulen des TCP- bzw. IP-Protokolls betrachten wir auch das Management-Protokoll ICMP. Eine Diskussion über IP-Adressen, ihre Verwendung im Routing, Nutzung privater IP-Adressen im lokalen Netz und ein Ausblick auf die Zukunft von IP runden das Kapitel ab.

Ein eigenes Kapitel widmet sich dann der Frage, wie IP-Adressen in einem Netzwerk vergeben werden. In vielen Netzen lautet heute die Antwort darauf nicht mehr statische Adressenvergabe, sondern Verwendung von DHCP. Die Funktionsweise dieses Protokolls bzw. die Konfiguration der entsprechenden Software ist Thema von Kapitel 8. Kapitel 9 schließt dann die Diskussion dieser noch sehr anwendungsfernen Technologien ab, indem es im Detail auf die vorherrschende Kommunikationsform zwischen Rechnern im Internet eingeht, nämlich

auf das Client-Server-Modell. Es wird beschrieben, wie diese Kommunikation grundsätzlich funktioniert und wie auf der Basis der im vorangegangenen Kapitel eingeführten Protokolle eigene Dienste und Client-Server-Anwendungen entwickelt werden können. Wir haben damit die Schnittstelle zwischen dem so genannten *Transportsystem* und den eher anwendungsorientierten Protokollen und Diensten erreicht.

Letztere betrachten wir in den nun noch folgenden drei Kapiteln des zweiten Teils. Wir beginnen in Kapitel 10 mit dem Zugriff auf entfernte Ressourcen. Dabei gehen wir besonders auf den Zugriff auf Datei-Server ein, der entweder transparent über Systeme wie Samba oder NFS erfolgen oder explizit mittels FTP durchgeführt werden kann. Andere Ressourcen, die im Netz eine Rolle spielen, sind Drucker; auch hierfür werden wir Lösungen vorstellen.

Schließlich kann man auch die komplette Kontrolle über die Ein- und Ausgabe eines Rechners übernehmen. Die hierzu vorgestellte Lösung sind VNC und RDP. Kapitel 11 behandelt die wichtigsten im Internet schon verfügbaren Standard-Dienste, die ja auch den großen Erfolg dieses Netzes ausmachen. Es sind dies E-Mail, WWW und Network News. Zum Ende des zweiten Teils wird mit DNS der Name Service des Internet und damit ein wichtiger Dienst zur Erhöhung der Benutzerfreundlichkeit eingeführt. Mit seiner Hilfe ist es möglich, symbolische Namen wie `www.milllin.de` statt der etwas kryptischen eigentlichen Internet-Adressen zu benutzen.

Nach diesem zweiten Teil sollte klar sein, wie das Internet prinzipiell funktioniert und insbesondere, wie Rechner für die Benutzung im Internet konfiguriert werden. Daher beschäftigt sich der dritte Teil mit Sicherheitsfragen, die entstehen, wenn ein organisationsweites Netz, das auf der Basis der Internet-Protokolle läuft, an das weltweite Internet angeschlossen wird. Dazu beginnt Kapitel 12 mit einer Analyse möglicher Sicherheitsrisiken, die sich durch die Anbindung des Netzwerks einer Organisation an das Internet ergeben. Konkreter wird Kapitel 13, das detailliert auf typische Angriffe eingeht, denen Rechner im Internet ausgesetzt sein können bzw. es in der Vergangenheit waren. Die meisten dieser Angriffe werden heute durch bessere Implementierungen vermieden, aber trotzdem ist es interessant, die möglichen Sicherheitslücken zu kennen und zu wissen, wie sie ausgenutzt werden können.

Die vier abschließenden Kapitel gehen auf mögliche Lösungen der Sicherheitsproblematik ein. Zunächst betrachten wir die Grundlage vieler Sicherheitslösungen im Kommunikationsbereich, die sicheren Protokolle. Hier hat sich rund um Secure Socket Layer (SSL) eine breite Palette von Lösungen entwickelt, die Thema von Kapitel 14 sind. In Kapitel 15 wird mit Firewalls das zentrale Mittel zur Absicherung eines internen Netzes gegen unbefugte Zugriffe aus dem Internet diskutiert. Als dritte heute sehr gängige Lösung betrachten wir dann in Kapitel 16 die so genannten virtuellen privaten Netze (VPNs). Schließlich muss es noch

Möglichkeiten geben, die Effektivität all dieser Maßnahmen zu beurteilen. Einige der für diesen Zweck verfügbaren Werkzeuge stellen wir in Kapitel 17 vor.

Wie versprochen werden im vierten Teil insgesamt drei Beispielkonfigurationen in je einem Kapitel präsentiert. Kapitel 18 beginnt mit der denkbar einfachsten Variante: ein einzelner Benutzer möchte seinen Rechner ans Internet anschließen. Kapitel 19 geht einen Schritt weiter und stellt eine sinnvolle Konfiguration zum Anschluss eines kleinen Heimnetzwerkes ans Internet vor. Zum Abschluss des Buches geht Kapitel 20 dann auf die Möglichkeiten zur Einrichtung eines großen Firmennetzwerkes ein. Damit sollten alle wesentlichen Szenarien abgedeckt sein, so dass sich jeder Leser die auf die eigene Situation passenden Ratschläge leicht zusammenstellen kann.

Kapitel 2

Geschichte, Entwicklung und Aufbau des Internet

2.1 Entstehungsgeschichte

Gegen Ende der sechziger Jahre, als der „kalte Krieg“ seinen Höhepunkt erreichte, wurde vom US-Verteidigungsministerium (Department of Defence - DoD) eine Netzwerktechnologie gefordert, die in einem hohen Maß gegenüber Ausfällen sicher ist.

Wichtigste Anforderung an das neue Netz war, dass es in der Lage sein sollte, auch im Falle eines Atomkrieges weiter zu operieren.

Eine leitungsvermittelte Datenübermittlung über Telefonleitungen war zu diesem Zweck nicht geeignet, da diese gegenüber Ausfällen zu verletzlich sind. Wird eine solche Leitung zerstört, ist die Kommunikationsverbindung verloren.

Aus diesem Grund beauftragte das US-Verteidigungsministerium die *Advanced Research Projects Agency (ARPA)* mit der Entwicklung einer zuverlässigen Netzwerktechnologie. Die ARPA ist selbst keine Organisation, die Wissenschaftler und Forscher beschäftigte, sondern verteilt Aufträge an Universitäten und Forschungsinstitute. Dementsprechend entstand das Internet im universitären Umfeld.

Die geforderte Zuverlässigkeit des Netzes wurde erreicht, indem das Prinzip der *Paketvermittlung* (Packet Switching) eingesetzt wurde. Bei der Paketvermittlung werden zwei Partner während der Kommunikation nur virtuell miteinander verbunden. Die zu übertragenden Daten werden vom Absender in kleine Stücke zerlegt; diese Stücke werden dann völlig unabhängig voneinander übertragen. Der Empfänger muss die Stücke nach dem Eintreffen wieder zusammensetzen. Der große Vorteil dieser unabhängigen Übertragung der Stücke (der sog. Pakete) ist, dass sie sich ihren eigenen Weg durch das Netz suchen können. Wird beispielsweise eine Verbindung zwischen zwei Paketvermittlungsrechnern zerstört, dann

werden die Pakete einfach über eine andere, noch funktionstüchtige Leitung gesendet. Voraussetzung ist natürlich, dass es zwischen je zwei Rechnern mehrere Verbindungen gibt.

Ende 1969 wurde von der University of California Los Angeles (UCLA), der University of California Santa Barbara (UCSB), dem Stanford Research Institute (SRI) und der University of Utah ein experimentelles Netz, das *ARPANET*, mit vier Knoten in Betrieb genommen. Das ARPA-Netz wuchs mit der Zeit rasant und überspannte bald ein großes Gebiet der USA.

Je größer das ARPANET wurde, desto offensichtlicher wurde, dass die bis dahin verwendeten Kommunikationsprotokolle für ein so großes Netz nur noch bedingt tauglich waren. Ein Grund dafür war auch die Notwendigkeit, das Netz schließlich in kleinere Teilnetze aufzuteilen, um eine effiziente Verwaltung und Kommunikation zu ermöglichen. Dafür waren die existierenden Protokolle jedoch nicht ausgelegt. Aus diesem Grund wurden schließlich in weiteren Forschungsarbeiten die *TCP/IP-Protokolle* bzw. das TCP/IP-Modell entwickelt.

TCP/IP wurde mit der expliziten Zielsetzung entwickelt, mehrere verschiedenartige Netze zur Datenübertragung miteinander zu verbinden. Um die Einbindung der TCP/IP-Protokolle in das ARPANET zu forcieren, wurden die Firma Bolt, Beranek & Newman (BBN) und die University of California at Berkeley zur Integration von TCP/IP in Berkeley Unix beauftragt. Dies bildete dann auch den Grundstein des Erfolges von TCP/IP in der Unix-Welt.

Im Jahr 1983 wurde das ARPANET schließlich aufgeteilt. Der militärische Teil des ARPANET wurde in ein separates Teilnetz, das MILNET, abgetrennt, das durch streng kontrollierte Gateways vom Rest des ARPANET – dem Forschungsteil – separiert wurde.

Nachdem TCP/IP das einzige offizielle Protokoll des ARPANET wurde, nahm die Zahl der angeschlossenen Netze und Hosts rapide zu, denn nun war es eine einigermaßen sichere Investition, TCP/IP-Implementierungen zu erstellen.

Die Sammlung von Netzen, die das ARPANET darstellte, wurde zunehmend als Netzwerk betrachtet. Dieser Netzwerk wird heute allgemein als das *Internet* bezeichnet. Was alle diese Teilnetze gemeinsam haben, ist die Tatsache, dass sie die TCP/IP-Protokolle einsetzen.

Seit den 90er Jahren wandelt sich das Internet immer mehr vom einstigen Forschungsnetz zu einem kommerziellen Netzwerk, das vor allem bei weltweit operierenden Firmen immer größeres Interesse findet.

Neue Begriffe haben dementsprechend ihren Eingang in die Netzwelt gefunden, wie z. B. Electronic Commerce, der durch die Offenheit des Internets globalen Handel zwischen Unternehmen, aber auch Individuen ermöglicht. Internet Service Providers (ISPs) spielen eine zentrale Rolle bei der Anbindung neuer Teilnetze oder auch nur einzelner Rechner an den globalen Netzwerk. Es ist heute

ein Kinderspiel, sein Netz oder seinen Rechner mit dem globalen Internet zu verbinden.

2.2 Größenentwicklung

Das Internet wächst in seiner Größe seit Jahren dramatisch an. Jedes Jahr kommen unzählige neue Rechner und Domänen hinzu. Daran konnte auch das Platzen der so genannten „Dotcom-Blase“ nichts ändern, das zu Beginn dieses Jahrtausends zahllose neue Firmen in den Abgrund riss und die Euphorie rund um das Internet deutlich bremste.

Einerseits ist das gut, denn um so mehr Menschen können die Vorteile der globalen Kommunikation nutzen. Andererseits wirft das ungebremste Wachstum jedoch auch Probleme auf. Beispielsweise werden langsam die Internet-Adressen knapp, also die Adressen für Rechner im Netz¹.

Abbildung 2.1 zeigt die Größenentwicklung des Internet grafisch auf (Stand: April 2004). Jeweils aktuelle Zahlen werden unter <http://www.nw.com/> veröffentlicht.

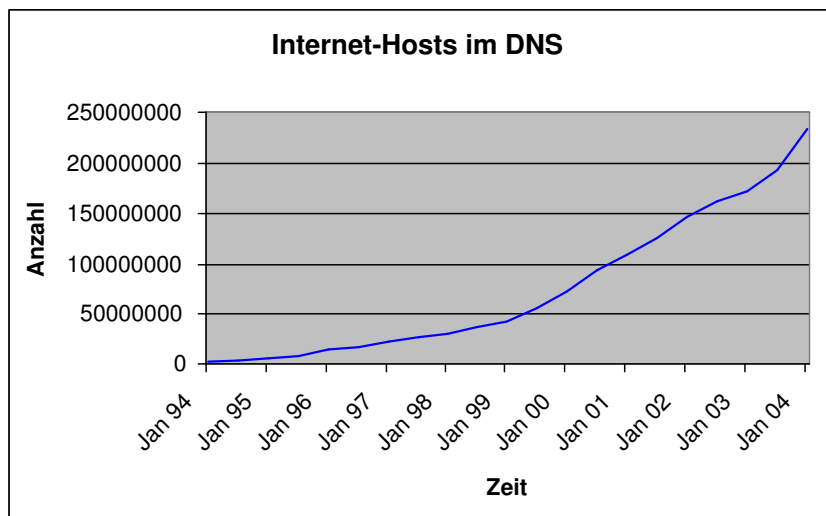


Abbildung 2.1: Entwicklung des Internets

¹Dieses Buch stellt vor allem im Kapitel 6 einige Mittel vor, mit denen versucht wird, diesem Problem abzuwehren.

2.3 Aufbau des Internet

Wie schon gesagt, ist das Internet im Wesentlichen ein Konglomerat von Teilnetzen, die alle zumindest auf den TCP/IP-Protokollen beruhen. Die verbundenen Netze selbst können dabei ganz unterschiedlicher Natur sein; so findet man im Internet sowohl Weit- wie auch Nahverkehrsnetze, Ethernets, Frame Relay, X.25, Token Ring, WLAN etc. Schauen wir uns an, wodurch dieses Netz eigentlich vor allem zusammen gehalten wird.

2.3.1 Gemeinsames Adressierungsschema

Um alle diese Netze verbinden zu können (in anderen Worten: damit jeder Rechner mit jedem anderen Rechner des Internets kommunizieren kann), besitzt das Internet ein einheitliches *Adressierungsschema*. Jeder Rechner im Netz besitzt eine weltweit eindeutige Adresse, unter der er erreichbar ist. Abbildung 2.2 zeigt ein Internet bestehend aus drei Subnetzen, von denen jedes seine eigenen Adressen hat, die jedoch alle eindeutig sind². Der genaue Aufbau solcher Adressen ist Thema von Kapitel 3 bzw. Abschnitt 6.1.3. Grob gesagt hat jedes der Subnetze des Internet eine eigene Netzwerkadresse, und innerhalb dieses Adressraumes besitzt jeder Rechner des Netzes dann wieder seine eigene Unter-Adresse. Zusammen wird die Kombination aus Netz- und Rechneradresse im Internet als *IP-Adresse* bezeichnet, da sie dem IP-Protokoll zur Adressierung dient.

Um also eine Nachricht an einen bestimmten Rechner im Internet schicken zu können, muss dem Sender eine der IP-Adressen dieses Rechners bekannt sein.

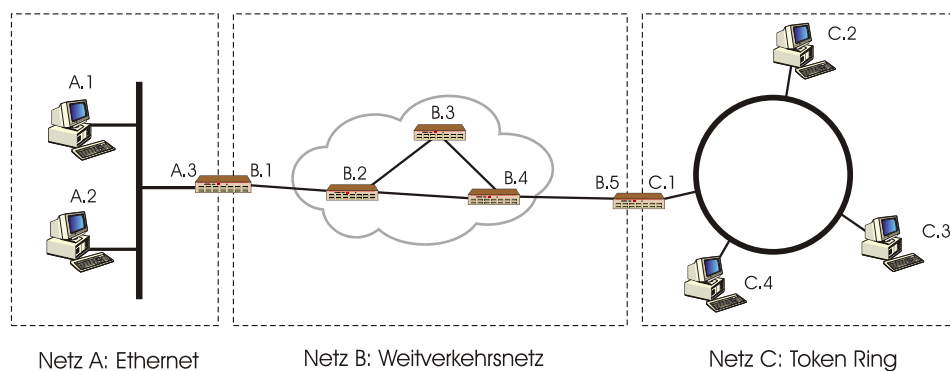


Abbildung 2.2: Internet, aus drei Subnetzen bestehend

²Die Adressen sind hier sehr vereinfacht dargestellt und entsprechen nur schematisch dem Aussehen wirklicher Netzwerkadressen.

Innerhalb des IP-Protokolls werden dann *Routing-Protokolle* eingesetzt, um die Vermittlung eines Pakets vom Quell- zum Zielrechner zu bewerkstelligen.

Computer können zwar sehr gut mit solchen Adressen umgehen, Menschen haben jedoch meist Schwierigkeiten, sich solche eher nichtssagenden und für die maschinelle Verarbeitung optimierten Zahlenketten zu merken. Daher wurde parallel zu der IP-Adressnotation eine ebenfalls hierarchisch aufgebaute Namensnotation festgelegt. Ein Rechnername ist dabei wie die IP-Adresse nach einer Punktnotation aufgebaut, allerdings stehen zwischen den Punkten keine Zahlen, sondern Namen. Generell sieht ein solcher Rechnername etwa folgendermaßen aus:

```
lokalerName.Unter-Organisation.Organsiation.Land
```

Ein typisches Beispiel hierfür wäre der Rechner `www.suse.de`. An diesem Beispiel ist zu sehen, dass nicht immer alle Komponenten verwendet werden müssen. Symbolische Namen aus drei oder vier Komponenten sind typisch.

Die Zuordnung eines solchen Rechnernamens zu seiner tatsächlichen IP-Adresse geschieht im Internet durch lokale Netzdatenbanken und den übergeordneten *Domain Name System* (DNS, s. Kapitel 7).

Heute hat sich die Erkenntnis weitgehend durchgesetzt, dass der Einsatz von TCP/IP-Netzen eine Reihe von Vorteilen bringt, z. B. die Robustheit und die Interoperabilität mit anderen Netzwerken, die auf der gleichen Technologie basieren (also dem Internet oder den Netzen von Partnerunternehmen). Andererseits möchte sich nicht jede Organisation völlig ungeschützt dem Internet anschließen. Als Konsequenz daraus entstanden die sogenannten *Intranets*, also organisationsweite Netze, die auf der TCP/IP-Technologie basieren und zumeist auf eine mehr oder weniger restriktive Weise mit dem Internet verbunden sind. Gefahren im Internet und der Schutz davor sind Thema der Kapitel 12 bis 17.

2.3.2 Client-Server-Modell

Wie funktioniert nun grundsätzlich die Kommunikation zwischen zwei Partnern im Internet, d. h., wie kommen die beiden in Kontakt und wie tauschen sie Daten aus?

Das *Client-Server-Modell* ist das Kommunikationsmodell des Internets. Dieses Modell geht von zwei Kommunikationspartnern aus, von denen einer etwas anzubieten hat, was der andere nutzen möchte. Man spricht auch von einem *Dienst*, der von einem *Server* angeboten und von einem *Client* genutzt wird.

Kapitel 5

Physikalische Netzwerktechnologien

Nachdem wir in den vorangegangenen Kapiteln die konzeptionellen Grundlagen von Netzwerken kennengelernt haben, besonders das in Kapitel 3 behandelte Internet-Referenzmodell, werden wir im Laufe dieses Kapitels die konzeptionelle Ebene verlassen und uns an erste praktische Netzwerkexperimente mit SuSE Linuxwagen.

Der erste Schritt beim Aufbau eines Netzwerks besteht in der physikalischen Vernetzung von Rechnern. Die im Netzwerk übermittelten Nachrichten müssen schließlich über irgendein Medium zu ihrem Bestimmungsort übertragen werden. Wir betrachten in diesem Kapitel also vor allem die OSI-Bitübertragungsschicht, parallel dazu aber auch die OSI-Sicherungsschicht, weil diese in aller Regel sehr eng mit der Bitübertragungsschicht zusammenarbeiten muss und auf deren spezifische Anforderungen zugeschnitten ist (vgl. auch Abbildung 3.5 auf Seite 21).

Nun gibt es eine ganze Reihe von Möglichkeiten, Rechner physikalisch miteinander zu vernetzen – beispielsweise über Kupferkabel, optische Leiter oder Funkwellen. Jede dieser Kategorien unterteilt sich wiederum in viele weitere Varianten, die von nationalen oder internationalen Gremien jeweils in Form von Standards festgeschrieben sind. Besonders bekannte Beispiele für solche Standards kommen vom „LAN/MAN Standards Committee (Project 802)“ der IEEE: Ethernet (802.3), Token Ring (802.5) und Wireless LAN (802.11).

Mitunter haben wir es in physikalischen Netzwerken also mit einer sehr heterogenen Netzinfrastruktur zu tun. Im Abschnitt 5.1 geht es noch einmal um grundlegende Strategien, wie mit einer solchen Heterogenität in physikalischen Netzwerken umgehen können.

In Abschnitt 5.2 werden wir zunächst auf die altbewährte Vernetzung über Ethernet eingehen, im folgenden Abschnitt 5.3 dann aber auch Alternativen wie WLAN

nach 802.11 und andere kabellose Vernetzungsstrategien ansprechen. Schließlich wird in Abschnitt 5.5 auch das Point-to-Point-Protocol (PPP) behandelt, das immer dann zum Einsatz kommt, wenn ein Rechner nicht über ein Netzkabel oder ein Funkinterface in ein LAN eingebunden ist, sondern über eine Punkt-zu-Punkt-Verbindung „logisch“ in ein Netzwerk integriert wird. Wir werden die Funktionsweise von PPP am Beispiel der Einwahl zum Internetprovider über Modem, ISDN und DSL besprechen.

Die einzelnen Abschnitte geben jeweils einen Überblick über die Funktionsweisen der einzelnen Technologien, behandeln ihre spezifischen Vor- und Nachteile und behandeln parallel auch ihre konkrete Umsetzung mit SuSE Linux. Am Ende dieses Kapitels sollten dann alle Grundlagen vorhanden sein, um ein eigenes Netzwerk aufzubauen.

5.1 Grundlagen

5.1.1 Netzwerkkarte und -interfaces

Zum Anschluss eines Rechners an ein physikalisches Netzwerk benötigt man einen Netzwerkkarte, häufig auch als Netzwerkkarte bezeichnet. Dieses Gerät wird an eine Schnittstelle (meistens ISA-, PCI-, PCMCIA- oder USB-Bus) des Rechners angeschlossen und kann Netzwerknachrichten, die auf dem Netzwerkmedium transportiert werden, senden und empfangen. Wie oben bereits angedeutet, gibt es nun eine Vielzahl von Ansätzen, solche physikalischen Netzwerke aufzubauen, und entsprechend viele unterschiedliche Netzwerkkartentypen gibt es. Bevor man sich für einen Netzwerkkarte entscheidet, sollte man sich darüber informieren, ob dieser Adapter auch von SuSE Linux unterstützt wird. Einen sehr guten Überblick gibt hierzu die SuSE Hardwaredatenbank: <http://hardwaredb.suse.de>.

Um die Heterogenität zwischen den verschiedenen Netzwerktechnologien beherrschbar zu machen, gibt es unter Linux das Konzept der Netzwerkinterfaces. Diese Schnittstelle, die vom Betriebssystem (bzw. vom Gerätetreiber) für jeden Netzwerkkarte bereitgestellt wird, ermöglicht eine einheitliche, abstrakte Sicht auf jeden Netzwerkkarte – ganz unabhängig davon, ob dieser den Rechner mit einem WLAN-, Ethernet- oder Token-Ring-Netzwerk verbindet.

Weiterhin gibt es noch virtuelle Netzwerk-Interfaces, die nicht mit einem „echten“ Netzwerkkarte in Verbindung stehen. Sie bilden eine Schnittstelle zu einem virtuellen Netzwerkgerät, das vom Betriebssystem selbst bereitgestellt wird. Über solche virtuellen Interfaces lassen sich, wie wir später noch sehen werden, sehr elegant Konzepte wie Paket-Tunnel realisieren.

Der folgende Befehl listet alle Netzwerk-Interfaces auf einem Linux-System auf:

```

linux: # /sbin/ifconfig -a
eth0      Protokoll:Ethernet
          Hardware Adresse 00:08:74:E0:3D:8A
          inet Adresse:192.168.1.102  Bcast:192.168.1.255
          inet6 Adresse: fe80::248:54ff:fec0:dca0/64
          UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:430 errors:0 dropped:0 overruns:0 frame:0
          TX packets:97 errors:0 dropped:0 overruns:0 carrier:0
          Kollisionen:0 Sendewarteschlangenlänge:100
          RX bytes:57866 (56.5 Kb)  TX bytes:15923 (15.5 Kb)
          Interrupt:3 Basisadresse:0x300

lo        Protokoll:Lokale Schleife
          inet Adresse:127.0.0.1  Maske:255.0.0.0
          inet6 Adresse: ::1/128  Gültigkeitsbereich:Maschine
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:60 errors:0 dropped:0 overruns:0 frame:0
          TX packets:60 errors:0 dropped:0 overruns:0 carrier:0
          Kollisionen:0 Sendewarteschlangenlänge:0
          RX bytes:3912 (3.8 Kb)  TX bytes:3912 (3.8 Kb)

sit0      Protokoll:IPv6-nach-IPv4
          NOARP  MTU:1480  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          Kollisionen:0 Sendewarteschlangenlänge:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

```

In diesem Beispiel gibt es drei Interfaces: `lo`, `eth0` und `sit0`. Zu jedem Interface werden von `ifconfig -a` noch eine ganze Reihe zusätzlicher Informationen angezeigt, etwa die Anzahl von gesendeten (TX) und empfangenen (RX) Paketen, wir wollen an dieser Stelle jedoch hierauf nicht näher eingehen.

Das Interface `lo` ist die Schnittstelle zum sog. Loopback-Device, das auf nahezu jedem Linux-System vorhanden ist. Hier finden wir schon einen Beispiel für ein virtuelles Gerät. Daten, die an dieses Gerät geschickt werden, laufen nicht wie bei „normalen“ Netzwerkadaptern über ein Netzwerkmedium in den Protokollstack eines Zielrechners, sondern werden vom Betriebssystem direkt wieder an die darüberliegende Netzwerkschicht des eigenen Rechners (häufig auch als `localhost` bezeichnet) zurückgeschickt. Sie können Ihr Loopback-Device mit dem Befehl `ping localhost` testen (vgl. Abschnitt 6.4). Nach diesem Befehl sollten sich unter die Anzahl der gesendeten (TX) und empfangenen (RX) Pakete für `lo` in der Ausgabe von `ifconfig -a` erhöht haben.

Das Gerät `sit0` ist ebenfalls kein physikalisches, sondern ein virtuelles Gerät. Es dient dazu, IPv6-Pakete über eine IPv4-Verbindung zu verschicken.

Das Gerät `eth0` ist dagegen ein Schnittstelle zu einem „echten“ Netzwerk-Adapter. Falls ein solcher Eintrag in der Liste auftaucht, deutet dies auf einen bereits korrekt konfigurierten Ethernet-Adapter hin. Der Name `eth0` bezeichnet immer einen Ethernet-Adapter, wobei die `0` für das erste Gerät steht. Haben Sie mehrere Ethernet-Adapter in Ihrem PC, so würden die Geräte mit `eth0`, `eth1`, `eth2` usw. bezeichnet. Interfaces, deren Bezeichnung mit `wlan` beginnen, gehören zu Wireless-LAN-Adaptern, die Interfaces von Token-Ring-Geräten beginnen mit `tr`.

5.1.2 Adressierung in physikalischen Netzen

Weiterhin interessant ist an dieser Stelle noch die „Hardware Adresse“ von `eth0`, die in der Ausgabe von `ifconfig` auftaucht: in unserem Beispiel `00:08:74:E0:3D:8A`. Jeder IEEE-802- (also nicht nur 802.3-)konforme Netzwerkadapter ist mit einer solchen Identifikationsnummer versehen, der sog. MAC-Adresse. Die MAC-Adresse ist weltweit eindeutig. Auf diese Weise wird sichergestellt, dass jedes einzelne Gerät im LAN eindeutig adressierbar ist. Die ersten sechs Stellen dieser Nummer identifizieren den Hersteller eines Netzwerkadapters und werden von einer zentralen Registrierungsstelle einem jeden Netzwerkadapterhersteller zugewiesen. Eine Aufstellung über diese Zuordnung findet man z.B. hier: <http://standards.ieee.org/regauth/oui/index.shtml>. Anhand einer solchen Liste kann man dann feststellen, dass der zu `eth0` zugehörige Netzwerkadapter von der Firma Dell hergestellt wurde. Jeder Hersteller ist dann dafür verantwortlich, dass die restlichen Stellen dieser ID für seine Karten eindeutig sind.

Anwendungsbeispiel: Bridge

Es ist nun, dank des einheitlichen Adressierungsschemas über MAC-Adressen, auch möglich, mehrere physikalische Vernetzungsstrategien miteinander zu kombinieren – und zwar für die darüberliegenden Netzwerkschichten völlig transparent. Es ist also nicht notwendig, sich im LAN auf eine einzige physikalische Netzwerktechnologie festzulegen. Vielmehr lassen sich, dank der technologieübergreifend eindeutigen MAC-Adressen, mehrere Ansätze miteinander kombinieren. So ist es durchaus üblich, für Notebooks und andere mobile Geräte eine WLAN-Infrastruktur bereitzustellen und Server und festinstallierte Arbeitsplatzrechner beispielsweise über Ethernet zu vernetzen.

Ein Gerät, das Teilnetze auf Schicht zwei miteinander verbindet, bezeichnet man als Bridge. Eine Bridge besteht aus physikalischen Schnittstellen zu zwei (meistens unterschiedlichen) physikalischen Technologien, z.B. einem Ethernet-Anschluss und einem Kommunikationsteil für WLAN. Über eine Schaltungslogik wird der Datenverkehr in beiden Teilnetzen beobachtet. Eine Bridge untersucht

dabei ständig die Absender-MAC-Adressen eines jeden Datenpakets und speichert in einer Tabelle, in welchem Teilnetz sich der Netzwerkadapter mit MAC-Adresse befindet. Taucht nun ein Datenpaket in einem Teilnetz auf, dessen Ziel-MAC-Adresse im jeweils anderen Teilnetz liegt, so leitet die Bridge dieses Datenpaket entsprechend weiter.

5.1.3 Aufbau eines Layer-2-Frames am Beispiel Ethernet

Mit Hilfe der MAC-Adressen ist es nun möglich, Nachrichten in einem physikalischen Netzwerk gezielt von einem Rechner an einen anderen zu schicken. Zwar werden wir im weiteren Verlauf dieses Buches hiermit nicht mehr allzu häufig zu tun haben, weil wir für die Adressierung auf höheren Protokollschichten IP-Adressen verwenden; jedoch wird hinter den Kulissen zu jedem IP-Paket, das in einem physikalischen Netzwerk transportiert wird, eine MAC-Adresse ermittelt. In Abschnitt 6.2.2 werden wir noch genauer darauf eingehen.

Nachdem wir in Abschnitt 3 gesehen haben, wie die eigentlichen Nutzdaten (Payload) auf jeder Netzwerkschicht mit Headern oder Trailern versehen werden, und nun das Konzept der Adressierung über MAC-Adressen kennen, wollen wir uns nun am Beispiel Ethernet anschauen, was denn nun tatsächlich über das Netzwerkmedium transportiert wird.

Zwar können wir die Datenpakete nicht auf der untersten Schicht beobachten (hierzu müsste man z.B. mit einem Oszilloskop die elektrischen Signale auf einem Ethernet-Kabel messen), aber mit einem Werkzeug wie Ethereal lassen sich Pakete auf Schicht zwei (diese bezeichnet man auch als Layer-2-Frames) sichtbar machen.

Sie können auch selbst Experimente mit Ethereal durchführen. Es wird als Paket mit SuSE Linux mitgeliefert, muss aber zumeist separat installiert werden, weil dieses Paket nicht zur Standardinstallation gehört. Ethereal wird mit dem Kommando `ethereal` von der Kommandozeile gestartet (Root-Berechtigung erforderlich).

Abbildung 5.1 zeigt die typische dreigeteilte Benutzeroberfläche von Ethereal: Im oberen Fenster sind alle Layer-2-Frames aufgelistet, die während des vorausgehenden Capture-Vorgangs am Interface `eth0` aufgezeichnet wurden (Capture>Start...). Während dieser Aufzeichnungsphase wurde im Web-Browser die deutsche Web-Seite von SuSE aufgerufen. Der dazugehörige HTTP-Get-Request wurde zunächst in einem TCP-Paket eingepackt und das wiederum in ein IP-Paket. Schließlich wurde dieses IP-Paket noch in einen Layer-2-Frame eingepackt und vom Netzwerkadapter über das Ethernet-Kabel übertragen.

Den Aufbau dieses Layer-2-Frames können wir uns nun mit Ethereal anschauen: Es ist das im oberen Fenster markierte Frame mit der von Ethereal vergebenen fortlaufenden Nummer 31. Alle anderen protokollierten Frames wollen wir hier

Kapitel 11

Informations- und Kommunikationsdienste

Neben dem Zugriff auf entfernte Ressourcen steht bei der Einrichtung von Netzen vor allem die Kommunikation und Information im Vordergrund. Der wichtigste und bekannteste Kommunikationsdienst im Internet ist *E-Mail*, wobei dieser asynchrone Dienst heute durch diverse synchrone Chat-Anwendungen wie etwa den *Internet Relay Chat (IRC)* ergänzt wird. Im Bereich der Informationsdienste gab es Ende der achtziger Jahre eine Reihe konkurrierender Ansätze, zu denen zum Beispiel *gopher* gehörte. Tatsächlich haben sich bis heute allerdings nur zwei Standardanwendungen wirklich durchgesetzt, nämlich *Network News* und *World Wide Web*.

In diesem Kapitel wollen wir je zwei Kommunikations- und Informationsdienste detailliert vorstellen.

Im ersten Abschnitt schauen wir uns die Architektur des E-Mail-Dienstes im Internet an und stellen die notwendigen Protokolle SMTP, POP und IMAP vor. Dazu gehören auch die entsprechenden Software-Pakete, in diesem Fall E-Mail-Server und Mail-Client. Den Abschluss des Abschnitts bildet eine Betrachtung von Werkzeugen zur Verwaltung von Mailinglisten.

Der zweite Abschnitt geht auf den IRC ein. Auch hier stellen wir kurz das Protokoll vor, um anschließend kurz auf entsprechende Software-Pakete einzugehen.

Im dritten Abschnitt steht das News-System im Vordergrund, das auf dem NNTP-Protokoll beruht. Der Abschnitt folgt im Aufbau den beiden vorhergehenden: einer kurzen Einführung in NNTP folgt wiederum eine Betrachtung von News-Servern sowie News-Clients.

Schließlich beschreiben wir im vierten Abschnitt den Aufbau des World Wide Web und stellen das Anwendungsprotokoll HTTP vor. Eine Übersicht über die am häufigsten eingesetzten Client- und Server-Programme schließt das Kapitel ab.

11.1 E-Mail

E-Mail war von Anfang an der wichtigste Dienst im Internet. Noch bis Ende der 80er Jahre waren es hauptsächlich Wissenschaftler, die mittels E-Mail im Internet die neuesten Forschungsergebnisse austauschten.

Heute wird der E-Mail-Datenverkehr zwar eindeutig vom WWW-Verkehr dominiert, was aber vor allem auf den Multimedia-Inhalt des letzteren zurückzuführen ist. Seit das Internet populär geworden ist, hat auch das Aufkommen von E-Mail ganz erheblich zugenommen.

Wie in Abbildung 11.1 dargestellt, besteht ein E-Mail-System aus mehreren Komponenten. Der *Mail User Agent (MUA)*, häufig auch als Mail-Client bezeichnet, ermöglicht dem Benutzer eine einfache Interaktion mit dem E-Mail-System. Dazu bietet der MUA dem Benutzer eine grafische Oberfläche zum Lesen und Erstellen neuer E-Mail. Beim Versenden einer neuen E-Mail schickt der MUA die E-Mail typischerweise nicht direkt an den Empfänger, sondern zunächst an einen *Mail Submission Agent (MSA)*.

Dieser reicht sie dann an einen *Mail Transfer Agent (MTA)* weiter, der sich um die Zustellung der E-Mail kümmert. Dabei muss der MTA die E-Mail nicht immer direkt zustellen. Er kann die E-Mail auch an einen anderen E-Mail-Server, einen sog. Relay-Server, weiterleiten, der sich dann um die Auslieferung kümmert. Eine E-Mail kann also durchaus über mehrere MTAs laufen, bevor sie ihr Ziel erreicht. Auf dem Ziel-System speichert ein *Mail Delivery Agent (MDA)* die E-Mail schließlich im *Mail Storage (MS)*. Der Empfänger kann seine E-Mail dann

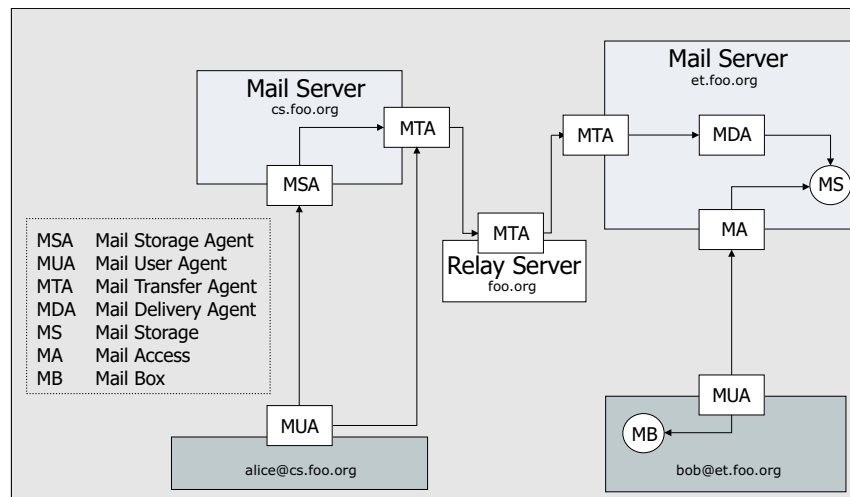


Abbildung 11.1: Komponenten eines E-Mail-Systems

über einen MUA abrufen. Dieser greift über einen *Mail Access (MA)* unter Verwendung eines speziellen Protokolls auf den *Mail Storage (MS)* zu und kann die E-Mails in seine lokale *Mail Box (MB)* herunterladen oder, falls das verwendete Protokoll dies unterstützt, auch direkt auf dem Server bearbeiten.

Informationen, wie eine E-Mail jeweils weitergeleitet werden muss, entnehmen die beteiligten E-Mail-Server zum einen direkt aus der E-Mail und zum anderen aus dem Domain Name System (DNS). Mittels der Empfängeradresse, in der Form `Mailboxname@Domänenname`, wird über das DNS der Eintrag *Mail eXchange (MX)* für den betreffenden Domännennamen ermittelt. Der MX-Eintrag enthält eine oder mehrere Mail-Server bzw. MTAs, die für die Zustellung von E-Mails in diesem Bereich zuständig sind.

Bereits 1971 wurde die erste E-Mail durch Ray Tomlinson im damaligen ARPANET versandt. Darauf aufbauend haben sich die Standards *Internet Message Format* für das Nachrichtenformat einer E-Mail und *Simple Message Transfer Protocol (SMTP)* für den Transfer einer E-Mail entwickelt. Für den Abruf einer E-Mail von einem E-Mail-Server, haben sich das *Post Office Protocol Version 3 (POP3)* und das *Internet Message Access Protocol (IMAP)* etabliert. Bei allen Protokollen, die für die Übertragung von E-Mails zuständig sind, handelt es sich um Protokolle, die auf der Client-/Server-Architektur basieren.

Diese Protokolle werden fortlaufend weiterentwickelt. Die einzelnen Spezifikationen umfassen heute z. T. bereits eine ganze Reihe unterschiedlicher RFCs, die sich mit diversen Aspekten der einzelnen Protokolle befassen. In den folgenden Teilabschnitten werden wir zunächst den Aufbau einer E-Mail nach dem „Internet Message Format“ und dann die Protokolle SMTP, POP und IMAP kurz beschreiben.

11.1.1 Das Internet Message Format

Der Aufbau einer E-Mail wird in RFC 2822 beschrieben. Bei dem RFC 2822 handelt es sich um eine aktuelle Überarbeitung des lange Zeit gültigen RFC 822 und dessen Erweiterungen. Dabei ist zu bemerken, dass RFC 2822 nur den Aufbau einer E-Mail beschreibt und nicht die Übertragung von E-Mails zwischen Hosts definiert.

Eine E-Mail besteht aus zwei Teilen: einem *Header* und dem *Body*. Beide Teile werden durch ASCII-Text dargestellt. Der Header einer E-Mail enthält verschiedene Parameter in der Form `Typ: Wert`. Eine einzelne Zeile ist durch die Zeichen `carriage return` (ASCII-Code 13) gefolgt vom einem `linefeed` (ASCII-Code 10), kurz `<CRLF>`, abgegrenzt und darf maximal, einschließlich dieser Zeichenkette, 1000 Zeichen lang sein. Die Header-Zeile `To:` gibt beispielsweise den Empfänger einer E-Mail an, `From:` den Absender, `Subject:` den Betreff und `Date:` das Sendedatum der E-Mail. Das Header-Feld `Received:` wird von dem

Tabelle 11.1: Typische Felder des E-Mail-Header

Feldname	Beschreibung
From:	Adresse des Verfassers der E-Mail
Sender:	Adresse des Absenders der E-Mail
To:	primäre Empfängeradresse
Cc:	sekundäre Empfängeradresse (Carbon Copy)
Bcc:	Empfängeradresse, die nicht angezeigt wird (Blind Carbon Copy)
Date:	Sendedatum der E-Mail
Reply-To:	Antwortadresse
Message-ID:	Eindeutige ID der E-Mail
Received:	Kennzeichnet den Empfang einer E-Mail durch einen MTA
Subject:	Betreffzeile der E-Mail

zustellenden E-Mail-System ausgefüllt und beinhaltet die Adresse von jedem E-Mail-Server, den die E-Mail durchlaufen hat. Tabelle 11.1 zeigt die typischen Felder eines E-Mail-Headers. Parameter, deren Name mit `X` beginnt, sind nicht reserviert und können für den lokalen Gebrauch verwendet werden.

Vom Header durch eine Leerzeile getrennt, enthält der Body die eigentliche Nachricht. Um die Einschränkungen durch den 7-bit ASCII-Zeichensatz zu umgehen, wurde 1993 die Erweiterung *Multipurpose Internet Mail Extensions* (MIME) definiert. MIME ermöglicht es, unterschiedliche Datentypen in E-Mails zu übertragen, beispielsweise Audio, Video und Grafiken. Dazu benutzt MIME eine Reihe weiterer Header-Felder, die den Typ der MIME-Nachricht klassifizieren. Im Einzelnen sind dies `Mime-Version`: die benutzte MIME-Version, `Content-Description`: eine Beschreibung des Inhalts der Nachricht, `Content-Type`: der Inhaltstyp der Nachricht und `Content-Transfer-Encoding`: die Kodierung der im Body enthaltenen Daten. RFC 2046 definiert eine Reihe von Inhaltstypen und Subtypen für den Parameter `Content-Type`. Durch die unterschiedlichen Inhaltstypen/Subtypen wird es einem MUA ermöglicht bzw. in Kenntnis gesetzt, wie die Nachricht korrekt darzustellen ist. Die Typen `image/gif` und `image/jpeg` definieren beispielsweise zwei verschiedene Typen für Grafiken, `text/xml` ist der Typ für eine XML-kodierte Nachricht.

11.1.2 Das Simple Mail Transfer Protocol (SMTP)

Das Protokoll *Simple Mail Transfer Protocol* (SMTP) aus RFC 2821 dient der Übertragung von E-Mails von einem Host zu einem anderen. Das Kommunikationsmodell von SMTP beruht auf der Client/Server-Architektur. Als Transportprotokoll wird überwiegend TCP verwendet; SMTP ist aber an kein spezielles Protokoll gebunden. Denkbar wären auch andere zuverlässige, bidirektionale Kom-

Kapitel 12

Gefahren, Risikoabschätzung und Sicherheitskonzepte

Das Thema Datensicherheit spielt in der öffentlichen Diskussion eine immer größere Rolle. Mit dem Anschluß eines Computers oder auch eines ganzen lokalen Netzes an ein öffentliches Netz muss man sich sofort die Frage stellen, welchen Gefahren die eigenen Ressourcen (Daten, Software, Hardware) ausgesetzt sind, und wie man diesen Gefahren entgegentreten kann.

Der dritte Teil dieses Buches, der mit diesem Kapitel beginnt, beschäftigt sich intensiv mit den Sicherheitsrisiken in einem Netz, deren Ausnutzung durch potentielle Angreifer sowie schließlich dem Aufbau von Verteidigungsmaßnahmen, einerseits durch die Einführung eines durchgängigen Sicherheitskonzepts, andererseits durch die Durchführung gezielter Maßnahmen wie beispielsweise dem Einsatz von Firewall-Rechnern.

Dieser dritte Teil geht davon aus, dass ein lokales Netz erfolgreich eingerichtet und mit dem Internet verbunden wurde. Nun sollen die Probleme dieser Anbindung bzw. der Vernetzung überhaupt aufgezeigt und Lösungen angeboten werden.

Dieses Kapitel diskutiert zunächst allgemein Gefahren und Sicherheitsrisiken in einem Netz. Nach einer kurzen Einführung des Sicherheitsbegriffs werden allgemeine konzeptionelle Probleme des Netzbetriebs angesprochen, bevor eine detaillierte Kommunikations- und Risikoanalyse durchgeführt wird. Dieser letzte Punkt beschäftigt sich insbesondere mit verschiedenen Schwachpunkten bezüglich der Sicherheit eines Informationssystems, wie etwa den Benutzern oder verwendeten Diensten und Protokollen.

Anschließend stellen wir allgemeine Richtlinien zur Lösung der Sicherheitsproblematik vor, die man allgemein als Sicherheitskonzept bzw. in ihrer Umsetzung dann als Sicherheitspolitik bezeichnet. Die beschriebene Vorgehensweise richtet

sich weitgehend nach den Vorschlägen des Bundesamts für die Sicherheit in der Informationstechnik (BSI, www.bsi.de).

12.1 Was ist Sicherheit?

Bevor man sich mit den Gefahren für die Sicherheit einer Installation beschäftigen kann, muss zunächst einmal klargestellt werden, was eigentlich Sicherheit bedeutet. Die Sicherheit eines Rechners wird allgemein durch die folgenden sechs Punkte definiert:

1. Schutz der Ressourcen vor unberechtigter Benutzung
2. Zugang zu den Informationen für berechtigte Benutzer
3. Verfügbarkeit von Daten
4. Integrität der Daten (die Daten wurden nicht von Unberechtigten verändert)
5. Vertraulichkeit der Daten (die Daten wurden nicht von Unberechtigten gelesen)
6. Schutz der Privatsphäre

Jeder dieser Punkte stellt ein potentielles Sicherheitsrisiko dar und muss durch ein Sicherheitskonzept (s. Kapitel 12.4) erfaßt sein. Potentielle Sicherheitsrisiken sind vielfältig und nicht nur durch den Einsatz von Netzwerken bedingt:

- ☐ direkter physikalischer Zugriff von Unberechtigten auf den Rechner, z. B. durch Einbrecher,
- ☐ Naturkatastrophen wie Hochwasser, Erdbeben, Stürme sowie deren Folgen wie Feuer, Überflutungen etc. – Computer sind dagegen extrem anfällig,
- ☐ fehlerhafte Hardware und Software, deren Einsatz zur Zerstörung von Rechnern oder zur Vernichtung von Daten führen kann,
- ☐ Verlust von Speichermedien wie Disketten, Bändern, CD-ROMs,
- ☐ Ausnutzung der elektromagnetischen Strahlung von Computer-Monitoren, Netzkabeln und Rechnern. Diese Strahlung kann „abgehört“ und „ausgelesen“ werden,
- ☐ ungeschulte Benutzer, die sich der Sicherheitsrisiken nicht bewußt sind,
- ☐ Benutzer, die absichtlich Schaden anrichten,
- ☐ Kommunikation über Netze.

Dieser dritte Teil des Buches wird sich im wesentlichen auf die letzten drei Punkte beschränken bzw. sich besonders auf den letzten Punkt konzentrieren und in den folgenden Abschnitten und Kapiteln auf die Risiken durch den Anschluß an öffentliche Netze eingehen.

12.2 Konzeptionelle Probleme des Netzwerkbetriebs

Die Übertragung von Daten von einem Rechner zu einem anderen über ein Netzwerk ist aus Sicherheitssicht grundsätzlich riskant:

- ❑ Die Daten können abgehört werden (*passiver Angriff*). Die Möglichkeit dazu hängt grundsätzlich von der Beschaffenheit des Kommunikationsmediums ab:
 - Funkübertragung ist sehr einfach abhörbar.
 - Zum Anzapfen metallischer Leitungen wird zumindest ein physikalischer Zugang benötigt.
 - Das Abhören optischer Lichtwellenleiter ist teuer.
 - In lokalen Netzen ist das Abhören des lokalen Verkehrs meist kein Problem (d. h. wenn **ein** Rechner nicht mehr sicher ist, kann unter Umständen das gesamte lokale Netz abgehört werden).
- ❑ *Verkehrsanalyse* erlaubt das Ziehen von Schlußfolgerungen aus bestimmtem auffälligen Sendeverhalten.
- ❑ Veränderung gesendeter Daten oder Erzeugung von Daten an Stelle eines anderen (*aktiver Angriff*). Ein aktiver Angriff kann die folgenden Formen annehmen:
 - Modifizierung, Ergänzung, Löschung oder Neusendung von Dateneinheiten.
 - Veränderung von Routing-Tabellen
Mittels dieser Methode wird der Verkehrsfluß im Netz geändert.
 - Überflutung des Empfängers
Dies wird auch oft als „Denial of Service Angriff“ (DoS) bezeichnet. Rechtmäßige Benutzer können das System nicht mehr nutzen, da ihnen die entsprechenden Ressourcen nicht mehr zur Verfügung stehen.

Oft werden passive und aktive Angriffen kombiniert. Mögliche Angriffe gegen ein Netz oder einen einzelnen Rechner werden im Detail in Kapitel 13 besprochen. Hier seien nur schon einmal einige typische Beispiele genannt:

- ❑ *Password Guessing*
Raten der Passwörter eines Benutzers. Dies ist bei vielen Benutzern recht einfach, da oft „sinnvolle“ Worte verwendet werden, die sich z. B. in einem Wörterbuch finden lassen.
- ❑ *Abhörangriffe*
Passwörter werden oft noch im Klartext übertragen, z. B. bei Telnet oder FTP mit der Sequenz: PASS <password>.

- ❑ *Blinde Authentizitätsangriffe*, z. B. das „Sequence Number Guessing“ oder „IP Spoofing“ (s. Kapitel 13).
- ❑ *DNS Cache Poisoning* (s. Kapitel 13).
- ❑ *Trojanische Pferde, Viren und Würmer*.

12.3 Detaillierte Kommunikations- und Risikoanalyse

12.3.1 Kommunikationsbedarf

Vor einem übereilten Anschluß eines Rechners bzw. eines lokalen an das Internet muss zunächst einmal festgestellt werden, ob der Internetanschluß überhaupt notwendig ist. Ein isolierter Rechner ist bereits wesentlich weniger Risiken ausgesetzt als ein angeschlossener.

Die Beantwortung dieser Frage hängt sehr stark davon ab, welche Dienste des Internets eigentlich genutzt werden sollen. Möchte man etwa nur von Zeit zu Zeit ein bißchen „Web-Surfen“, dann wäre es wahrscheinlich viel sinnvoller, einen einzigen isolierten Rechner mit einer Wählleitung (z. B. T-Online oder AOL) auszustatten und den ganzen Rest des Netzes weiter in Isolierung zu betreiben. Hat die Organisation dagegen vielfältige Kontakte mit anderen Organisationen, die sich von E-Mail über Datei-Transfer bis zu Electronic Commerce erstrecken, dann wird sicher eine Internet-Anbindung des lokalen Netzes eine wünschenswerte Alternative sein. Abbildung 12.1 stellt diese beiden Alternativen grafisch dar.

Die Art des zu realisierenden Zugangs hängt also ganz wesentlich davon ab, welche Dienste des Internets genutzt werden sollen. Dabei ist es wichtig, zwischen Diensten, die von lokalen Benutzern im Internet abgerufen werden, und Diensten, die von lokalen Rechnern für Benutzer im Internet erbracht werden, zu unterscheiden. Diese Anforderungen sollten auf Grund der unterschiedlichen Aufgaben auf jeden Fall sowohl für den zentralen Zugang zum Internet als auch für jeden einzelnen Rechner analysiert werden. In diese Analyse müssen zwei Faktoren eingehen:

- ❑ der Schutzbedarf der einzelnen Daten auf unterschiedlichen Rechnern,
- ❑ die Risiken der unterschiedlichen Dienste.

12.3.2 Wichtige Fragen bei der Risikoanalyse

Das BSI-Grundschutzhandbuch (Bundesamt für Sicherheit in der Kommunikationstechnik, s. auch http://www.bsi.bund.de/gshb/_start.htm, empfiehlt, im Rahmen einer Risikoanalyse zur Feststellung des Schutzbedarfs folgende Fragen zu beantworten:

Kapitel 13

Angriffe

Die Familie der TCP/IP-Protokolle wurde vor fast 30 Jahren in einer Zeit spezifiziert, in der sich kaum jemand Gedanken um Vertraulichkeit oder Authentifizierung machte, da die vorhandenen Netze klein waren und die Nutzer sich in den meisten Fällen vertrauten. Aus diesem Grund wurden die Protokolle lediglich im Hinblick auf den Transport von Daten entwickelt, welches aus heutiger Sicht in diversen Sicherheitslücken in ihrem Design resultierte. Es soll allerdings auch festgehalten werden, dass sich viele der Probleme in diesen Bereichen gar nicht durch die Protokollspezifikation, sondern vielmehr durch eine unsorgfältige Implementierung ergaben. Insbesondere der Bereich der Sicherheit hat einige interessante Beispiele zu bieten.

Dieses Kapitel beschreibt, wie Sicherheitslücken in den TCP/IP-Protokollen zu bestimmten Angriffen genutzt werden können. Welche Typen von Angriffen man unterscheiden kann, wurde bereits in Kapitel 12 beschrieben. In diesem Kapitel werden nun detailliert einige bekannte Angriffsvarianten beschrieben. Hierbei werden die beschriebenen Angriffe nach ihren Zielen in Gruppen zusammengefasst beschrieben. Anzumerken ist jedoch, dass die meisten der heutigen Angriffe Angriffsmöglichkeiten und Schwachstellen auf verschiedenen Ebenen im TCP/IP-Protokollstack kombinieren um daraus den resultierenden Angriff zu bauen.

Bevor wir jedoch auf konkrete Angriffe eingehen, schauen wir zuerst auf einen grundlegenden Baustein, der sozusagen die Grundlage vieler Angriffe liefert: die Möglichkeit gefälschte (engl. spoofed) Pakete zu verschicken.

13.1 Angreifer — Herr der Pakete

Angriffe auf bestimmte Protokolle oder Dienste im Internet erfordern in den meisten Fällen, dass der Angreifer die zu sendenden Datenpakete komplett selbst zusammenstellen kann, um beispielsweise die IP-Adresse des Absenders zu fälschen.

Normalerweise wird, wie in Kapitel 9 beschrieben ein TCP oder UDP Socket geöffnet um darüber Daten zu verschicken. Der Kernel sorgt in diesem Fall dafür, dass das IP-Paket erstellt wird und das TCP bzw. UDP Paket darin eingebettet verschickt wird. Somit können im Protokoll Stack oberhalb von IP maßgeschneiderte Pakete vom Benutzer gebaut und verschickt werden; Angreifern reicht dieses jedoch selten aus.

Es gibt allerdings noch eine weitaus mächtigere Socket-Variante, *Raw Sockets*, die nur mit `root`-Rechten benutzt werden können. Diese erlauben es dem Benutzer das komplette IP-Paket im Speicher zusammenzubauen und dann sozusagen am Kernel vorbei in die Welt zu verschicken. So lassen sich natürlich auf einfachste Weise die IP-Adresse des Absenders, wie auch alle weiteren Header-Felder an die eigenen Wünsche anpassen.

Ein Raw Socket wird, wie auch TCP bzw. UDP Sockets, mit der Funktion `socket()` erstellt:

```
int socket(int family, int type, int protocol);
```

Für die Protokollfamilie (*family*) geben wir im diesem Fall die Internet-Protokollfamilie `PF_INET` an, der Typ (*type*) des Sockets ist `SOCK_RAW` und das Protokoll (*protocol*) `IPPROTO_RAW`.

Darüberhinaus muss dem Kernel noch mitgeteilt werden, dass der IP-Header in den zu versendenden Daten schon vorhanden ist. Dies erledigt die Funktion `setsockopt`, mit der verschiedene Optionen für einen Socket gesetzt werden können:

```
int setsockopt(int sock, int level, int optname, void *optval,
               socklen_t *optlen);
```

❑ `sock`

Der erste Parameter bestimmt den Socket für den die Optionen gesetzt werden sollen.

❑ `level`

Die Optionen für den Socket können verschiedene Ebenen im Protokollstapel beeinflussen. Der Parameter *level* bestimmt hierbei für welche Ebene die Option gilt. Bei unseren Raw Sockets wollen wir das IP Paket selber zusammenbauen und beeinflussen somit die IP-Ebene. Folglich geben wir an dieser Stelle `IPPROTO_IP` an. Optionen für die TCP-Ebene würde beispielsweise mit `IPPROTO_TCP` bestimmt.

❑ `optname`

Der Parameter *optname* gibt an, welche Option gesetzt werden soll. Um dem Kernel mitzuteilen, dass wir uns selbst um die Erstellung des IP-Headers kümmern, übergeben wir die Option `IP_HDRINCL`. Weitere mögliche Optionen auf IP-Ebene sind beispielsweise `IP_OPTIONS` mit der IP-Optionen

gesetzt werden können und `IP_TTL` mit dem sich der *Time to Life*-Wert setzen läßt.

- ❑ `optval`
Die Speicheradresse an der die Optionswerte gespeichert sind wird mit `optval` übergeben.
- ❑ `optlen`
Die Größe des zu schreibenden Optionsblock wird mit `optlen` bestimmt.

In unserem Fall sieht der Aufruf also wie folgt aus:

```
int setsockopt(setsockopt (sock, IPPROTO_IP, IP_HDRINCL, *optval, optlen);
```

Nun muss der Angreifer sein maßgeschneidertes Paket nur noch im Speicher zusammenbauen und kann es über den Socket versenden und den Angriff starten. Verschickt er hierbei IP-Pakete die nicht seine richtige IP-Adresse tragen, so nennt man dies auch *IP-Spoofing*.

Die Möglichkeit Pakete maßgeschneidert zu verschicken, ermöglicht es Angreifern Rechner und Netze auf diverse Art und Weise anzugreifen. Im Folgenden werden wir einige Angriffe etwas näher beleuchten.

13.2 Umlenken von Datenströmen

Das größte Gefahrenpotential im Internet stellen Protokolle dar, die Befehle und Daten im Klartext übertragen. Gelingt es einem Angreifer, dass die Datenpakete an seinem Rechner vorbeikommen, kann er auf einfachste Weise die Daten mitlesen. Das Umlenken von Datenpaketen ist dabei auf mehreren Ebenen im Protokollstack möglich. Zwei Beispiele sollen hier näher erläutert werden: das Umlenken auf der Sicherungs-/Netzwerkschicht mittels *ARP Spoofing* und auf Anwendungsebene mithilfe des *DNS Cache Poisoning*.

13.2.1 ARP Spoofing

ARP Spoofing ermöglicht das Umlenken von Datenverkehr lediglich innerhalb eines Netzsegmentes, da es in die Abbildungsvorgang von IP-Adressen zu MAC-Adressen eingreift. Zum besseren Verständnis des Angriffs schauen wir noch einmal auf die wichtigsten Aspekte der Abbildung von MAC-Adressen zu IP-Adressen.

Kurzer Rückblick: ARP

Die Abbildung erfolgt mittels des *Address Resolution Protocols* (ARP). Befinden sich zwei kommunizierende Rechner im gleichen Netzsegment erfragt der sende-

Kapitel 14

Sichere Kommunikationsprotokolle

Wie das letzte Kapitel angedeutet hat gibt es vielfältige Möglichkeiten die in TCP/IP-Netzen verwendeten Protokolle anzugreifen. Aus diesem Grund wurden sichere Kommunikationsprotokolle entwickelt, die vor vielen dieser Angriffe schützen sollen und dieses Ziel auch erreichen.

Sichere Protokolle können hierbei auf fast jeder Ebene des TCP/IP-Protokollstacks zum Einsatz kommen. In diesem Kapitel werden prominente Vertreter auf Netzwerkebene (*IPSec*), auf Transportebene (*SSL*) und auf Applikationsebene (*SSH*) besprochen. Zu jedem Protokoll wird zusätzlich die Konfiguration und der Einsatz unter Linux besprochen.

Generell verwenden sichere Protokolle kryptographische Maßnahmen und Verfahren um die Kommunikation abzusichern. Ziel dieses Buches ist es jedoch nicht diese kryptographischen Verfahren und ihre mathematischen Grundlagen zu beschreiben (sehr gute Bücher in diesem Bereich finden sich in der Literaturliste), sondern vielmehr die Auswirkungen ihres Einsatz zu betrachten.

14.1 IPSec

IP bietet von sich aus keine Sicherheitsdienste. Das bedeutet beispielsweise, dass die IP-Adresse des Absenders gefälscht werden kann (vgl. Abschnitt 13.1) oder die Daten im Datagramm beim Transfer eventuell verändert werden können. Weiterhin kann jede Station, die das Datagramm durchlaufen hat in dessen Inhalt schauen, sprich die Daten auslesen.

IPSec schafft hier Abhilfe, indem es verschiedene Sicherheitsdienste auf Netzwerkebene bietet. Im Betrieb können verschiedene Sicherheitsprotokolle mit unterschiedlichen Sicherheitseigenschaften gewählt werden, sowie kryptographische Verfahren ausgesucht und Schlüssel ausgetauscht werden. IPSec kann sowohl in IPv4 und IPv6-Netzen genutzt werden. Spezifiziert ist IPSec in mehreren RFCs, wobei die wichtigsten RFC2401, RFC2402, RFC2406 und RFC2408 sind.

14.1.1 Sicherheitsdienste

Die Sicherheitsdienste die IPSec für die einzelnen IP-Datagramme anbietet sind:

- ❑ *Datenintegrität*
Datenintegrität schützt das Datagramm vor nicht legitimer Veränderung. Dies wird durch den Einsatz eines so genannten *HMAC* (Hash-based Message Authentication Code) erreicht, der vereinfacht gesagt einem durch einen geheimen Schlüssel gesicherten Fingerabdruck der Daten entspricht.
- ❑ *Datenauthentifizierung*
Die Datenauthentifizierung stellt sicher, dass das empfangene Datagramm auch wirklich vom angegebenen Absender geschickt wurde. Dies wird ebenfalls durch den eingesetzten HMAC gewährleistet, da der geheime Schlüssel, der in dessen Berechnung mit eingeht nur dem Sender und Empfänger des Datagramms bekannt ist.
- ❑ *Vertraulichkeit*
IPSec bietet zwei Arten von vertraulicher Kommunikation, die Geheimhaltung der Daten im Datagramm und die Geheimhaltung des kompletten Datagramms, je nachdem in welchem Modus IPSec genutzt wird. Die Vertraulichkeit wird durch den Einsatz eines symmetrischen Verschlüsselungsverfahrens gewährleistet.
- ❑ *Verkehrsflussvertraulichkeit*
Wird IPSec im Tunnel-Modus betrieben wird das gesamte Datagramm in ein anderes Datagramm eingepackt und verschlüsselt. In diesem Fall kann ein Angreifer keine Informationen über die stattfindende Kommunikation erlangen, da das gesamte Datagramm inklusive Empfänger und Absender Adresse für ihn nicht zugänglich ist.
- ❑ *Schutz vor Replay-Angriffen*
Fängt ein Angreifer ein legitimes Datagramm (also ein authentifiziertes Datagramm mit intakter Integrität) einer Verbindung ab und spielt dieses zu einem späteren Zeitpunkt wieder in die Verbindung ein, so bezeichnet man dieses als *Replay Angriff*. Hierbei wird versucht, durch bestimmt wiederspielte Datagramme bestimmte Handlungen auszulösen oder die Verbindung zu destabilisieren. Um dies zu verhindern überprüft IPSec bei jedem empfangenen Paket anhand einer Sequenznummer ob es schon einmal empfangen wurde.

Kryptographische Verfahren

Zum Erbringen seiner Sicherheitsdienste nutzt IPSec verschiedene kryptographische Verfahren. Die Auswahl der Verfahren kann sich pro sicherer Verbindung unterscheiden. Einige Verfahren, wie beispielsweise das Verschlüsselungsverfahren

ren DES im *Cipher Block Chaining Mode* (DES/CBC) und die Verfahren zur Integritätssicherung HMAC-MD5-96 und HMAC-SHA-1-96 () sind hierbei als zwingend vorgeschrieben, während diese Verfahren mit andere Schlüssellängen beziehungsweise andere Verschlüsselungsverfahren wie *Triple DES*, *RC5* oder *IDEA* optional ebenfalls genutzt werden können.

Schlüsselmanagement

Da in die Verschlüsselung und die Integritätssicherung geheime Schlüssel eingehen, die nur dem Sender und dem Empfänger bekannt sein dürfen, benötigt IPSec zwingend ein Schlüsselmanagement, das diese zur Verfügung stellt. Zwei verschiedene Arten von Schlüsselmanagement werden von IPSec unterstützt:

- ❑ Manuelle Verteilung und Konfiguration der Schlüssel beim Sender und Empfänger; die so genannten *preshared keys*.
- ❑ Automatische Generierung und Bereitstellung der Schlüssel pro Verbindung durch ein Schlüsselverteilsystem. Standardmäßig benutzt IPSec hier ISAKMP/Oakley. Alternative Schlüsselverteilsystem sind aber ebenfalls denkbar.

Das *Oakley Key Determination Protocol* basiert auf dem *Diffie-Hellman Algorithmus*, jedoch mit erweiterten Sicherheitseigenschaften. Das *Internet Security Association and Key Management Protocol* (ISAKMP) bietet ein Rahmenwerk für den Schlüsselaustausch im Internet. Innerhalb dieses Rahmenwerks können verschiedene Schlüsselaustauschmechanismen benutzt werden, da es keinen speziellen Algorithmus vorschreibt. Gemeinhin wird hier das *Internet Key Exchange* Protokoll (*IKE* – RFC2409) genutzt, das ISAKMP und neben anderen Protokollen auch Oakley implementiert.

14.1.2 Protokolle

Ein Schlüsselkonzept von IPSec sind die so genannten *Security Associations* (SA). Eine SA steht für eine sichere Verbindung in einer Richtung zwischen einem Sender und einem Empfänger und beschreibt die Sicherheitsdienste, die für diese Verbindung gewünscht sind. SAs werden eindeutig durch drei Parameter bestimmt:

- ❑ *Security Parameters Index* (SPI)
Der SPI ist ein String, der der SA zugeordnet ist. Es erlaubt dem empfangenden System eine Zuordnung des IP-Datagramms zu einer SA, für den Fall, dass mehrere SAs zu diesem System existieren.
- ❑ IP-Adresse des Empfängers

❑ *Sicherheitsprotokoll-Bezeichner*

Der Sicherheitsprotokoll-Bezeichner gibt an, welches Protokoll (Authentication Header (AH) oder Encapsulating Security Payload (ESP)) in der SA zum Einsatz kommt.

Zu den SAs kommen noch die *Security Policies* (Sicherheits-Richtlinien) hinzu. Diese Richtlinien geben an, in welcher Situation welche SA zu nutzen ist. Beispielsweise können verschiedene nahezu identische SAs bestehen wobei die passende vom Kernel anhand der Richtlinien ausgewählt wird. Kurz gesagt: Security Policies beschreiben abhängig von der Situation *was* benutzt werden soll und die SAs beschreiben *wie* dies geschieht.

Transport- und Tunnel-Modus

Pro SA kommt bei IPSec eines der beiden Protokolle AH oder ESP im Tunnel- oder Transport-Modus zum Einsatz. Durch diese Kombinationsmöglichkeiten lässt sich die Sicherheit der Verbindung den eigenen Ansprüchen optimal anzupassen.

❑ *Transport-Modus*

Im Transport-Modus schützen die Sicherheitsdienste hauptsächlich den Payload, also die Datenpakete höherer Schichten. Typischerweise wird der Transport-Modus zum Erreichen von Ende-zu-Ende Sicherheit zwischen zwei Endsystemen (Bsp. Client und Server) genutzt.

❑ *Tunnel-Modus*

Im Tunnel-Modus wird das gesamte IP-Datagramm als Payload in ein neues IP-Datagramm gekapselt und durch die Sicherheitsdienste geschützt. Das innere IP-Datagramm enthält die Originaladressen der Kommunikation, während das äußere IP-Datagramm davon abweichende Adressen enthalten kann; beispielsweise die von IPSec-Gateways. Typischerweise wird der Tunnel-Modus zwischen diesen genutzt, um dahinterliegenden Systemen sichere Kommunikation zu ermöglichen ohne dass IPSec bei den Endsystemen implementiert werden muss.

Authentication Header

Das *Authentication Header* (AH) Protokoll bietet Integritätssicherung und Authentifizierung von IP Datagrammen, d. h., das Datagramm kann nicht unbemerkt verändert werden und der Absender ist der, dessen IP-Adresse als Absender angegeben ist. Weiterhin werden Replay Angriffe verhindert.

Im Transport-Modus wird der AH direkt nach dem IP-Header (inkl. Optionen) vor die zu schützenden Daten höherer Protokollschichten in das Datagramm eingefügt. Im Tunnel-Modus wird der AH direkt zwischen dem neuen äußeren IP-Header und den Originalheader des IP-Datagramms eingefügt.

Kapitel 16

Virtual Private Networks (VPN)

16.1 Einführung

Ein *Virtual Private Network* (VPN, *virtuelles privates Netz*) ist im Wesentlichen ein logisches, meist gesichertes und nur für eine beschränkte Teilnehmerzahl zur Verfügung stehendes Kommunikationsnetz auf einem öffentlichen Medium. Das öffentliche Medium kann z.B. das Internet sein.

Einige typische Szenarien, bei denen der Einsatz eines VPNs sinnvoll ist, sind:

- ❑ Zusammenschluss von entfernten Firmen-Intranets, beispielsweise von verschiedenen Filialen, zu einem großen virtuellen Intranet.
- ❑ Anbindung eines mobilen Mitarbeiters an das Firmennetz.
- ❑ Anbindung von Zulieferern, Partnern etc. an ein firmenspezifisches Netzwerk, in dem Geschäftsprozesse abgewickelt werden. Diese Art von Netzwerk wird auch als Extranet bezeichnet.

Virtuelle private Netze sind hierbei durch zwei wesentliche Eigenschaften gekennzeichnet:

- ❑ Sie sind virtuell, d.h. die Zugehörigkeit eines Endsystems zum Netz ist nicht durch seine geografische Lokation innerhalb eines Netzes gegeben.
- ❑ Sie sind privat, d.h. nur ausgewählte Endsysteme gehören zu einem solchen Netz. Alle anderen Rechner können nicht auf die über ein VPN fließenden Daten zugreifen.

Abbildung 16.1 zeigt ein Beispiel für ein VPN. Physikalisch sind hier drei lokale Netze und ein Weitverkehrsnetz vorhanden. Aus jedem der lokalen Netze werden einige Rechner ausgewählt und zu einem VPN zusammengeschlossen.

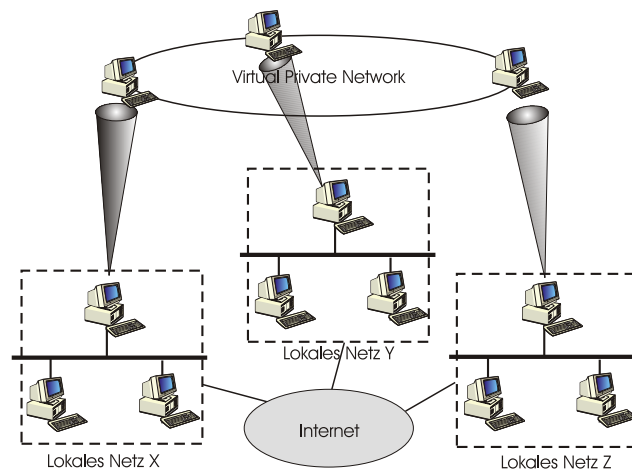


Abbildung 16.1: Beispiel für ein VPN

16.2 Motivation für den Einsatz von VPNs

Welche Vorteile bringt der Einsatz eines VPNs?

❑ Kosten

Der Aufbau und Betrieb eines physikalischen Netzes führt zu einem großen Fixkostenanteil. Hier können sich VPNs einerseits für den Provider und andererseits auch für den Nutzer auszahlen:

- Der Provider kann auf seinem physikalischen Netz durch VPN-Technologie mehreren Kunden virtuelle eigene Netze zur Verfügung stellen, die gegenüber den Netzen anderer Kunden abgeschottet sind.
- Große Firmen müssen keine eigenen physikalischen Netze betreiben, um ihre Standorte sicher zu vernetzen, sondern können auf VPN-Provider oder komplett öffentliche Netze zurückgreifen, wie z.B. das Internet.

❑ Mobilität der Mitarbeiter

Telearbeit wird eine immer wichtigere Arbeitsform. Es stellt sich die Frage, wie mobile Mitarbeiter so an das Firmennetz angebunden werden können, dass die Übertragung sicher ist. VPNs tun hier gute Dienste.

❑ Verbindung zu Kunden, Partnern etc.

Auch mit den Kunden oder Partnern soll eine sichere und vertrauliche Kommunikation möglich sein, die jedoch oftmals über das Internet stattfindet. Eine gute Lösung sind hier VPNs, die auf den jeweiligen Bedarf hin konfiguriert werden.

VPNs sind zwar logische Netze, können aber trotzdem auf jeden Einsatzzweck hin genau so konfiguriert werden, wie das auch mit physikalischen Netzen möglich ist. So können beispielsweise verschiedene Sicherheitsdienste (z.B. Verschlüsselung) genutzt werden, um das VPN nach außen abzuschotten oder auch um den Datenverkehr innerhalb des VPNs durch den Einsatz von Dienstgütemechanismen zu beeinflussen.

16.3 VPN-Architektur

Eine der Basistechnologien für VPNs ist das so genannte *Tunneling*, das den Aufbau eines virtuellen Tunnels erst ermöglicht. In Kapitel 6.5.4 haben wir dieses Konzept bereits verwendet, um IPv6-Pakete über IPv4-Netze zu transportieren. Deshalb wollen wir hier nur noch einmal kurz die wichtigsten Punkte wiederholen.

Prinzipiell bauen zwei Rechner, die Tunnelrechner oder Gateways, eine virtuelle Punkt-zu-Punkt Verbindung zwischen sich auf und bilden somit die Endpunkte des Tunnels. Alle Pakete, die nun durch den Tunnel geschickt werden sollen, werden am Eingangsgateway als Nutzdaten in ein neues Paket eingepackt und dann durch den Tunnel zum anderen Gateway geschickt. Dieser packt das Originalpaket wieder aus und leitet es an sein eigentliches Ziel weiter. Grafisch ist dieses Konzept in Abbildung 6.8 auf Seite 100 visualisiert.

Technisch gibt es für Tunneling verschiedene Umsetzungen. Beispiele sind IP-in-IP Tunneling, wobei ein IP-Paket einfach in ein anderes IP-Paket eingepackt wird, das General Routing Encapsulation Protokoll, wie es bei PPTP verwendet wird (vgl. Abschnitt 16.5) oder die IPSec-Verfahren AH und ESP. Bedenken sollte man bei der Wahl der Tunneling-Methode die jeweiligen Eigenschaften des resultierenden Tunnels. Ein wichtiger Punkt hierbei ist beispielsweise Sicherheit. Ein Tunnelpaket, dessen Nutzdaten beispielsweise nicht verschlüsselt ist, erlaubt jeder Zwischenstation den Zugriff auf diese und somit auf das Originalpaket. Unser Virtual Private Network wird sozusagen nur noch zum Virtual Network, da die Vertraulichkeit der Daten nicht mehr gewährleistet ist.

Welche Sicherheitsdienste letztendlich wirklich genutzt werden und welche kryptographischen Verfahren zum Erreichen dieser Dienste eingesetzt werden, ist vom Betreiber des VPNs frei wählbar. Die einzige Bedingung ist, dass die beiden Tunnelendpunkte die eingesetzten Verfahren unterstützen.

16.3.1 Typen von VPNs

Prinzipiell können VPNs auf unterschiedlichen Ebenen im Schichtenmodell erstellt werden. Folglich unterscheidet man im Wesentlichen folgende Typen von

VPNs, basierend auf dem jeweilig verwendeten physikalischen Adressierungsschema:

- ❑ *Application-Layer VPNs*
Dies ist eine wenig genutzte Variante, da sie mit einem hohen Aufwand einher geht; das VPN muss in allen Anwendungen die genutzt werden sollen bekannt sein.
- ❑ *Transport-Layer VPNs*
Häufig als das „VPN des kleinen Mannes“ wird die Möglichkeit bezeichnet auf der Transportebene ein VPN beispielsweise mittels SSL/TLS (vgl. Kapitel 14.2) einzusetzen. Es wird gemeinhin zur Absicherung von Datenverbindungen zwischen Server und Client genutzt.
- ❑ *Network-Layer VPNs*
VPNs auf Netzwerkschicht bieten den Vorteil, dass alle darüberliegenden Protokolle und Anwendungen transparent durch das VPN geschützt werden und somit keine Kenntnis über das VPN haben müssen. Auf dieser Ebene kommt fast ausschließlich IPSec (vgl. Kapitel 14.1) zum Aufbau von VPNs zum Einsatz.
- ❑ *Link-Layer VPNs*
VPNs auf der Sicherungsschicht haben den grundsätzlichen Vorteil, dass sie multiprotokollfähig sind, d.h. man ist nicht auf IP als Netzwerkprotokoll festgelegt. Weiterhin ist man hier unabhängig von höherliegenden Problemen, wie sie zum Beispiel NAT bei IPSec auf Netzwerkebene bietet. Zwei Techniken dominieren VPNs auf der Sicherungsschicht: PPTP und L2TP.

16.3.2 Konfigurationen

Es gibt verschiedene Arten, ein VPN zu konfigurieren:

- ❑ *End-to-End*
Hier wird die Verbindung direkt zwischen den Endsystemen, also den Rechnern hergestellt. Diese sind sich der Tatsache bewusst, dass ein VPN im Einsatz ist, und müssen folglich alle notwendigen Protokolle implementiert haben und diese verwenden.
- ❑ *Site-to-Site*
Dies ist die klassische Variante. Nur die Router bzw. VPN-Gateways kennen die VPN-Protokolle; für alle Rechner innerhalb eines LANs findet die Kommunikation mit anderen Netzen, die durch das VPN angeschlossen sind, völlig transparent statt. Somit müssen auch nur auf den Gateways für das VPN benötigte Protokolle und Mechanismen implementiert sein.

Kapitel 18

Einzelbenutzer über ISP ans Netz

Als erstes der drei Szenarien betrachten wir eine denkbar einfache Konstellation: den Anschluss eines einzelnen PCs an das Internet. Dies ist sozusagen die Einsteigervariante und wird vielfach im Heimbereich anzutreffen sein. Aber auch für SOHO-Lösungen (Small Office Home Office) bietet sich diese Art des Netzanschlusses oftmals an, zur Realisierung von Heimarbeitsplätzen oder auch zum Zugriff von Außendienstmitarbeitern auf das Firmennetzwerk. Gerade diese Anwendungen erfordern dann schon wieder etwas mehr Aufwand als der einfache Internet-Anschluss, denn hier wird das Internet zumeist nur als Transportmittel genutzt, um auf das Netz der Firma zugreifen zu können. Wir werden diese professionellere Variante erst in Kapitel 20 betrachten, in dem wir das Problem aus der Lage des Unternehmens betrachten. Bleiben wir in diesem Kapitel deshalb zunächst bei der einfachen Internet-Nutzung.

Entsprechend der relativ geringen Komplexität dieser Aufgabe bleibt die Zahl der zu lösenden Probleme auch einigermaßen überschaubar. Wir gehen davon aus, dass der PC über einen Internet Service Provider ans Netz angeschlossen werden soll. Als erstes muss der Rechner entsprechend hardwaremäßig den Vorgaben des gewählten ISPs ausgerüstet werden; dabei stellen praktische alle heute aktiven ISPs zahlreiche Zugriffsvarianten zur Verfügung, von denen die wichtigsten wohl Modem über analoges Telefon, ISDN und DSL sein werden. Weiterhin wird dann von den ISPs (z.B. T-Online) eine spezielle Zugangssoftware bereit gestellt, die für den Benutzer die Verwendung des Netzes komfortabler machen soll. Unter Linux gibt es solche Lösungen jedoch praktisch nicht, und abgesehen davon kann es ohnehin nicht schaden zu wissen, wie denn der Internetzugang eigentlich hergestellt wird. Deswegen werden wir in diesem Kapitel zeigen, wie die Bord-Mittel von Linux eingesetzt werden können, um den Internet-Zugang zu konfigurieren.

Damit der stolze Surfer nicht schon nach kurzer Zeit mit den unangenehmen Seiten der Internetnutzung in Kontakt kommt – im Netz lauern schließlich al-

len möglichen Arten von Viren, Würmern, Trojanern, etc.- ist eine Basissicherung unerlässlich. Wir wollen daher eine Personal Firewall verwenden, die aus den in Teil III vorgestellten Techniken aufgebaut ist. Ist auch diese Hürde genommen und der E-Mail-Zugang aktiviert, steht dem weltweiten Datenverkehr nichts mehr im Wege.

Versuchskaninchen für unseren Aufbau ist Werner Schmidt, der gerade von seinen nichts Böses ahnenden Eltern einen PC zu Weihnachten geschenkt bekommen hat. Wir wollen Werner im Folgenden bei seinen Konfigurationsarbeiten über die Schulter schauen:

18.1 Grundkonfiguration und Einrichten der Hardware

Werner staunte nicht schlecht, als er den riesigen Geschenkkarton auswickelte: Ein brandneuer PC mit allem was dazu gehört – auch ein Ethernet-Adapter ist bereits eingebaut. Da Werners Eltern gehört haben, daß es sich am komfortabelsten mit DSL-Technologie surfen läßt, haben sie mit einem großen Internetprovider einen Vertrag über einen DSL-Internetzugang abgeschlossen. Als kostenlose Zugabe haben die Schmidts gleich ein PPPoE-fähiges DSL-Modem mitgeliefert bekommen. Auch das DSL-Modem findet Werner im Geschenkkarton, zusammen mit den Vertragsunterlagen, auf denen Benutzerkennung und Passwort für den Internetzugang vermerkt sind.

Zusammen mit dem Vertragsabschluß über den DSL-Internetzugang haben die Schmidts auch ihren Telekommunikationsanbieter beauftragt, zusätzlich zum bestehenden ISDN-Anschluß DSL zur Verfügung zu stellen. Der Telekommunikationsanbieter hat daraufhin den Schmidts einen DSL-Splitter geliefert, den Frau Schmidt ihrem Sohn mit in den Geschenkkarton gelegt hat. Als Umstellungstermin auf DSL-Technik hat der Telekommunikationsanbieter den 24.12. mitgeteilt.

Weiterhin findet Werner SuSE Linux 9.2 und das Buch „Linux-Netzwerke“ auf seinem Gabentisch.

Kurz nach 18:00 Uhr möchte Herr Schmidt bei seiner Tante Gerdi anrufen, doch das Telefon ist tot. Werner hat aber schon im Buch „Linux-Netzwerke“ geblättert und weiß die Lösung für dieses Problem. Offensichtlich hat der Telekommunikationsanbieter die DSL-Technik für ihren Anschluß gerade freigeschaltet. Da nun ISDN- und DSL-Signale parallel auf der Telefonleitung übertragen werden (über sog. Frequenzmultiplexing), muß der DSL-Splitter zwischen NTBA und Telefondose geschaltet werden. Dieser trennt ISDN- und DSL-Signale und sorgt dafür, daß ISDN- und DSL-Technik reibungslos funktionieren.

Bei dieser Gelegenheit schließt Werner auch gleich das DSL-Modem an seinen neuen PC an. Die komplette Konfiguration ist in Abbildung 18.1 dargestellt.

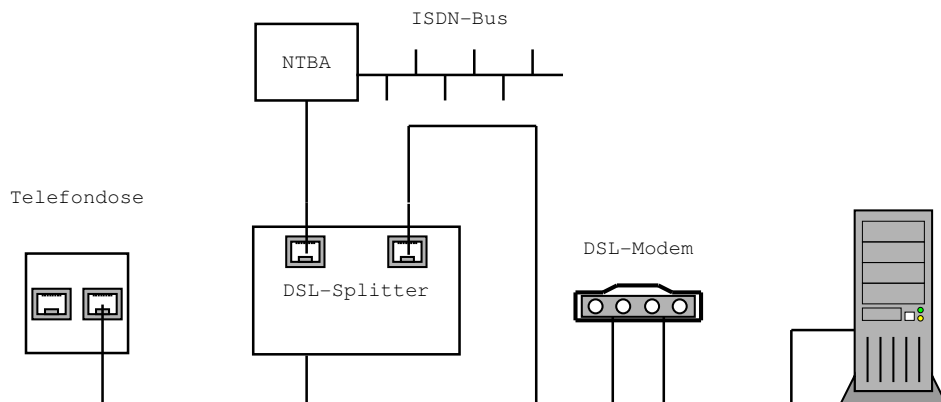


Abbildung 18.1: Anschluß der Netzwerk-Hardware

18.2 Einrichten des DSL-Zugangs

Als nächstes installiert Werner SuSE Linux auf seinem neuen PC. Die Installation läuft weitgehend automatisch ab, und Werner sitzt schließlich an einem PC mit frisch aufgesetztem SuSE Linux.

Natürlich möchte Werner so schnell wie möglich ins Internet, um seinen Freunden noch elektronische Weihnachtsgrüße zukommen zu lassen. Daher macht er sich gleich an die Arbeit und richtet den DSL-Zugang ein.

Zunächst überprüft er, ob der Ethernet-Adapter in seinem Rechner korrekt konfiguriert ist. Schließlich muß ja das DSL-Modem über die Ethernet-Schnittstelle mit dem Linux-PC kommunizieren. Werner meldet sich also als `root` an seinem PC an und gibt das Kommando `ifconfig -a` ein. In der Ausgabe dieses Kommandos findet er einen Eintrag für `eth0`:

```
eth0      Protokoll:Ethernet  Hardware Adresse 00:50:04:D2:71:9E
UP BROADCAST MULTICAST  MTU:1500  Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
Kollisionen:0 Sendewarteschlangenlänge:1000
RX bytes:0 (0.0 b)  TX bytes:342 (342.0 b)
Interrupt:9 Basisadresse:0xec00
```

Das sieht schon einmal gut aus. Fehlte dieser Eintrag, so deutete dies auf ein Problem mit dem Treiber für den Netzwerkadapter hin.

Als nächstes ruft Werner YaST auf, um die nötigen Parameter für eine PPPoE-Verbindung zu seinem Internetprovider zu konfigurieren. Zunächst wählt er

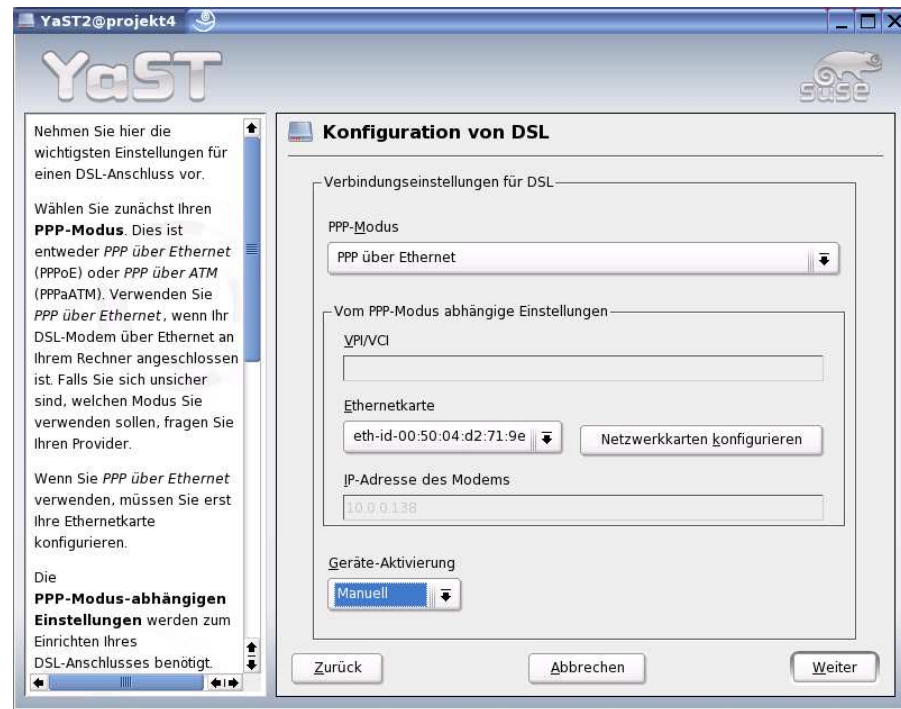


Abbildung 18.2: Einstellungen für den PPPoE-Zugang

„Netzwerkgeräte > DSL“ und dort dann „Konfigurieren > DSL-Geräte hinzufügen“. Hier nimmt er die in Abbildung 18.2 dargestellten Einstellungen vor.

Im nächsten Dialog gibt Werner die Zugangsdaten ein, die sein Internetprovider „Zwei und Drei“ in den Vertragsunterlagen vermerkt hat (Abbildung 18.3).

Anschließend kann Werner noch verschiedene Parameter für den Verbindungsaufbau und -abbau angeben (Abbildung 18.4). Hier kann er zunächst festlegen, ob er „Dial-on-Demand“ machen möchte – also automatisches Wählen, sobald Daten über das PPP-Interface übertragen werden sollen. Zwar ist Dial-on-Demand sehr komfortabel, weil man sich nicht mehr selbst ums Wählen kümmern muß, jedoch möchte Werner bei seinen ersten Verbindungsversuchen die volle Kontrolle über den Wählvorgang behalten und wählt daher diese Option erstmal nicht aus. Die beiden Felder „Während der Verbindung DNS ändern“ und „DNS automatisch abrufen“ kreuzt Werner dagegen an: So werden bei jedem Verbindungsaufbau über das IP Control Protocol (IPCP) aktuelle Informationen zu Adressen von DNS-Servern im Netzwerk des Providers übertragen und in unsere lokale Resolver-Konfiguration übernommen. Schließlich kann Werner noch ein Inaktivitätszeitspanne festsetzen. Werner trägt hier 600 ein, so dass die Ver-

Index

A

Abfangen
 von Paketen 243
 Abgleich UID/GID .. 175
 Abhörangriff 237
 Abhören 236
 accept () 145, 148
 Access Concentrator .. 63
 Access Control List .. 350
 Access Point 399
 Access-Point 44
 Account
 nobody 173
 Active X 245
 Address Resolution
 Protocol 79, 243
 Adressbereich 71
 Adresse
 physikalische 79
 Adressgröße 96
 Adressierung 78
 Adressierungsschema 12
 Adressklasse 70
 Adressraum 12, 22, 70, 75
 Verknappung des ... 95
 Adressvergabe 74
 ADSL 59
 Analysis Console for
 Intrusion Databases .
 385
 Angreifer ... 235, 243, 244
 Angriff 6
 aktiver 237

 passiver 237
 Angriffe 257
 Angriffspunkt 241
 Angriffstool 269
 Verteiltes 269, 270
 Angriffsvarianten ... 257
 Angrifssnetzwerk ... 269
 Anomalien
 erkennen 379
 Anonymität 250
 Anonymous FTP 168
 Anwendung . 17, 104, 106
 datenintensive 103
 verteilte 25, 147
 Anwendungsdaten
 Kodierung von 20
 Anwendungsebene ... 18
 Anwendungsprogramm .
 27
 Anwendungsprotokoll 25
 Anwendungsprozess . 19,
 20
 Anwendungsschicht
 18–21, 24–25
 Anycast 96
 AOL 238
 Apache 230, 306, 423
 Applet 246
 Application Gateways ...
 317, 321
 Application
 Programming
 Interface 142

Application Proxy ... 348
 Applixware 29
 ARP 79, 259
 ARP Cache 260
 ARP Spoofing 41,
 259–261, 279
 ARPA 9
 ARPANET 10
 arpwatch 279
 ASCII 20, 228, 229
 ATM 22, 60, 78
 Außendienstmitarbeiter .
 359
 Audiodaten 97
 Auslieferung
 direkte 79
 indirekte 79
 Authentifizierung 96, 97,
 239, 243, 257
 Authentifizierungsver-
 fahren 239
 Authentizitätsangriff
 blinder 238

B

Büroautomation 21
 Bürosoftware 29
 Bastion Host 317, 324–326
 Interner Bastion Host ..
 325
 Nonrouting
 Dual-Homed Host ..
 324

- Victim Machine ... 325
- Benutzer 28
 - anonymer 169
 - autorisierter 169
 - berechtigte 253
 - böswilliger 240
- Benutzerfreundlichkeit 6
- Benutzerhandbuch ... 30
- Benutzeroberfläche ... 30
- Benutzerschnittstelle
 - grafische 29
- Betriebsmittel 27
- Betriebssystem .. 4, 27, 28
- Big-Endian 64
- Bind 422
- BIND 126–129
 - Master Server 127
 - Slave Server 127
- bind() 145, 148
- Bitübertragungsfehler 19
- Bitübertragungsschicht .. 19
- Bluetooth 50
- BOOTP 131
- Bridge 36
- Broadcast-Adresse 70
- Browser 226, 246
- BSD UNIX 274
- BSI 238
 - Grundschutzhandbuch 238, 248
- BufferedReader 151
- Bug 28
- Bugtraq 378
- Bytelänge 20
- C**
- Cache 245
- CERT 277, 378
- Certificate Authority 300
- Certificate Revocation
 - List 300
- CGI-Skript 228, 245
- Chat-Programm 140
- Chatten 14
- CIDR 73
 - Präfix-Bit 73
- CIFS 180
- Classless Interdomain Routing 73
- Client 13, 14, 106, 140
- Client-Server-Anwendung
 - Programmieren 142–153
- Client-Server-Modell .. 6, 13, 139
- close() 146, 148
- Codesignierung 246
- Common Vulnerabilities and Exposures .. 377
- Computer Professionals for Social Responsibility ... 277
- Computernetz 3
 - Anbindung an das Internet 6
 - kommerzielles 10
 - Komplexität 15
 - organisationsweites .. 6
 - physikalisches 78
- connect() 144
- Cookie 247
- Crossover-Kabel 40
- CUPS 190–193
 - Benutzer- und Gruppenmanagement 192
 - Client 193
 - Server 191
 - Server-Webinterface ... 191
- Cyrus E-Mail-Server . 209
- Cyrus IMAP-Server . 426
- D**
- Darstellungsschicht .. 19, 20
- Datagramm 66, 78, 79
- Datei-Server 29
- Dateitransfer 4, 20, 24
- Daten
 - redundante 19
- Datenübermittlung
 - leitungsvermittelt 9
- Datenübertragung 15, 23
 - in Echtzeit 95
 - sichere 97
 - verbindungslose 23
 - zwischen Prozessen 23
- Datenbank
 - verteilte 110
- Datenbestand 241
- Dateneinheit 102
- Datenformat
 - lokales 20
- Datenintegrität 97
- Datenpaket 16, 22, 66, 140
- Datenschutz 96
- Datensicherheit 235
- Datenströme
 - Umlenken von 259
- Datenstrom 98, 144
- Datenstruktur 20
- ddclient 407
- debian 30
- DEC 274
- Demilitarized Zone . 329, 412
- Denial of Service Angriff 237
- Denial of Service Angriff 243, 265–271, 273
 - Verteilter . 267–271, 280
- DENIC 111
- Department of Defence 9
- Destination IP-Adresse .. 68
- DFÜ 211
- DHCP 131–137
 - Client 132
 - Client Einrichten .. 137
 - Nachrichten 132
 - Server 132
 - Server Einrichten .. 135
- DHCP-Server 44, 400
- Diagnoseinformation . 94
- Dial-on-Demand 392, 399
- Dienst ... 6, 13, 14, 16, 25
 - echo 106
 - Risiko eines 238

- Standard- ... 6, 14, 141, 195–231, 242
- Dienstanfrage 147
- Dienstarten 96, 97
- Dienste 19
- Dienstklasse 98
- Dienstschnittstelle 16
- dig 123
- Digitale Signatur 243, 246
- Distributed
- Denial-of-Service 267
- Distribution 4, 27, 29
- DNS 6, 13, 25, 99, 262, 419
- Authoritative Server ... 115
 - Canonical Name .. 117
 - dynamisch 406
 - Iterative Abfrage .. 112
 - MX-Datensatz 116
 - Rekursive Abfrage 113
 - Resource Records . 112
 - Secondary Server .. 113
 - Start of Authority . 114
 - Zonefile 112
- DNS Cache Poisoning ... 238, 259, 261–263, 280, 370
- DNS-Datenbank 111–118
- DNS-Root 112
- DNS-Server 394
- Domäne 11
- Domain 110
- länderspezifische .. 110
- Domain Name System ... 13, 25, 109–130, 141, 147, 245
- Dreiwege-Handshake ... 104, 266
- Drucken-Tools
- lp 191
 - lpadmin 191
 - lpoptions 191
 - lpq 191
 - lpr 191
 - lprm 191
 - lpstat 191
- DSL 44, 59
- Anschluss 390
 - Einrichten 391
 - Varianten 59
- DSL-Modem 60
- DSL-Splitter 60
- DSLAM 60
- Dual-Homed Host ... 327
- Durchsatz 23
- Dynamic DNS 130
- E**
- E-Mail 3, 6, 141, 196–216, 238, 244
- Aufbau 197
 - Body 198
 - Client 202, 209
 - Filter 210
 - Header 197
- E-Mail-Adresse 210
- E-Mail-Server 197
- EBCDIC 20
- Echtzeitübertragung .. 97
- Echtzeitanwendungen 96
- Effizienz 107
- Einbrecher 236
- Eindringling 244
- Einmalpasswort 243
- Eins-zu-Eins-Zuordnung 90
- Einteilung in Schichten .. 20
- Electronic Commerce 10, 238
- Empfänger .. 79, 101, 103
- Empfängeradresse 68
- Empfängerseite 102
- Empfangsquittung .. 102
- Ende-zu-Ende-Verbindung 66
- Endsystem 17, 23, 65, 101
- Entfernte Dateisysteme .. 174–185
- Entfernte Terminals 185–189
- Entmilitarisierte Zone ... 329
- Entwicklungsumgebung . 29
- Erdbeben 236
- Ergänzung
- von Daten 237
- Ersetzen
- von Paketen 243
- Ethereal 37
- Ethernet 12, 21, 37, 39, 78
- Übertragungsgeschwindigkeit 39
 - Frame 39
 - Varianten 39
- Ethernet-Karte 21
- ETSI 43
- Exterior Gateway
- Protocol 23
- Extranet 355
- F**
- False Positive 378
- Fast Ethernet 39
- Fehlerkorrektur
- vorausschauende ... 19
- Fertigung
- computergestützte .. 21
- File-Descriptor 143
- File-Server 403
- Filterregel . 253, 319, 333, 337, 344
- Finger-Dienst 245
- fingerd 275
- Firewall . 6, 235, 243, 251, 253, 302, 315–354, 404, 417
- Funktionsumfang . 316
 - Komponenten 317
 - Konfigurationen 327–332
- Firewall-Builder 333
- Flow Label 98
- Forschungsnetz 10
- Fraggle Angriff . 266, 280
- Fragmentieren
- von Datagrammen . 66, 68
- Frame 79

- Frame Relay 12, 22
FreeS/WAN 290
FTP 4, 24, 141, 147,
166–174, 237, 244, 307
 Anonymous 168
 Client 166, 169, 170
 Server 166, 169, 244
 Sitzung 170
Fully Qualified Domain
 Name 111
Funkübertragung 237
Funkstrecke 19
- G**
Gateway 4
Gefahrenpotential ... 259
Geschwindigkeit 107
gethostbyname() . 147
getservbyname() 147,
148
Gigabit Ethernet .. 21, 39
GNU Public License .. 29
GNU-Projekt 29
GPL 29
Guarddog 333
- H**
Hardware 28
 automatische
 Erkennung 41
 Installation von 41
Hardwarekomponente 22
Herstellerfirma 28
HIPERLAN/2 43
Hops 68
Host 69, 78
Hostrechner 23
HTML 228
HTML-Seite 246
HTTP 25, 228
HTTP-Server 229
HTTPS 244
Hub 40
Hydra 371
Hypertext 226
Hypertext Transfer
 Protocol 25
- Hypertext Transfer
 Protocol 228
- I**
IANA 74, 105
ICANN 111
ICMP 94
 Header 94
ICMP-Angriff ... 243, 370
Identifikationsschema 70
IDS 271
IEEE
 WG 802.3 39
 LAN/MAN Standards
 Committee 33
 WG 802.11 43
IETF 95, 306, 307
ifconfig 34
IKE 285, 296
IMAP 197, 205–207
 Phasenmodell 206
Implementierung 6
 unsorgfältige 257
incoming 169
Initiator 104
InputStream 151
Insider 253
Installationsprogramm ..
 30
Installationsunterstützung
 30
Integrität 236
Interior Gateway Protocol
 23
Internet ... 3, 78, 226, 235,
246
 Adresse 11
 Entstehungsgeschichte
 9–11
 Größenentwicklung 11
 Protokolle 6
 Schichtenmodell des ..
 21–25
Internet Assigned
 Numbers Authority .
 74, 105
- Internet Control Message
 Protocol 94
Internet Engineering Task
 Force 95
Internet Protocol 65
Internet Protokolle 3
Internet Service Provider
 10, 74
Internet Wurm . 274–277,
281
Internet-Adressen 6
Internet-Dienst 147
Internet-Modell 15
Internet-Notation 147
Internet-Welt 65
Internetadresse
 eindeutige 12
Internetanbindung 4
Internetschicht . 22–23, 78
Interoperabilität 13
Interprozeßkommunika-
 tion 144
Intranet 3, 13, 25, 91
Intrusion Detection
 Systeme 378
 hostbasiert 379
 netzbasiert 379
IP 23
 Datenfeld 67
 Datenteil 66
 Header 66
 Headerlänge 66
 Optionsfeld 67
 Version 66
IP Control Protocol .. 392
IP Masquerading .. 89–94
IP Spoofing 238, 243, 259,
321
IP-Adresse 12, 25, 42,
69–78, 109, 141, 144,
145, 147
 Adressraum 70, 75
 Adressvergabe 74
 doppelte Vergabe ... 74
 dynamische 74
 Eindeutigkeit 74

-
- Fälschen 257
 - Hostanteil 70
 - klassenlose 73–74
 - Netzanteil 70
 - private 75
 - unbenutzte 73
 - Zuweisung 74
 - IP-Adressnotation 13
 - IP-in-IP 101, 357
 - IP-Protokoll 22
 - IP-Spoofing 272, 370
 - IP-Telefongespräch ... 67
 - IP_HDRINCL 258
 - ipchains 333
 - IPnG 96
 - IPSec 283–302, 412
 - AH Aufbau 287
 - Authentication Header 286
 - Encapsulating Security Payload 288
 - ESP Aufbau 288
 - Konfiguration
 - automatische 415
 - Automatische .. 296–301
 - Manuelle 291–296
 - Schlüsselmanagement . 285
 - Security Association ... 285
 - Security Policy 286
 - Sicherheitsdienste . 284
 - Sicherheitskonfigurationen 289
 - SPI 285
 - Transport Modus .. 286
 - Tunnel Modus 286
 - IPSec-Gateway 286
 - IPTables 94, 333–343
 - Architektur 333
 - Aufrufkonventionen .. 334
 - Chains 333
 - Tables 334
 - IPv4 65, 95, 100, 357
 - Paket 101
 - IPv6 ... 35, 65, 95, 96, 100, 357
 - Erweiterungsheader ... 97, 99
 - Header-Format 97
 - Optionen 97
 - Paket 101
 - Protokollversion 98
 - Router 98
 - ISAKMP/Oakley 285
 - isakmpd 291
 - ISDN 22, 56
 - CAPI 58
 - Kanalbündelung ... 59
 - ISDN-Adapter 22
 - ISO/OSI-Referenzmodell 18–20
 - ISP 78, 389
 - IT Grundschutz 248
 - IT-Landschaft 248
 - iwconfig 49
 - J**
 - Java 4
 - Java Virtual Machine 246
 - Jugendschutz 353
 - Juggernaut 244
 - K**
 - Kabel 19
 - Kartenleser 252
 - KDE 29
 - Kernel 28, 29
 - Kernel-Modul 46
 - KInternet 54
 - Klartext 237, 244, 259
 - Klartextübertragung . 243
 - KMail 210
 - Kommunikation
 - gesicherte 15
 - sichere 4
 - vertrauliche 356
 - Kommunikationsdienst .. 20
 - Kommunikationsendpunkt 102, 105, 107
 - Kommunikationskanal .. 17
 - Kommunikationskosten . 29
 - Kommunikationsmedium globales 28
 - Kommunikationsmodell . 13, 142
 - Kommunikationspartner 13, 17, 18, 70, 140
 - Kommunikationsprotokoll 10
 - Kommunikationssteuerungsschicht .. 19, 20
 - Kommunikationssystem . 18
 - Kommunikationsverbindung 9
 - Konfigurationsprogramm 30
 - Kontrolle
 - der Sicherheit 255
 - Kosten-Nutzen-Analyse . 252
 - Kreditkartennummer 244
 - Kryptographische
 - Verfahren .. 283, 284, 303, 308
 - L**
 - Löschung
 - von Daten 237
 - LAN-Analyzer 243
 - LaTeX 29
 - Layer-2-Frame 37, 51
 - Leistung
 - eines Protokolls ... 103
 - Leiter
 - optische 23
 - Leitung 15
 - Linux 4, 22, 27–30
 - Entstehungsgeschichte 28–29
 - Linux-Rechner
 - Administration 4
 - listen() 145, 148
 - Listserv 214

Little-Endian 64
Local Area Network .. 21
Loopback-Device 35
Loveletter Wurm 274,
277–278

M

MAC-Adresse 36
Mafiaboy 269
Mail Transfer Agent . 196
Mail User Agent 196
Mail-Relay-Server ... 208
Mail-Server 423
Majorcool 214
Majordomo 214
Man-in-the-Middle
Angriff 261, 263, 307,
321
Manipulation
eines Datenpakets . 243
Masquerading .. 333, 344,
404
Match 337
Maximum Transfer Unit .
38, 69
Medium 19
physikalisches 16
Metropolitan Area
Network 21
Microsoft 245
Microsoft Windows ... 27
Microsoft-Monokultur ...
277
Middleware 139, 157
Minix 28
Mitarbeiter
mobiler 355
Mitschneiden
einer Kommunikation .
243
Mixer 271
Mobile IP 97, 101
Mobilität
von Rechnern 96
mod_ssl 306
Modem 52
Modifizierung

von Daten 237
Morris Worm *siehe*
Internet Wurm
Morris-Wurm ... 242, 245
MS-DOS 28
Multicast .. 19, 70, 96, 101
Multimedia 106
Multimediaübertragungen
95

N

Nahverkehrsnetz 12
Name
symbolischer ... 25, 70,
109, 141, 147
Name Service 6
Name Service Cache
Daemon 165
Namensauflösung .. 111,
120, 141
Namensbereich 110
Namensdienst 109
Namenshierarchie ... 111
Namenskonvention
des Internet 109
Nameserver 111, 419
NAT 89, 250, 302
NAT-Traversal 302
Nessus 255, 326, 370–378
Client 374
Plugins 373, 376
Server 373
Server-Zertifikat ... 373
Nessus Attack Skripting
Language 370
NetBIOS 182
API 182
Netfilter 333
Architektur 334
Netscape 226, 246
Network Address
Translation ... 89–94,
333
Network News 6,
221–226, 244
Diskussionsforum . 221
Netz

öffentliches 235
logisches 355
lokales 235, 355
physikalisches 356
Netzarchitektur 21
Netzbandbreite 97
Netzdatenbank 13
Netzklasse 70
Klasse-A 70
Klasse-C 70
Netzmaske 74
Netzsicherheit
Prüfung 367
Netzverbund 10
Netzwerkadapter 34
manuelle Konfiguration
45
Netzwerkadministrator ..
25
Netzwerkadresse .. 12, 72
Netzwerkarchitektur . 15
Netzwerkinterface 34
virtuell 34
Netzwerkkarte 4
Netzwerkkommunikation
15
Netzwerkplanung .. 398,
410
Netzwerkrechner 25
Netzwerkscanner 368
Netzwerktechnologien ..
33–64
Netzwerkverteiler 40
Netzzugangsschicht
21–22
News-Provider 222
News-Reader ... 221, 223
News-Server 221, 222
Replikation 222
Newsgroup 221
Newsgruppe 4
NFS 107, 174–179
Client 174
Client Einrichten .. 178
Export 174
root_squash 177

-
- Server 174
 - Server Einrichten .. 176
 - NFS-Export 369
 - NFS-Daemon
 - lockd 176
 - mountd 176
 - nfsd 176
 - portmap 176
 - statd 176
 - Nikto 371
 - NIS 156–166, 369
 - Client Einrichten .. 161
 - Domäne 156
 - Fehlerdiagnose 164
 - Map 156
 - Master-Server 156
 - Server Einrichten .. 157
 - Slave-Server 157
 - nmap 371
 - NNTP 222
 - nslookup 122
 - Nutzdaten 97
- O**
- Offline-Betrieb 211
 - Online-Betrieb 211
 - Open-Source Software .. 29
 - Open-Source-Modell ... 4
 - OpenOffice 29
 - OpenSSH 307, 309
 - Automatische
 - Authentifizierung ... 312
 - /etc/ssh/sshd_config 309
 - sshd 309
 - X11 Weiterleitung . 310
 - OpenSSL 306
 - Outlook 277
 - OutputStream 151
 - Overhead 18, 24, 106
- P**
- Packet Sniffer 243
 - Packet Switching 9
 - Paket
 - fehlerhaftes 19
 - gefälschtes 257
 - gekapseltes 101
 - maßgeschneidertes 259
 - Pakete 9
 - Paketfilter 317–321
 - Architektur 318
 - Filterregeln 319
 - Funktionsumfang . 318
 - Regelformate 320
 - Paketvermittlung ... 9, 23
 - Paketvermittlungsrechner
 - 9
 - Parallelisierung 145
 - Partnerinstanz 17
 - Password Guessing .. 237
 - Passwort 237, 240
 - PAT 89
 - Payload 18
 - PC 28
 - PCI-Netzwerkkarte ... 41
 - Peer-to-Peer-
 - Kommunikation 140
 - Penetration Testing .. 367
 - Perl 245
 - Personal Computer ... 27
 - Pfad 23
 - Plug-In 247
 - Pluggable Authentication
 - Modules 310
 - Point-to-Point-Protocol .. 22
 - POP 202–205
 - Phasenmodell 203
 - POP3 197
 - PoPToP 361
 - Port 141, 145, 147
 - Private 142
 - Well-known .. 141, 435
 - Port and Address
 - Translation 92
 - Port and Adress
 - Translation 89
 - Portmapper 369
 - Portnummer ... 104, 141, 144, 145, 147
 - postfix 424
 - Postfix 208
 - PPP 51
 - CCP 59
 - DSL-Verbindung ... 62
 - IPCP 52
 - ISDN-Verbindung .. 58
 - LCP 51
 - Modemverbindung . 54
 - MPPP 59
 - pppd 55
 - PPPoE 61, 63, 391
 - PADI 63
 - PADO 63
 - PADR 63
 - PADS 64
 - Präsentationsprogramm . 29
 - Preshared Keys . 285, 296
 - /etc/psk.txt ... 296
 - Primary Domain
 - Controller 182
 - PrintWriter 151
 - Priorität 99
 - Privatkunde 74
 - Privatsphäre 236, 239
 - Procmail 208
 - Programmierer ... 28, 147
 - Programmiererteam .. 28
 - Programmierfehler .. 246
 - Programmierschnittstelle
 - 142, 143
 - Programmiersprache . 20, 142
 - C 143
 - Java 143, 150, 246
 - JavaScript 246
 - Protokoll 16
 - höerschichtiges ... 24
 - unzuverlässiges ... 106
 - verbindungsloses .. 66, 106
 - Protokoll-Header 17
 - Protokoll-Trailer 17
 - Protokollfamilie 144
 - AF_INET 144
 - AF_UNIX 144

- Protokollieren 239
- Protokollimplementierung
18
- Protokollinformation . 17
- Protokollkommunikation
16
- Protokollstack 4
- Proxy Server 317, 321–324
 - Architektur 321
- Prozeß 142
- Prozeduraufruf
 - entfernter 20
- Prozess 101
- Puffer 146
- Punkt-zu-Punkt-
 - Verbindung 51
 - Virtuelle 357
- Punkt-zu-Punkt-
 - Verbindung 100
- Punktnotation 13, 69
- Q**
- Qualität 67
- Quelladresse 99, 103
- Quellcode 29
- R**
- r* Dienste .. 244, 272, 275,
307
- racoona 291, 296
 - /etc/racoona.conf .
296
- Raw Sockets 258
- RawIP 57
- RDP 189
 - Clients 189
 - Server 189
- read() 146
- Rechner
 - blind fernsteuern .. 272
- Rechnerkommunikation .
20
- Rechnername 13
- RedHat 30
- Referenzmodell 5
- Reihenfolge 102, 106
 - von Paketen 24
- Reinfektion 276
- Replay Angriff 243
- Replay Angriff 284
- Resolver 118
- Ressourcen 23
- Restrisiko 240
- Richtlinie 253
- RIP 88
- RIPE 74
- Risikoabschätzung
235–255
- Risikoanalyse ... 235, 251
- RJ45 39
- Road Warrior 359
- Robustheit 13, 23
- Root Zertifikat 300
- Routen
 - Default 82
 - Host 81
 - Netzwerk 81
- Router 23, 68, 78
- Routing 19, 22, 66, 78–89,
344
 - Algorithmus 96
 - Metrik 78
 - Protokoll 88
 - Tabelle 80, 412
- Routing Information
 - Protocol 88
- Routing-Algorithmus . 23
- Routing-Protokoll 23
- Routing-Tabelle 237
- RPC Dienst 157
- rsh 369
- S**
- Samba 179–185
 - Benutzerverwaltung ...
184
 - Server Einrichten .. 182
 - smb.conf 184
 - Zugriffskontrolle .. 184
- Samba-Server 403
- SAP R/3 29
- SATAN 367
- Satzsystem 29
- Schicht 15
 - anwendungsbezogene .
24
 - höhere 15
 - Instanz einer 16
 - tiefer 15
- Schichtenmodell 5, 15–18,
20
 - Datenfluss im 16
- Schlüsselmanagement ..
308
- Schutzbedarf 238
- Schutzmaßnahme ... 243
- Schutzvorkehrung ... 240
- Schwachstellen . 257, 272
- scp 172, 309, 312
- SCP 406
- Screened Subnet 329
- Screened-Host 328
- SDSL 60
- Second-Level-Domain ...
111
- Secure Shell 244
- Secure Socket Layer . 244
- Security Association
 - Database 292
- Security Policy 286
- Security Policy Database .
292
- Seitenbeschreibungsspra-
che 228
- Sender 78
- Sender-IP-Adresse
 - Fälschen einer . *siehe* IP
Spoofing
- Senderadresse 68
- Sendeversuch 102
- Sendewiederholung . 106
- sendmail 244
- Sendmail 208, 275
- Sequence Number
 - Guessing ... 238, 273
- Sequenznummer 104
- Serial-Line IP 22
- Server 13, 14, 106, 139
 - Lokalisierung 14

-
- Lokalisierung des . 141, 142, 147
 - Server, Adresse des .. 147
 - ServerSocket 152
 - setkey 291
 - /etc/setkey.conf . 292, 296
 - sftp 309
 - Shorewall 333
 - Sicherheitsdienste
 - IPSec 284
 - Sicherheit 3, 96, 236
 - auf Netzebene 251
 - auf Rechnerebene . 250
 - Sicherheits-Level 250
 - Sicherheitsüberprüfung . 367
 - Fehlalarm 378
 - Sicherheitsanforderung .. 251
 - Sicherheitsaspekte 5
 - Sicherheitsbeauftragter .. 251
 - Sicherheitsbegriff 235
 - Sicherheitskonzept .. 235, 236, 248–255
 - Sicherheitslücke .. 3, 240, 243
 - Sicherheitslücken 370
 - Sicherheitslücken ... 257, 267, 275, 281
 - Sicherheitsmechanismus . 252
 - Sicherheitspolitik ... 251, 316, 320
 - Sicherheitsprozess ... 248
 - Sicherheitsrisiko .. 6, 235, 236
 - Sicherheitssetup 321
 - Sicherheitsteam 254
 - Sicherheitstechnologie ... 254
 - Sicherungskomponente .. 24
 - Sicherungsschicht 19
 - Signal
 - elektrisches 15, 16
 - optisches 16
 - Simple Network Management Protocol 245
 - SMB 180
 - smpppd 55
 - SMTP ... 24, 147, 197–202
 - Kommandos 199
 - Sitzung 201
 - Smurf Angriff .. 265, 271, 280
 - Sniffing 370
 - SNMP 245
 - Snort 380–385
 - Betriebsmodi 381
 - NIDS Konfiguration ... 382
 - Regeln 383
 - Socket 105, 258, 302
 - socket() 143, 148
 - Socket-Modus 145
 - Socket-Schnittstelle 142–153
 - Socket-Typ 144
 - SOCK_DGRAM 144
 - SOCK_STREAM 144
 - Software 29
 - SOHO 389
 - Source IP-Adresse 68
 - Sprungadresse 99
 - Squid 348–354
 - Konfiguration 348
 - Zugriffskontrolle .. 350
 - ssh 244
 - ssh 309
 - Entfernter
 - Programmaufruf 311
 - Verbindungen Tunneln 311
 - X11 Weiterleitung . 311
 - SSH ... 186, 307–313, 406
 - Authentication Protocol 308
 - Client 186
 - Connection Protocol ... 309
 - Host Key 308
 - Sicherheitsdienste . 307
 - Transport Layer Protocol 308
 - X11 Weiterleitung . 309
 - ssh-agent 313
 - SSL 302–306
 - Handshake Protocol ... 304
 - Record Layer 303
 - Schlüsselmanagement . 303
 - Steuerprotokoll 303
 - ST-II 96
 - Stabilität
 - von Linux 28
 - Stallmann, Richard ... 29
 - Standardsoftware
 - betriebswirtschaftliche 29
 - Stateful Firewall 342
 - Struktur
 - interne 250
 - Strukturanalyse 248
 - stunnel 306
 - Subnetz 12, 71–73, 78
 - Subnetzmaske 71–73
 - Sun 246, 274
 - Superserver 141
 - SUSE 4, 30
 - SUSE
 - Hardwaredatenbank 34
 - SuSE-Security
 - Announcements . 378
 - SuSEFirewall2 68, 94, 343–348, 417
 - Switch 40
 - Synchronisationspunkt .. 20
 - SyncPPP 57
 - Systemadministration 25, 29
 - Systemadministrator . 76

- Systemaufruf 143
- Systemdatei
 - /etc/group 156, 158, 175
 - /etc/gshadow ... 158
 - /etc/hosts 147
 - /etc/nsswitch.conf 161
 - /etc/passwd 156, 158, 175
 - /etc/services . 105, 149
 - /etc/shadow 156, 158
 - /etc/xinetd.conf . 141, 149
- Systemprogramm 29
- T**
- T-Online 238
- Tanenbaum, Andrew . 28
- Target 337, 338
- TCP 24, 101, 140
 - Acknowledgement-Nummer 103
 - Destination Port Number 102
 - Instanzen 102
 - Port 104
 - Segment 102, 104
 - Sequenznummer .. 102
 - Source Port Number ... 102
 - Verbindung 104
- TCP SYN Flooding .. 243, 266–267, 271, 273, 280
- TCP Verbindung
 - Desynchronisation von 271
- TCP/IP 3, 65–107
- TCP/IP Protokolle ... 257
- TCP/IP
 - Implementierungen 10
 - Modell 10
 - Protokollsuite 20
- TCP/IP-Protokollstack 25
- TCP/IP-Protokollsuite 20
- tcpdump 296
- Teilnetz 10, 12
- Telearbeit 356
- Telefonleitung 9
- Telefonwählleitungen . 22
- Telnet .. 24, 185, 237, 244, 307
- Terminal
 - virtuelles 24
- Textverarbeitung 29
- TFN 270
- TFN2k 270
- TFN3k 271
- TFTP 369
- The Cult of the Dead Cow 279
- Timer 102
- TinyCA 306, 413
- TLS 306
- Token Ring 12
- Token Bus 21
- Token Ring 21, 50, 78
- Top-Level-Domain .. 110
- Torvalds, Linus 28
- Torwächter 315
- Transmission Control Protocol 24
- Transmission Control Protocol 65, 101
- Transparenter Proxy 346, 354
- Transparenz 247
- Transportinstanz 24
- Transportprotokoll .. 101, 106, 140
- Transportschicht 19, 23–24
- Treiber 22, 28
- Tribe Flood Network 267
- trin00 267, 269
- Trojanische Pferde .. 274, 278, 281
- Trojanisches Pferd .. 238, 241
- Tunneling 100, 357
- Tunnelrechner 101
- Twisted-Pair 39
- U**
- UDP 24, 106–107, 140
 - Overhead 106
 - Port 104
- UDP Flooding 271
- Überflutung
 - eines Empfängers . 237
- Übertragung
 - sichere 20
 - unzuverlässige . 66, 106
 - verbindungslose ... 24, 106
 - verbindungsorientierte 102
 - zuverlässige ... 24, 102
- Übertragungsrate 21
- Übertragungstechnik . 23
- Uniform Resource Locator 226
- UNIX 4, 28, 29
 - Berkeley 10
- UNIX System V 274
- Unix-Mailbox 208
- UNIX-Variante 4
- URL 228
- USAGI Projekt 290
- USB-Netzwerkkarte .. 41
- User Datagram Protocol . 24, 65, 106–107
- V**
- Verarbeitungsschicht . 24
- Verbindung 19, 140
 - bidirektionale 140
 - halb-offene 267
 - physikalische 19
- Verbindungsabbau .. 140, 142
- Verbindungsaufbau . 140, 142
- Verbindungsaufbau-
wunsch 104
- Verbindungsaufbauphase 103
- verbindungsorientiert 24
- Verfügbarkeit 236

- Verkehrsanalyse 237
 Verkehrsfluß 237
 Vermarktung 30
 Vermittlungsrechner .. 17
 Vermittlungsrechnern 23
 Vermittlungsschicht .. 19
 Verschlüsselung 243
 Verschleiern 250
 Verschleierungstechniken
 276
 Verteidigungsmaßnahme
 235
 Verteiltes Vertrauen .. 275
 Vertrauensbeziehungen
 Ausnutzen . 272–274,
 280
 Vertraulichkeit .. 97, 236,
 257
 Verwaltungswerkzeug 30
 Verzögerung 67
 Verzeichnisdienst 71
 Videodaten 97
 Videodatenstrom 98
 Videokonferenz 3
 Virenbefall 242
 Virens Scanner 242, 244
 Virtual Private Networks
 101
 Virtuelle private Netze ..
 355–366
 Virus 238, 241, 246
 VisualBasic-Skript ... 277
 VNC 187
 Server 188
 Viewer 188
 Vorhersage
 von Sequenznummern
 243
 VPN 101, 413
 VPN-Funktionalität ... 44
 vsftp 244
 vsftpd 171
- W**
 WAN 52
 Warteschlange 149
 Web-Anwendung ... 246
 Web-Browser 4, 228
 Web-Seite .. 229, 244, 245
 Header 229
 Web-Server 247, 422
 Wegewahl .. 19, 22, 78–89
 Weitverkehrsnetz . 12, 22,
 355
 Well-known Ports ... 435
 Werkzeug 28
 Wide Area Network .. 21
 Wiedereinspielen 243
 Windows 4, 29
 Wireless LAN 43
 Wirtsprogramm 241
 WLAN 12, 43, 399
 Übertragungsgeschwin-
 digkeit 43
 ESSID 46
 NWID 46
 Verschlüsselung 46
 WEP 46
 World Wide Web 226–231
 World-Wide Web .. 6, 25,
 141, 244
 write() 146
 WU-FTP 172
 Wurm 238, 241, 274
 wvdial 55
 wvdial 56
- X**
 X Window 29
 X-Display Umlenken 186
 X-Window-System .. 186
 X.25 12
 X.509 Zertifikat . 296, 299,
 303, 413
 xinetd 141
- Y**
 YaST 30
 Yellow Pages (YP) ... 156
 YP-Kommandos
 ypcat 165
 ypchfn 165
 ypchsh 165
 ypdomainname ... 164
 ypmatch 165
 yppasswd 165
 ypset 164
 ypptest 165
 ypwhich 164
- Z**
 Zahlenkette 13
 Zertifikat *siehe* X.509
 Zertifikat
 Zieladresse 99, 103
 Zielnetz 81
 Zielrechner 19, 22, 78, 106
 Zugang
 physikalischer 237
 Zugangskontrolle ... 244
 Zugangsnetztechnologien
 22
 Zugangsrechte 253
 Zuordnungstabelle ... 91
 Zuteilungsverfahren .. 74
 Zuverlässigkeit 9