

Olaf Borkner-Delcarlo: *Das Samba-Buch*



Dr. Olaf Borkner-Delcarlo

Das Samba-Buch

Samba einrichten und konfigurieren

4., überarbeitete 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 3-89990-121-5

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

Umschlaggestaltung: Fritz Design GmbH, Erlangen

Gesamtlektorat: Nicolaus Millin

Fachlektorat: Matthias Eckermann, Klaus Kämpf, Frank Lemser und Volker Lendecke

Satz: L^AT_EX

Druck: Kösel, Krugzell

Printed in Germany on acid free paper.

Vorwort zur vierten Auflage

Der Anlaß für die Überarbeitung zur vierten Auflage ist natürlich das Erscheinen der Version 3.0.9 des Samba-Pakets. Die Änderung der Major-Release-Number rechtfertigt gewiß diesen Schritt.

Die Version 3.0.9 verfügt unter anderem über erweiterte Kompetenzen als Primary Domain Controller. Natürlich geht diese Auflage auf die erweiterte Funktionalität der Version 3.0.9 ein. Aber auch diese Version kann nicht alle Funktionen in einem LAN-Netz erfüllen. So kann auch die Version 3.0.9 einen Windows 2000-Server nicht ersetzen, und auch die Funktion eines Backup Domain Controllers kann von der gegenwärtig gültigen Version nur in Verbindung mit einem Samba-Primary Domain Controller ausgeübt werden.

Das gesamte Buch ist vollständig überarbeitet worden. Die Kapitel 8 auf Seite 325 «*Logon Server*», Kapitel 9 auf Seite 337 «*Domain Controller*», Kapitel 10 auf Seite 375 «*Windows 200x/XP Professional in eine Domain einfügen*» sowie Kapitel 13 auf Seite 417 «*Der Daemon winbindd*» sind neu hinzugefügt worden. Ebenso ist ein neues Kapitel eingefügt worden, welches sich mit dem Einsatz von LDAP beschäftigt (Kapitel 12 auf Seite 383 «*LDAP und Samba*») Das Kapitel über *Security* enthält genauere Informationen, wie sich beide Betriebssystem-Modelle, Microsoft und Unix, in ihrer Konzeption unterscheiden und wie die Methodik aufeinander abgebildet wird.

Aufbau des Buches

Das Buch ist sehr einfach und linear strukturiert, dabei ist eine gewisse Redundanz in der Behandlung einzelner Themen nicht zu vermeiden. Besonders die Parameter tauchen in verschiedenem Kontext auf, werden dann aber in Abhängigkeit der Problemstellung neu behandelt. Die Alternative wäre eine unstrukturierte Ansammlung von Kapiteln, die gegenseitig aufeinander verweisen und so das Lesen gewiß nicht einfacher machen. Um auch Lesern mit wenig Zeit einen Einstieg in das Thema Samba zu vermitteln, sind die Kapitel so gehalten, dass sie alle *für sich* gelesen werden können. Ich hoffe, dass auf diese Weise diese Leserschaft leichter Zugang zu den sie interessierenden Themen erhält.

Teil 1 des Buches hat einfache Installationsaufgaben zum Thema, die nicht unbedingt für den Einsatz in einer realen Netzumgebung taugen müssen. Angefangen von einem ersten Kennenlernen der Installation führt der Inhalt über zur Installation von Microsoft-Clients der älteren Bauart. Es folgt ein Kapitel, das sich mit den neuen Betriebssystemen der Firma Microsoft und deren Einbindung in ein Samba-Netz beschäftigt. Dann werden einfache Installationen beschrieben

und gleichzeitig wird ein wenig auf die Struktur der zentralen Konfigurations-Datei, der `/etc/samba/smb.conf`, eingegangen. Danach werden etwas komplexere Installationen vorgestellt, und auch der Ausblick auf die Theorie wird verfeinert. Die vermittelte Theorie ist bis zu diesem Kapitel freilich eher zweckgebunden. Die Darstellungen erläutern grundlegende Prinzipien des SMB-Protokolls und wie diese von Samba implementiert werden. Hier endet der erste Teil des Buches.

Teil 2 behandelt einfache praktische Installationen. Angefangen von der Planung eines Netzes über die Funktion eines Browsers bis hin zu einer kompletten Installation eines Primary Domain Controller werden die wichtigsten Funktionen innerhalb eines LAN-Netzes behandelt. Das geschieht in Kapiteln, die in sich abgeschlossenen sind. «*Browsing*», «*Logon-Server*», und «*Primary Domain Controller*» bilden die zentralen Themen dieser Kapitel. Nun folgen eine Reihe von immer komplexer werdenden praktischen Beispielen.

Am Ende von Teil 2 findet man einige Tips und Tricks und nützliche Konfigurationen, die bei professionell genutzten Installationen hilfreich sein können.

Teil 3 widmet sich der «*Referenz*» des Samba-Pakets. Hier werden alle Parameter erklärt und zwar bis zur Version 3.0.9. Die einzelnen Parameter sind, soweit es sich um Parameter der neueren Versionen handelt (\geq Version 2.0.x), mit ihren Release-Nummern kenntlich gemacht, so dass der Anwender einen Überblick erhält, welche Parameter er für seine spezielle Netzinstallation verwenden kann und welche noch nicht Teil seiner Samba-Version sind.

Es schließt sich die Behandlung der Hilfs-Werkzeuge an, von denen das Samba-Paket eine ganze Reihe zur Verfügung stellt.

Seit dem Erscheinen der Version 3.0.9 unterstützt Samba als Domain-Mitglied auch die Eigenschaft, Windows NT-Domain-Clients zu authentifizieren. Das konnte ab der Version 2.2.1a bereits eingerichtet werden, aber dazu gehörten profunde Kenntnisse beider Betriebssystem-Seiten. Jetzt ist es möglich, die unterschiedlichen Clients über einen `winbindd/PAM`-Mechanismus zu authentifizieren, ohne dass spezielle Kunstgriffe auf der Samba-Server-Seite nötig sind. Der Daemon `winbindd` wird in diesem Teil des Buches ausführlich behandelt.

Teil 4 beschäftigt sich neben anderen Themen mit Informationen des «*Versionswechsels*». Aber auch Themen wie «*Die Theorie über Sicherheit und Authentifizierung*» bestimmen den Inhalt dieses Teils. Es folgt ein Kapitel über «*Diagnose*»

und Optimierung» eines Samba-kontrollierten LAN-Netzes. Genau wie das Kapitel «Probleme beheben» sind diese Kapitel nur dann wichtig, wenn einmal etwas nicht funktioniert oder die Performance des Netzes zu wünschen übrig läßt.

Der Anhang beinhaltet alle Themen, die nicht ursächlich zum Subjekt Samba gehören, aber doch im Kontext einer Netz-Administration von Bedeutung sind. Viele fähige Netz-Administratoren sind in einer reinen Microsoft-Welt groß geworden. Die Philosophie der beiden Betriebssysteme, Unix- und Microsoft, unterscheidet sich doch recht stark voneinander, und es ist für einen Microsoft-Administrator nicht selbstverständlich, die Mechanismen der Unix-Welt zu kennen. Aus diesem Grund bildet der Anhang des Buches eine kleine Einführung in die Grundlagen des Betriebssystems Unix. Das Glossar bildet den Abschluß des Buches.

Neubeuern, im Januar 2005

Dr. Olaf Borkner-Delcarlo

Inhaltsverzeichnis

I	Installieren und Einrichten von Samba	1
1	Die erste Installation	5
1.1	Einleitung	5
1.2	Was ist Samba?	6
1.2.1	Was kann Samba, was kann es nicht!	8
1.2.2	Was kann Samba Version 3.0.9?	10
1.3	Der Einsatz von Samba	11
1.4	Verbinden mit dem Netz	12
1.5	Ein erster Test	13
1.6	Die erste Installation	14
1.7	Wenn man kein SuSE Linux verwendet	16
1.8	Samba aktivieren	17
1.8.1	Einrichten des Servers	21
1.8.2	Clients auf Samba vorbereiten	23
1.8.3	Share – User	23
1.8.4	Samba mit nur einem Rechner	24
1.9	Ein einfaches Login	25
1.10	Erste Konfiguration eines Windows-Clients	35
1.10.1	Funktioniert nicht?	37
1.10.2	Security-Einstellung bei neueren Samba-Versionen	43
2	Windows Clients konfigurieren	45
2.1	Windows 95/98/Windows XP	45
2.2	Einrichten von Windows 95/98 Clients	47
2.3	Windows einrichten	59
2.4	Einbindung verschiedener Clients	59
2.5	DOS- und Windows 3.1-Clients	59
2.6	OS/2-Clients	62
2.7	Windows for Workgroups	63

Inhaltsverzeichnis

2.8	Windows NT-Clients	65
2.9	Windows 2000	69
2.9.1	Einrichten von Windows 2000	71
3	Einfache Konfigurationen	79
3.1	Linux als Server für Windows	80
3.2	Die Konfigurationsdatei <code>/etc/samba/smb.conf</code>	80
3.3	Samba-Konfiguration für ein kleines Netzwerk	86
3.3.1	Aufbau des Netzwerks	87
3.3.2	Die Datei <code>/etc/samba/smb.conf</code> des Servers supporto	89
3.3.3	Die Datei <code>/etc/samba/smb.conf</code> des Servers develop	91
3.3.4	Die Datei <code>/etc/samba/smb.conf</code> des Servers delcarlo	92
3.4	File- und Print-Server	93
3.4.1	File-Server	93
3.4.2	CD-ROM-Server	96
3.5	Print-Server (für Windows NT)	97
3.6	Einige Beispiel-Konfigurationen	106
3.6.1	Vergabe von Rechten	107
3.6.2	Syntax der Rechtevergabe	108
3.6.3	Der Export des gesamten Linux-Dateisystems	109
3.6.4	User-Kennung	110
3.6.5	Samba konfigurieren	111
4	Samba für Fortgeschrittene	115
4.1	Motivation für den professionellen Einsatz	115
4.2	Das Dateiformat der <code>/etc/samba/smb.conf</code> -Datei	116
4.3	Struktur der <code>/etc/samba/smb.conf</code> -Datei	117
4.4	Sections	118
4.4.1	Shares	118
4.4.2	Die <code>[global]</code> -Section	120
4.4.3	Die <code>[homes]</code> -Section	121
4.4.4	Die <code>[printers]</code> -Section	131
4.5	Name Mangling	139
4.6	Grundkonfigurationen	140
4.6.1	Unterschiede im Sicherheitsmodell	140
4.6.2	Das Microsoft Authenticated RPC-Protocol	146
4.7	Ein komplexes Beispiel	150
4.7.1	Hilfsprogramme	150
4.7.2	Die <code>[homes]</code> -Section des Beispiels	151
4.7.3	Sicherheit auf der Linux-Seite	152

4.7.4	Sicherheit auf der Windows-Server-Seite	153
4.7.5	Password-Synchronisation	157
4.8	Unverschlüsselte Passwords	164
4.9	Verschlüsselte Passwords	165
4.9.1	Wechsel zu verschlüsselten Passwords in der Praxis	175
4.9.2	Password und Password-Synchronisation	180
5	Samba, Theorie und Hintergründe	183
5.1	Einige Hintergrundinformationen	183
5.2	Wie funktioniert ein Netz-Protokoll?	184
5.3	Was macht das NetBIOS?	186
5.3.1	Was ist NetBIOS	187
5.3.2	NetBEUI	188
5.3.3	Andere Transportprotokolle	189
5.3.4	NetBIOS und Samba	189
5.4	NetBIOS-Namensauflösung	190
5.4.1	IP-Adresse und NetBIOS-Namen	191
5.5	SMB und NFS, ein Vergleich	196
5.5.1	Unterschiede	197
5.5.2	SMB	199
5.5.3	NFS	200
5.6	Samba-Server ins Netz einbinden	201
5.6.1	Namensdienste	203
5.6.2	Datagram-Dienste	206
5.6.3	Session-Dienste	206
5.7	Clients	208
5.8	Zugriffsrechte und Security	209
5.9	Verschlüsselte Passwords	212
5.10	Benutzerabhängige Schreib- und Leserechte	213
5.11	Abbildung von DOS- auf Unix-Attribute	213
5.12	Berechtigungen für neu erzeugte Dateien und Verzeichnisse	214
5.13	Löschen von Dateien	215
5.14	Automatisches Mounnten	215
5.15	Fehlersuche	216
5.15.1	Die User Name/Password-Auswertung	225
5.16	Grundkonfigurationen	226
5.17	Share Level Security	227
5.18	User Level Security	242
5.19	Authentifizierung über NIS	247
5.20	Die Sicherheitsmodelle von Samba	250
5.21	Samba und Passwords	250

II	Samba in der Praxis	255
6	Planung eines Netzes	259
6.1	Die Clients	259
6.2	Server	261
6.3	Planungsleitfaden	262
6.4	Projektierungsphase	263
6.4.1	Kosten sparen mit Linux	266
6.5	Pilotphase	267
6.6	Investitionsphase	273
6.7	Umstellungsphase	273
6.8	Reporting und Administration	275
7	Samba als Browser	277
7.1	Parameter des Browsings	277
7.2	Was ist ein Browser?	281
7.3	User – Workgroup – Domain – LAN	282
7.3.1	LAN – Domain	283
7.3.2	User – Workgroup	284
7.3.3	Subdomains (eigentlich Multidomains)	284
7.4	Die Unix-Domain	285
7.5	Browsing	289
7.5.1	Was macht eigentlich ein Browser?	289
7.6	Local Master Browser	290
7.7	Domain Master Browser	292
7.7.1	Motivation für den Domain Master Browser	293
7.8	Browser-Wahl	295
7.8.1	Backup Browser	298
7.8.2	Browsing über Subnet-Grenzen	301
7.9	Samba in einer NT-Umgebung	310
7.9.1	Domain-Controller	310
7.10	Browsing verhindern	314
7.11	Shares-Voreinstellungen	316
7.12	Browsing Kurzanleitung	317
7.12.1	Fehlersuche und Debugging	320
7.13	Zusammenfassung	321
7.13.1	Browsing über Broadcast	321
7.13.2	Samba als WINS-Server	323
8	Samba als Logon-Server	325
8.1	Wann braucht man einen Logon-Server?	326
8.2	Was leistet ein Logon-Server?	326
8.2.1	Authentifizierung	326

8.2.2	Profilverwaltung	327
8.2.3	System Policies	327
8.3	Konfigurieren eines Logon-Servers	330
9	Samba als Primary Domain Controller	337
9.1	PDC Einrichten Schritt für Schritt	337
9.2	Windows 95- und Windows 98-Clients	338
9.3	Windows NT- und Windows2000-Clients	338
9.3.1	Anmeldescripts	340
9.4	Konfigurieren eines Domain-Controllers	341
9.5	Netztopologien gegenübergestellt	342
9.6	Peer to Peer	346
9.7	NT-Domain	346
9.8	Samba als Mitglied einer Domain	349
9.9	Samba als Domain Controller	360
9.9.1	Welchen Security Level?	361
9.9.2	Einstellungen der smb.conf-Datei	362
9.9.3	Ein komplettes Installationsbeispiel	363
9.9.4	Zugriff für Clients in einem Trusted Netz	368
9.9.5	User automatisch hinzufügen	371
9.9.6	Windows-NT Domain Security	374
10	Windows 200x/XP Professional in eine Domain einfügen	375
11	Aufsetzen eines Backup Domain Controller	379
11.1	Backup Domain Controller Konfiguration	380
11.2	Eine Beispiel-Konfiguration	381
12	LDAP und Samba	383
12.0.1	Was sind Verzeichnisdienste?	383
12.1	Warum sollte man LDAP einsetzen?	383
12.2	Was bedeutet LDAP?	385
12.2.1	Sicherheitsmodell	386
12.2.2	Namensmodell	387
12.2.3	Informationsmodell	387
12.2.4	Funktionsmodell	388
12.2.4.1	Betreten	388
12.2.4.2	Suchen	388
12.2.4.3	Ändern	388
12.3	Installation und Einrichten	388
12.3.1	Wo bekommt man die neueste Version?	389
12.3.2	Übersetzen des Pakets	389
12.4	LDIF	392

Inhaltsverzeichnis

12.5	OpenLDAP	393
12.6	Accounts verwalten in einem LDAP-Directory	396
12.7	OpenLDAP 2.2.5 unter SUSE-9.2	396
12.8	Häufig verwendete LDAP-Tools	398
12.9	Einträge generieren	400
12.9.1	Windows-Maschinen Domain-Login	401
12.10	Samba mit LDAP verbinden	403
12.11	pdbedit	404
12.12	LDAP-basierte BDC-Konfiguration	407
12.13	Einstellungen für PAM- und NSS-Clients	414
12.13.1	PAM-Einrichtung	415
13	Der Daemon winbindd	417
13.1	Pluggable Authentication Modules	419
13.2	Bestimmung der Account- und Group-IDs	420
13.3	Installieren und einrichten	421
13.3.1	Den Samba-Server beim Primary Domain Controller registrieren	423
13.4	Starten von winbind	424
14	Etwas komplexere Beispiele	425
14.1	Samba Version 3.0.9	426
14.1.1	Installieren	427
14.1.2	Logging	427
14.1.3	Hilfsprogramme	428
14.2	Beispiele	428
14.2.1	Kleine Netzwerklösung	430
14.2.2	Subnet-Lösung	432
14.2.3	Gemischte Subnet-Lösung	436
14.2.4	Password-Server	437
15	Tips und Tricks	441
15.1	Rund um den Server	442
15.2	Server nicht sichtbar	442
15.3	Optimieren auf Geschwindigkeit	442
15.4	Optimieren der Clients	450
15.5	Rund ums System	456
15.6	Abbilden von User-Namen (username map)	456
15.7	Verschiedene Rechtevergabe	457
15.8	Locking	458
15.9	Wide Links	458
15.10	Browsing	458
15.11	Read Raw	459

15.12	Write Raw	459
15.13	Umount-Probleme	460
15.14	System-Error 1240	460
15.15	TCP-Stack	460
15.16	Shell- und Share-Verbindung	461
15.17	Winpopup-Nachrichten	462
15.18	Windows NT 4.0 SP3 und AutoCAD	463
15.19	Login unter WinNT 4.0 SP3 und Windows98	464
15.20	16-Bit-Programme unter NT	465
15.21	Microsoft und Visual-C++	465
15.22	Zeitsynchronisation	465
15.23	Samba-Zugriff führt zu automatischer Einwahl ins Internet	466
15.24	Netzwerk-Performance	466
15.25	Samba-Optimierung	467
15.26	Schneller Verbindungsaufbau	469
15.27	Einstellungen für CDs	469
15.28	Optimierung für das Linux-SMB-File System	469
15.29	Groß-/Kleinschreibung	470
15.30	Zu viele Verbindungen offen	471
15.31	Probleme mit 100-MBit-Netzwerk-Karten	472
15.32	Password-Synchronisierung	474
15.33	Sonstiges	475
16	Nützliche Konfigurationen	479
16.1	Share-Dienste	480
16.2	Beschränkung der User-Anzahl	480
16.3	Gastzugang einrichten	481
16.3.1	Browser-Einstellungen	482
16.3.2	NetBIOS-Name	482
16.3.3	Samba-Server als Mitglied in einer Windows NT-Domain	483
16.4	Samba als PDC	490
III	Samba-Referenz	495
17	Samba Version 3.0.9-Referenz	501
17.1	Die Datei smb.conf	501
17.2	Kennzeichnung der Parameter	502
17.3	Erläuterung zu den Parametern	504
17.4	Die Parameter	504
	abort shutdown script	504
	acl compatibility	505

Inhaltsverzeichnis

add group script	505
add machine script	505
add printer command	506
add share command	507
add user script	507
add user to group script	508
admin users	509
afs share	509
afs username map	509
algorithmic rid base	510
allow hosts	510
allow trusted domains	510
announce as	511
announce version	511
auth methods	512
auto services	513
available	513
bind interfaces only	513
blocking locks	515
block size	515
browsable	516
browseable	516
browse list	516
casesignames	517
case sensitive	517
change notify timeout	517
change share command	518
character set	518
client lanman auth	519
client ntlmv2 auth	520
client plaintext auth	520
client schannel	521
client signing	521
client use spnego	522
codingsystem	522
comment	523
config file	523
copy	524
create mask	524
create mode	525
csc policy	525
deadtime	526

debug hires timestamp	526
debug level	527
debug pid	527
debug timestamp	527
debug uid	527
default case	528
default devmode	528
default	529
default service	529
delete group script	529
delete printer command	529
delete readonly	530
delete share command	530
delete user from group script	531
delete user script	532
delete veto files	533
deny hosts	534
dfree command	534
directory	534
directory mask	534
directory mode	535
directory security mask	535
disable netbios	536
disable spoolss	536
display charset	537
dns proxy	537
domain logons	538
domain master	538
dont descend	539
dos charset	539
dos filemode	540
dos filetime resolution	540
dos filetimes	541
enable rid algorithm	541
encrypt passwords	542
enhanced browsing	543
enumports command	543
exec	544
fake directory create times	544
fake oplocks	544
follow symlinks	545
force create mode	545

Inhaltsverzeichnis

force directory mode	545
force directory security mode	546
force group	547
force security mode	547
force user	548
fstype	548
get quota command	549
getwd cache	550
group	550
guest account	551
public	551
guest ok	551
guest only	551
hide dot files	552
hide files	552
hide special files	552
hide unreadable	553
hide unwriteable files	553
homedir map	553
hosts msdfs	554
hostname lookups	554
hosts allow	554
hosts deny	556
hosts equiv	556
idmap backend	557
idmap gid	557
idmap uid	557
include	558
inherit acls	558
inherit permissions	558
interfaces	559
invalid users	559
keepalive	560
kernel change notify	561
kernel oplocks	561
lanman auth	561
large readwrite	562
ldap admin dn	562
ldap delete dn	563
ldap filter	563
ldap group suffix	563
ldap idmap suffix	563

ldap machine suffix	564
ldap passwd sync	564
ldap port	564
ldap replication sleep	565
ldap server	565
ldap ssl	566
ldap suffix	566
ldap user suffix	566
level2 oplocks	567
lm announce	568
lm interval	568
load printers	569
local master	569
lock dir	569
lock directory	569
locking	570
lock spin count	570
lock spin time	571
log file	571
log level	571
logon drive	572
logon home	572
logon path	572
logon script	573
lppause command	573
lpq cache time	574
lpq command	574
lppresume command	575
lprm command	576
machine password timeout	576
magic output	576
magic script	577
mangle case	577
mangled map	578
mangled names	578
mangle prefix	580
mangling char	580
mangling method	580
map acl inherit	581
map archive	581
map hidden	582
map system	582

Inhaltsverzeichnis

map to guest	582
max connections	583
max disk size	584
max log size	584
max mux	585
max open files	585
max print jobs	585
max protocol	586
max reported print jobs	586
max smbd processes	587
max ttl	587
max wins ttl	587
max xmit	588
message command	588
min passwd length	589
min print space	590
min protocol	590
min wins ttl	591
msdfs proxy	591
msdfs root	591
name cache timeout	592
name resolve order	592
netbios aliases	593
netbios name	593
netbios scope	594
nis homedir	594
nt acl support	595
ntlm auth	595
nt pipe support	595
nt status support	596
null passwords	596
obey pam restrictions	596
ole locking compatibility	597
only user	597
oplock break wait time	597
oplock contention limit	598
oplocks	598
os2 driver map	599
os level	599
pam password change	599
panic action	600
paranoid server security	600

passdb backend	600
passwd chat	602
passwd chat debug	603
passwd chat timeout	603
passwd program	604
password level	604
password server	605
path	607
pid directory	608
posix locking	608
postexec	609
preexec	609
preexec close	609
preferred master	610
prefered master	610
preload	610
preload modules	611
preserve case	611
print ok	611
printable	611
printcap	612
printcap name	612
print command	613
printer admin	615
printer	615
printer name	615
printing	615
private dir	616
profile acls	616
queuepause command	617
queueresume command	617
read bmpx	618
read list	618
read only	618
read raw	618
realm	619
remote announce	619
remote browse sync	620
restrict anonymous	620
revalidate	620
root	621
root dir	621

Inhaltsverzeichnis

root directory	621
root postexec	622
root preexec	622
root preexec close	622
security	622
security mask	627
server schannel	628
server signing	628
server string	629
set directory	629
set primary group script	630
set quota command	630
share modes	631
short preserve case	631
show add printer wizard	632
shutdown script	633
smb passwd file	633
smb ports	634
socket address	634
socket options	634
stat cache	635
strict allocate	635
strict locking	636
strict sync	636
sync always	637
syslog	637
syslog only	638
template homedir	638
template primary group	638
template shell	639
time offset	639
time server	639
unicode	640
unix charset	640
unix extensions	640
unix password sync	641
update encrypted	641
use client driver	642
use mmap	643
user	643
users	643
username	643

username level	645
username map	646
use sendfile	649
use spnego	649
utmp	649
utmp directory	650
-valid	650
valid users	650
veto files	651
veto oplock files	652
vfs object	653
vfs objects	653
volume	653
wide links	654
winbind cache time	654
winbind enable local account	655
winbind enum groups	655
winbind enum users	655
winbind separator	656
winbind trusted domains only	656
winbind use default domain	657
wins hook	657
wins proxy	661
wins server	661
wins support	662
workgroup	662
writable	662
writeable	662
write cache size	663
write list	663
write ok	664
write raw	664
wtmp directory	664
18 Werkzeuge für Samba	665
18.1 smbdc	665
18.2 nmbd	669
18.3 Swat – Samba Web Administration Tool	671
18.3.1 Installieren von swat	674
18.3.2 Zugriff auf swat absichern	677
18.3.3 inetd	679
18.4 smbclient	680
18.4.1 Testen mit smbclient	694

Inhaltsverzeichnis

18.5	smbstatus	697
18.6	secrets.tdb	698
18.7	smbcalcs	698
18.8	testparm	698
18.8.1	Testen mit testparm	699
18.9	testprns	705
18.10	smbpasswd	706
18.10.1	Die smbpasswd-Datei	707
18.10.2	Konfiguration der smbpasswd-Datei	713
18.10.3	Pro und Kontra der Verschlüsselung	714
18.11	addtosmbpass	715
18.12	convert_smbpasswd	715
18.13	pwdump.exe	715
18.14	smbmount	716
18.15	smbumount	718
18.16	lmhosts	719
18.17	nmblookup	722
18.18	rpcclient	723
18.19	Mit smbtar Backups erzeugen	727
18.19.1	Die Syntax von smbtar	731
18.19.2	Beispiele für die Verwendung von smbtar	733
18.20	smbcontrol	734
18.21	Samba-Administration und KDE	738
18.22	net-Administration Werkzeug	745
IV Samba maßgeschneidert		747
19 Versionsänderungen		751
19.1	Wechsel von 2.x → 3.0.x	751
19.2	Wechsel von 1.9.18.x → 2.0.x	753
20 Anpassen von Samba		759
20.1	Installieren von Samba	759
20.1.1	Wann ist eine Neuinstallation notwendig?	760
20.2	Umfang der Software	761
20.3	Download des aktuellen Samba-Pakets	761
20.4	Die Installation des RPM-Pakets	762
20.5	Installieren des Quellcode	765
20.6	Optionen für das Übersetzen	768
20.7	Update einer Samba-Edition (SuSE Linux)	777
20.8	Noch eine Methode, Samba zu installieren	784
20.9	Andere Editionen	785

21 Samba konfigurieren	793
21.1 Attribute	793
21.1.1 File Locking	793
21.1.2 Locking unter DOS/Windows	794
22 Sicherheit und Authentifizierung	801
22.1 Client-Server-Kommunikation	803
22.2 Sicherheit gegen Fremdzugriffe	805
22.2.1 NetBIOS-Dienste blockieren	805
22.2.2 Testen des Port Filterings	806
22.2.3 Sicherheit gegen lokale Zugriffe	808
23 Diagnose und Optimierung	809
23.1 Diagnose und Benchmarking	809
23.2 Samba optimieren	811
23.2.1 Ein Vergleich	812
23.2.2 Hilfreiche Einstellungen der <code>/etc/samba/smb.conf</code>	814
23.3 Fehlerdiagnose	820
23.4 Testen mit ping	821
23.4.1 ping: keine Antwort	821
23.4.2 100% packet loss	821
23.5 Testen mit Telnet	822
23.5.1 smbld mit Telnet überprüfen	822
24 Probleme beheben	825
24.1 Diagnose	825
24.1.1 Testvoraussetzungen	826
24.1.2 Der Test	827
24.2 Domain-Controller	834
24.3 IP-Adressänderung	835
24.4 Ein Sicherheitsproblem	835
A Etwas Linux/Unix	837
A.1 Unix-Rechte	838
A.2 root – der Superuser	839
A.3 Konzeptionelle Unterschiede	840
A.4 Dateisysteme	842
A.5 Die Struktur von Linux	845
A.6 Was ist ein Terminal, eine Shell?	846
A.6.1 Die Pipes (<code>< ></code>) und die Filter	847
A.7 Befehle und Programme	849
A.7.1 apropos	849
A.7.2 passwd	849

Inhaltsverzeichnis

A.7.3	startx, xinit	849
A.7.4	Drucken unter Linux	849
A.7.5	man (Hilfe)	852
A.7.6	Datei-Operationen	853
A.7.7	Dateibaum-Operationen	853
A.7.8	Prozess-Operationen	855
A.7.9	Booten, Neustarten und Anhalten des Systems	855
A.7.10	Netzwerkbefehle	856
A.8	Der Editor vi	861
A.9	Daemons	864
A.10	Ports und Transfer	866
A.11	Das NFS-Dateisystem	869
A.11.1	Importieren von Dateisystemen	870
A.11.2	Exportieren von Dateisystemen	870
A.11.3	Links	872
B	Glossar		875
	Index		899

Teil I

**Installieren und Einrichten
von Samba**

Dieser Teil des Buches beschäftigt sich mit grundlegenden Fragen: Wie installiert man das Samba-Paket, wie richtet man Samba auf dem Rechner ein und wie verwendet man die Standard-Konfigurationsdateien¹.

Teil I des Buches gibt auch einen ersten Einblick in die Installation und Konfiguration eines Samba-Servers. Diese sollten von jedem Leser, der über mindestens zwei Rechner verfügt, leicht durchgeführt werden können. Dieser Einblick soll dazu dienen, die Vorteile, die Samba bietet, sichtbar zu machen. Wer den Preis der Software eines einfachen SMB-basierten kommerziellen Servers kennt, wird erstaunt sein, dass praktisch alle Dienste, die dieses doch recht teure Produkt liefert, von einem Linux-Samba-Server gleichermaßen erledigt werden können und das bei nachweisbar stabilerem Verhalten und oft besserer Performance.

Für alle, die nur an einem einfachen Netzbetrieb mit zwei oder drei Rechnern interessiert sind und die keine besonderen Eigenschaften des Samba-Pakets verwenden wollen, lassen sich die ersten Kapitel des nun folgenden Teils als einfache Installations- und Einrichtungsanleitung auffassen – die anderen Teile können unberücksichtigt bleiben.

Die Theorie, die hinter Samba steckt, ist vielfältig, diese Vielfalt ist aber nicht für alle Anwendungen notwendig, und manchmal reicht es, nach Anleitung eine einfache Vernetzung zustande zu bringen, ohne sich in komplizierten Konfigurationsanweisungen zu ergehen.

Gerade durch den rasanten Preisverfall der Hardware sieht sich auch ein privater Haushalt oft in der Situation, notgedrungen mehr als einen Rechner zur Verfügung zu haben. Es ist einfach nicht rentabel, einen alten Rechner zu verkaufen. Nicht einmal ein Schüler der ersten Oberschulklasse wird einen zwei Jahre alten Rechner zu einem annehmbaren Preis erstehen wollen, wenn er für einen geringen Aufpreis einen wesentlich leistungsfähigeren Computer bekommen kann. Und bevor ein Computermodell Aussicht hat, von einem Museum für teures Geld angekauft zu werden, können gewiß noch einige Jahre ins Land gehen.

Alle Ausgaben und Mühen bei dem Verkauf eines alten Rechners in Rechnung stellend, erscheint es meist besser, den veralteten Boliden der Sondermülldeponie zu übergeben. Aber diese Lösung kostet neuerdings auch Geld, denn den «Alten» wird man ohne Obulus bei keiner offiziellen Entsorgungsstelle mehr los. Die Preisentwicklung und die fortschreitende Technologie machen einen Verkauf immer zu einem Verlustgeschäft. Ein Rechner, der vor zwei Jahren noch mehrere Tausend Mark kostete, ist aus heutiger Sicht praktisch unverkäuflich. Die bessere Lösung ist es, den alten Rechner weiter zu verwenden. Daraus ergibt sich jedoch fast immer der Wunsch, die im Haushalt befindlichen Rechner zu vernetzen.

¹Wie man eine ältere, bereits laufende Version durch eine neuere Samba-Version ersetzt, wird ausführlich in Teil IV des Buches besprochen.

Ein kommerzielles Server-Betriebssystem ist teuer! Es kostet immer eine Summe die im vierstelligen Bereich liegt (Linux und FreeBSD natürlich ausgenommen). Für den privaten Haushalt läßt sich eine solche Anschaffung nicht rechtfertigen. Man kann sich diese Ausgaben auch sparen, indem man auf dem notwendig gewordenen neuen Server-Rechner Linux den Vorzug gibt. Das Betriebssystem selbst kostet nichts², und die Vernetzungs- und Server-Software kostet ebenfalls nichts. Und solange die Lizenzen nach Anzahl der Clients an einem **Server** bezahlt werden müssen, braucht man für beliebig viele Client-Lizenzen der Betriebssystem-Software keinen einzigen Pfennig zu bezahlen. Zudem macht die Vernetzung von Rechnern einfach Spaß und man lernt eine Menge dabei.

²Sieht man von den Kosten für die Edition selbst ab.

Kapitel 1

Die erste Installation

1.1 Einleitung

Das komplette Samba-Paket ist sehr komplex, und wenn man auch nur einen Überblick über die möglichen Konfigurationen bekommen will, braucht man eine Menge Erfahrung im Einsatz und der Administration von Unix- und Microsoft-Netzen. Die Optimierung eines Netzes und das Ausnutzen aller Möglichkeiten bedürfen noch intensiverer Beschäftigung mit der Materie.

Das soll aber niemanden entmutigen, denn die ersten Schritte, einen Linux-Rechner als Server in einem Microsoft-Netz zu nutzen, sind relativ einfach. Und mit diesen einfachen Schritten wollen wir beginnen. Die dazu notwendige Hardware ist in Abb. 1.1 gezeigt.

Aber selbst mit einem einzigen Rechner lassen sich viele der vorgestellten Beispiele ausprobieren, auch wenn der Spaßfaktor mit mehr als einem Rechner natürlich größer ist. Und auch wenn die Themenstellung des Buches sehr trocken zu sein scheint, so soll doch die Lektüre und das Ausprobieren der Beispiele schließlich trotzdem Freude machen.

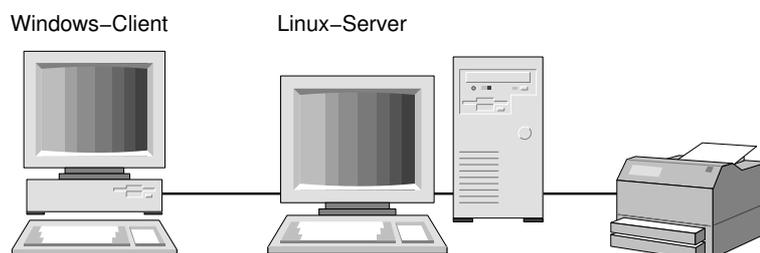


Abbildung 1.1: Ein Linux-Server und ein Windows-Client

Wir werden uns nun eine erste Installation ansehen, ohne jedoch auf die Besonderheiten und Möglichkeiten von Samba einzugehen. Zu diesem Zweck sehen wir uns alle notwendigen Vorgänge an, die zu einer erfolgreichen Installation gehören. Angefangen vom Installieren über das Konfigurieren bis zum Ausprobieren des Beispiels selbst. Später wird uns die Installation noch einmal beschäftigen (siehe Teil IV auf Seite 749). Dort geht es dann etwas komplexer zu, da wir uns dort auch ansehen, wie man das Samba-Paket selbst abändern kann und wie man eine alte Samba-Version durch eine neue ersetzt. Das scheint trivial, ist es aber nicht, denn die einzelnen Linux-Editionen verwenden leider auch verschiedene Installationsorte und verschiedene Startmechanismen, und das kann einem bei einem Update des Samba-Pakets recht große Probleme bereiten.

1.2 Was ist Samba?

Bevor wir nun auf die einzelnen Installationsschritte eingehen, wollen wir uns kurz einmal ansehen, was Samba überhaupt ist. *Samba* besteht aus einer Sammlung von Programmen, mit dessen Hilfe das SMB-Protokoll auf die Unix-Ebene abgebildet werden kann, und das gilt natürlich im besonderen Maße für das Betriebssystem Linux. Mit Samba kann jeder Unix-Rechner in einer Windows-Umgebung als echter Teilnehmer erscheinen, kann aber auch als echter Primary Domain Controller fungieren, vorausgesetzt, man verwendet die neueste Version 3.0.9. Das SMB-Protokoll wird traditionell in kleinen Netzwerken mit Microsoft- und/oder OS/2-Clients verwendet, die nicht mit entfernten Netzen kommunizieren müssen. Samba bietet durch diese Emulation die Möglichkeit, die Windows-Welt mit der Unix-Welt zu verbinden. Eine Linux-Server-Maschine kann eine Microsoft-Netzwerk-Server-Maschine (fast) vollständig ersetzen und bietet daher auch die folgenden Dienste:

- ❑ Mit der Version 3.0.9 kann ein Samba-Server als *Logon Server* dienen, er kann für diese Systeme daher auch die Domain-Registrierung vornehmen.
- ❑ Ein Samba-Server kann als *Share-Server* für eine Anzahl Windows-Clients dienen, dabei stellt der Samba-Server die Shares bereit, welche normalerweise ein Microsoft-Server bereitstellen würde.
- ❑ Auch wenn ein Samba-Server als Primary Domain Controller in einem Windows-Netz agieren kann, so ist auch in der Version 3.0.9-Version die Implementation nicht vollständig. Sie ist aber hinreichend, denn die Authentifizierung von Benutzern an Windows NT-Workstations kann mit einem Samba-Server einfach bewerkstelligt werden.
- ❑ Ein Samba-Server kann *Drucker* den Netzteilnehmern zur Verfügung stellen, die entweder über die Clients selbst oder über den Server verfügbar gemacht werden.

- Ein Samba-Server kann das vollständige *Browsing* in einem Netz mit Microsoft-Client-Rechnern übernehmen (in Kapitel 7 auf Seite 277 wird das Thema *Browsing* genauer erläutert).
- Ein Samba-Server kann die *Authentifizierung von Clients* übernehmen, die sich in eine Microsoft-Domain einloggen wollen.
- Ein Samba-Server kann als vollwertiger Ersatz für einen *WINS-Server* dienen.

Der Samba-Server der Version 3.0.9, basiert auf zwei *Daemons*, ohne die der Server nicht funktionieren kann. Da ist einmal der `smbd`-Daemon, der sich um die Share-Zuteilung kümmert und der `nmbd`-Daemon, der die Namensdienste bereitstellt.

- *smbd*-Daemon:

Der `smbd`-Daemon überwacht das Netzwerk, ob ein Client eine sogenannte `share-request`-Anforderung an den Server stellt. Wird eine solche Anfrage erkannt, so wird die angeforderte Ressource dem nachfragenden Client zur Verfügung gestellt. Es kann sich hierbei um Anfragen an die Mitbenutzung eines Directories, eines Druckers oder einer anderen, vom Linux-Samba-Server verwalteten Ressource handeln. Die angeforderte Ressource wird dabei vom Client als ein Directory, CD-ROM-Laufwerk oder Drucker vom Typ Windows erkannt. Das bedeutet, ein für einen Windows-Rechner verfügbar gemachtes (im Microsoft-Jargon bedeutet das «freigegebenes») Directory auf der Linux-Seite verhält sich exakt wie ein Directory auf dem lokalen Rechner bzw. wie eines auf einem Windows NT-Server. Natürlich muss berücksichtigt werden, dass die Zugriffszeiten für ein Netzlaufwerk sehr viel länger sind als bei einer normalen lokalen Festplatte, aber das ist dann auch der einzige wirkliche Unterschied zwischen einem Netzlaufwerk und einem lokalen Festplattenlaufwerk.

Für jede `share-request`-Anforderung ist ein `smbd`-Daemon zuständig, deshalb muss der `smbd`-Daemon des Linux-Samba-Servers eine Kopie von sich selbst herstellen, sobald ein für ihn bestimmter Dienst angefordert wird. Auf diese Weise bekommt jede Anforderung eines Client einen für ihn unabhängig arbeitenden Daemon zugewiesen.

Mit der Zeile:

```
user@linux:~ > ps x | grep smbd
```

kann man herausfinden, wieviele Dienste von Clients beim Server angefordert wurden, dabei ist die Anzahl der Dienste immer $n - 1$, wenn n die Anzahl der laufenden `smbd`-Daemons ist. Wird kein Dienst angefordert, so existiert nur ein `smbd`-Daemon. Das ist der, von dem beim Anfordern eines Dienstes oder einer Share Kopien erzeugt werden, sozusagen der Urvater aller `smbd`-Daemons.

Falls die Funktion des Kommandos `ps` nicht bekannt ist, so sei auf die *Einführung in Unix*, im Anhang A auf Seite 837 verwiesen.

□ *nmbd*-Daemon:

Der *nmbd*-Daemon fungiert als sogenannter *Nameserver*, das bedeutet, er löst einen Namen, wie `olaf.delcarlo.italisa`, in eine IP-Nummer wie `192.168.17.1` auf. Das geschieht im wesentlichen nach den gleichen Prinzipien wie bei einem DNS-Server oder einer `/etc/hosts`-Datei. Von diesem Daemon existiert immer nur ein Exemplar.

Mit der Zeile:

```
user@linux:~ > ps x | grep nmbd ↵
```

kann man diesen Umstand überprüfen.

Falls mehr als ein *nmbd*-Daemon existiert, so muss man herausfinden, wo der zweite Daemon gestartet wird. Meistens ist das der Fall, wenn man die Samba-Installation von SuSE Linux verwendet und dann undifferenziert eine neue Samba-Version installiert, ohne sich um solche Dinge wie Startmechanismen zu kümmern. Im Absatz 20.1 auf Seite 759 kann nachgelesen werden, wie man das verhindert.

Beide Daemons von Samba verwenden eine zentrale Konfigurationsdatei, die Datei `/etc/samba/smb.conf`, um die entsprechenden Dienste bereitzustellen.

In dieser Datei können folgende Dienste eingerichtet und konfiguriert werden:

- Alle an Linux angeschlossenen Drucker können für Windows-Clients als proprietäre Netzwerkdrucker zur Verfügung gestellt werden, und zwar mit allen Vorteilen, die das «Filtering» von Unix-Druckern bietet (siehe Seite 236).
- Der Drucker-Zugriff kann für einzelne Benutzer, für Gruppen oder alle am Netz beteiligten Clients freigegeben werden.
- Linux-Directories können über das Netz von Windows-Clients verwendet werden.
- Zugriffe auf die freigegebenen Directories können an einzelne User, aber auch an ganze Workgroups vergeben werden.
- Es können spezielle Einstellungen bezüglich Sicherheit, Zugriff und auch Effizienz des Netzverkehrs eingestellt und verwaltet werden.

1.2.1 Was kann Samba, was kann es nicht!

Samba kann also generell als ein Platten- und Drucker-Server angesehen werden, der auf einer Unix-Plattform implementiert ist und der als ganz normaler SMB-Server in einem Windows- oder OS/2-Netz dienen kann. Samba kann Windows NT-Server ersetzen, sofern sie nur Platten und Drucker bereitstellen.

Ursprünglich ist die SMB-Technologie für kleine nicht-strukturierte Netze entwickelt worden, die in der Lage sein sollten, sich selbst zu administrieren. Die

Idee eines selbstadministrierten Netzes ist auf den ersten Blick recht attraktiv, bringt aber heute Probleme mit sich, an die man zu Beginn der Entwicklung nicht gedacht hatte. Dieser Umstand macht sich auch heute noch in der SMB-Netzstruktur bemerkbar. Die SMB-Technologie der Windows NT-Server lässt viele Wünsche offen. Insbesondere bei mehreren Sub-Netzen zeigt der Windows NT-Server ein manchmal merkwürdiges Verhalten. So gibt es nicht erscheinende Shares in nicht direkt erreichbaren Subnetzen, intolerabel lange Latenzzeiten, bis auf Veränderungen im Netz reagiert wird und die mangelhafte Freigabe von Ressourcen, wenn sie nicht mehr gebraucht werden. Auch in Microsoft-Kreisen kursiert die Meinung, dass ein Windows NT-Server oftmals das Verhalten eines Eichhörnchen kurz vor dem Winterschlaf hat. Die von Microsoft entwickelte Domain-Strukturen schaffen in vielen Fällen Abhilfe, aber sie führt auch immer weiter weg von einem selbstadministrierten Netz. Wer einem versierten Microsoft-Netzadministrator heute erzählt, dass das LAN-Netz der Microsoft-Welt selbstadministrierend ist, erntet nur ein nachsichtiges Lächeln. Sogar mittelgroße Firmen beschäftigen heute Vollzeit-Netzadministratoren, und die sind gewiss nicht unterbeschäftigt in dieser Position.

Aber auch Samba kann gewisse Dienste in einer Windows NT-Domain nicht bereitstellen. So kann ein Samba-Server zwar mittlerweile einen PDC (siehe Glossar  PDC) ersetzen, er kann aber die Aufgabe eines Backup Domain Controllers nur unter der Bedingung übernehmen, dass der Primary Domain Controller der Domain ebenfalls ein Samba-Server ist. Als ausgesprochener Fehler kann das aber nicht interpretiert werden, denn die Entwickler von Samba können nur Eigenschaften implementieren, deren Funktionsweise und Protokolle zu 100 % bekannt sind. Leider werden bestimmte Protokolle vom Hersteller und Vertreiber dieser Protokolle nicht freigegeben und können als Konsequenz auch nicht implementiert werden.

Aus dem gleichem Grund kann Samba auch nicht den Exchange-Server komplett ersetzen. Es fehlen die Protokolle! Der Mail-Server-Teil kann zwar problemlos durch Samba ersetzt werden, nicht aber andere Teile, wie zum Beispiel die Terminverwaltung des Exchange-Servers. Verteilte Datenbank-Anwendungen, die auf den Zielrechnern einen Windows NT-Server voraussetzen, werden ebenfalls nicht funktionieren.

Auch bei den Druckerdiensten kann Samba gewisse Drucker nicht einbinden. Es sind dies die Drucker, die das GDI-Protokoll verwenden. Wenn eine Windows NT-Workstation einen von einem Windows NT-Server freigegebenen Drucker verwenden möchte, so kann die Druckaufbereitung ganz vom Windows NT-Server übernommen werden, ohne dass der Samba-Server konsultiert wird. Zu diesem Zweck benötigt man keine Treiber-Installation auf der Server-Seite. Man kann dennoch die sogenannten Komfortfunktionen nutzen (zweiter Papierschlacht, Briefumschlag-Druck etc.), wenn man den freigegebenen Drucker des Windows NT-Servers verwendet.

Druckerwarteschlangen, die das EMF-Prinzip unterstützen, werden von Samba z. Zt. nicht bedient. Es sieht allerdings auch nicht so aus, als ob sich in absehbarer Zeit hier etwas ändern wird. Eine Implementierung dieser Eigenschaften auf der Samba-Seite verlangt eine Nachimplementierung des gesamten GDI-Interfaces. Der geringe Vorteil dieser Implementierung steht freilich in keinem Verhältnis zum Aufwand, der getrieben werden müsste, um auch diese Funktionalität zu implementieren. In der betrieblichen Praxis sind derartige Billig-Drucker sowieso nicht zu finden, daher ist das auch keine wirkliche Einschränkung des Samba-Servers.

Das alles mag einschränkend klingen, ist aber halb so schlimm. Der überwiegende Anteil der Netze, die in der Praxis anzutreffen sind, lässt sich mit einem Samba-Server bereichern, da die Hauptaufgabe einer Netzsoftware darin liegt, verteilte Shares den Netzteilnehmern zur Verfügung zu stellen und die zentrale Nutzung von Druckern und Backup-Devices zu ermöglichen.

Wenn es darum geht, wirklich große verteilte Datenbestände zu verwalten, dann werden diese Server in der Regel weder mit Linux noch mit Windows NT als Betriebssystem realisiert. Bei diesen Anwendungen spielen Betriebssysteme einer gänzlich anderen Liga eine Rolle.

1.2.2 Was kann Samba Version 3.0.9?

Wie Tabelle 1.1 zeigt, kann auch die neue Samba-Version nicht alle Funktionen eines echten Windows-Server erfüllen, aber die Aufgabenfülle ist doch sehr be-

Tabelle 1.1: Funktionen, die Version 3.0.9 übernehmen kann

Funktion	
File server	*
Printer server	*
Microsoft Dfs server	*
Primary domain controller	*
Backup domain controller	–
Active Directory domain controller	–
Windows 95/98/Me authentication	*
Windows NT/2000/XP authentication	*
Local master browser	*
Local backup browser	*
Domain master browser	*
Primary WINS server	*
Secondary WINS server	–

eindruckend und diejenigen Funktionen, die noch fehlen, werden nur in sehr wenigen und ausgewählten Netztopologien wirklich benötigt.

So kann die Funktion eines Secondary WINS Server ohne weiteres von einer Windows-Maschine übernommen werden und auch die Aufgabe eines Backup Domain Controller muss in einem komplexen Netz nicht unbedingt von einem Samba-Server übernommen werden. In Zukunft wird freilich die Funktion eines *Active Directory domain controllers* wichtig werden, so dass dies die einzige Funktion ist, die ein Samba-Server in Zukunft noch gerüstet werden sollte.

Vergleicht man die neuen Eigenschaften mit den älteren Versionen, so geht das Einsatzgebiet eines Samba-Servers in einer Microsoft-Umgebung weit über das einfache File-Serving hinaus. Außerdem ist das Einbinden und Konfigurieren eines Samba-Servers in einer solchen Umgebung wesentlich einfacher geworden, da fast alle Funktionen ohne großen manuellen Installationsaufwand zu nutzen sind.

1.3 Der Einsatz von Samba

Die einfachste und unkomplizierteste Methode, einen Samba-Server an ein bestehendes Netz zu bringen, funktioniert wie folgt:

- ❑ Man verbindet den Linux-Server mit einem Windows-Rechner. Das kann über eine einfache Koaxial-Verbindung geschehen oder aber auch über eine schnelle sogenannte *twisted pair*-Verbindung, bei der alle Netzteilnehmer über einen Hub oder Switch miteinander verbunden werden.
- ❑ Man vergewissert sich, dass man über eine `/etc/samba/smb.conf`-Datei verfügt und
- ❑ dass die beiden Daemons von Samba (`smbd`, `nmbd`) auf der Linux-Workstation laufen.

Abb. 1.1 auf Seite 5 zeigt eine einfache Installation eines Samba-kontrollierten Netzes. Ein Windows-Client kann Shares auf dem Samba-Server nutzen und gleichzeitig dient die Linux-Workstation als Druckerserver.

Hinweis:

Die alten 10MBit-Ethernet-Karten sollten in einer modernen Netzumgebung nicht mehr zum Einsatz kommen, sie sind einfach zu langsam. Aber auch Hubs sind nicht gerade eine optimale Lösung. Man sollte sogenannte Switchable Hubs verwenden, die bieten tatsächlich eine konstante Übertragungs-Bandbreite von 100MBit.

Die von SuSE Linux mitgelieferte Datei `/etc/samba/smb.conf` kann man ohne Probleme verwenden. Diese Datei ist syntaktisch korrekt, und man braucht die Datei daher nicht auf Korrektheit zu überprüfen. Das ist eine gute Ausgangsposition, denn häufigste Ursache für das Nicht-Funktionieren einer Samba-Installation ist das Fehlen der `/etc/samba/smb.conf`-Datei. Es kommt aber auch häufig vor, dass die Datei nicht gefunden wird oder dass sie syntaktisch nicht korrekt ist.

Wenn Probleme auftauchen, so kann man Hinweise darauf, wie sie zu beheben sind, in den beiden Log-Dateien finden, die sich unter `/var/log/samba` befinden (`log.nmbd` und `log.smbd`).

Wenn auch diese Konfiguration nicht viel nützen wird, so kann man doch einen Windows-Client an einen funktionierenden Samba-Server anbinden, und das sollte für das erste Erfolgserlebnis schon mal reichen. In den folgenden Abschnitten werden nun die einzelnen zur Einbindung eines Samba-Servers notwendigen Schritte etwas detaillierter beschrieben.

1.4 Verbinden mit dem Netz

Bevor wir irgend eine Netz-Software einsetzen können, und sei es auch nur `ping`, `ftp` oder `telnet`, muss die entsprechende Netzwerk-Software installiert sein. Manche Einzelplatzrechner werden mit einem einfachen `loopback`-Interface installiert, weil eine Vernetzung über eine Ethernet-Verbindung nicht geplant ist. In einem solchen Fall muss das Netz natürlich nachimplementiert werden,



Abbildung 1.2: Die Netzwerk-Grundkonfiguration

falls ein Samba-Server in Verbindung mit Clients eingesetzt werden soll (siehe Abb. 1.2 auf der vorherigen Seite). Das gilt natürlich auch für die Netzwerkkarten.

Auch wenn man nur über einen einzigen Linux-Rechner verfügt, kann die folgende Installation vorgenommen werden, denn auch mit nur einem Rechner, und völlig ohne Netzwerkkarte, kann ein Netz installiert werden. Das macht in der Regel nicht viel Sinn und ist auch nicht so spektakulär wie die Anbindung eines Windows-Rechners an den Samba-Server, denn auf die *Shares*, so nennt man die Directories, die von Samba zur Verfügung gestellt werden, hat man sowieso Zugriff, aber zum Ausprobieren reicht eben auch ein Rechner und das soll hier demonstriert werden.

Wir gehen jedoch davon aus, dass mindestens zwei Rechner zur Verfügung stehen, einen Linux-Rechner, der als Server fungiert, und einen Windows-Rechner, der den Client spielt.

Der bloße Einsatz von Linux als Server beeinträchtigt übrigens eine bereits bestehende Windows-Umgebung aber in keiner Weise, denn außer dass Linux die Datenhaltung übernimmt, beeinflusst der Linux-Server die Software-Infrastruktur einer Firma nicht. Die Mitarbeiter der Firma werden es nicht bemerken, dass die Datenhaltung auf einem Linux-Rechner stattfindet.

1.5 Ein erster Test

Zunächst überprüft man, ob im Rahmen einer bestehenden Installation Samba bereits auf dem eigenen System aktiv ist. Dazu gibt man folgende Befehle in einer Kommandozeile ein:

```
ps aux | grep smb  
ps aux | grep nmb
```

Falls Samba auf dem eigenen System läuft, sollte man eine Ausgabe ähnlich 1.1 auf dem `stdout`-Terminal beobachten können. `stdout` bedeutet übersetzt; *Standard Output* er stellt den Standard-Ausgabekanal dar, der auf einem Terminal-Fenster ausgegeben wird (siehe die Einführung in Unix, Anhang A auf Seite 837).

Ausgabe 1.1 Ist Samba aktiv?

```
olaf2@delcarlo:~ > ps aux | grep smb  
root 180 0.0 0.3 1764 452 ? S 06:39 0:00 /usr/sbin/smbd -D  
olaf2@delcarlo:~ > ps aux | grep nmb  
root 175 0.0 0.5 1416 716 ? S 10:01 0:00 /usr/sbin/nmbd -D
```

In diesem Fall kann man die folgenden Installationsschritte überspringen und direkt zu Abschnitt 1.8.1 auf Seite 21 übergehen.

Falls der Samba-Server nicht installiert ist, folgt jetzt die Installationsanweisung, so wie sie mit SuSE Linux durchgeführt werden kann.

1.6 Die erste Installation

Eine Standard-Installation des Samba-Servers gestaltet sich sehr einfach, besonders, wenn man die SUSE-Edition verwendet. Zur Zeit der Drucklegung des Buches wird in dieser Version nur die Version 3.0.7 unterstützt. Will man die neueste Version installieren (Version 3.0.9), so muss man sich diese aus dem Netz holen. Wie man das macht und die Version dann auf seinem Rechner installiert, wird in Kapitel 1.7 vorgestellt.

Für eine Standard-Installation mit SUSE LINUX benötigt man zuerst die Rechte des Superusers (siehe Abb. 1.3).



Abbildung 1.3: Für Yast benötigt man das root-Pasword!

Die werden abgefragt, wenn man YaST als ganz normaler Account öffnet. Dann wählt man die Softwareinstallation (siehe Abb. 1.4 auf der nächsten Seite). Und dann sollte man warten. Wenn man nicht gerade über einen AMD 64 Bit-Prozessor verfügt, dann kann das ganz schön lange dauern, also nicht ungeduldig werden.

In Abb. 1.5 auf der nächsten Seite sucht man dann nach allen Paketen, die für die Samba-Installation relevant sein können, und das sind mehr als man vermuten könnte. Seit der 3.0-Version gehören auch Pakete wie LDAP und eventuell ACL dazu. Zudem existieren einige smbclients, die ganz nützlich sein können.

Der Startmechanismus, der von den einzelnen Linux-Editionen verwendet wird, ist sehr unterschiedlich, deshalb muss von Fall zu Fall entschieden werden, wie vorzugehen ist. Für die SUSE-9.2-Edition existiert eine RPM-Version von der Samba-Version 3.0.7. Davon kann aber nicht immer ausgegangen werden. Für

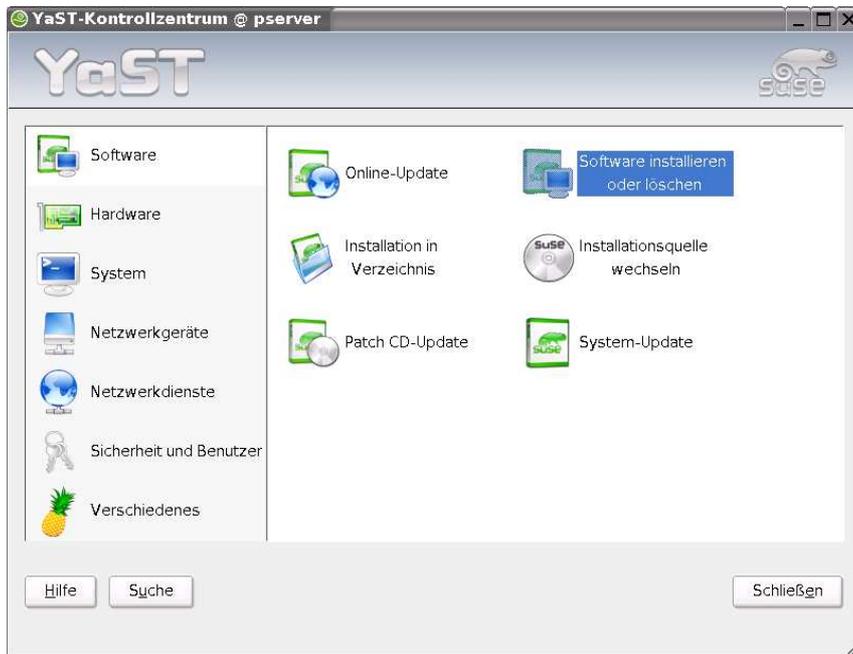
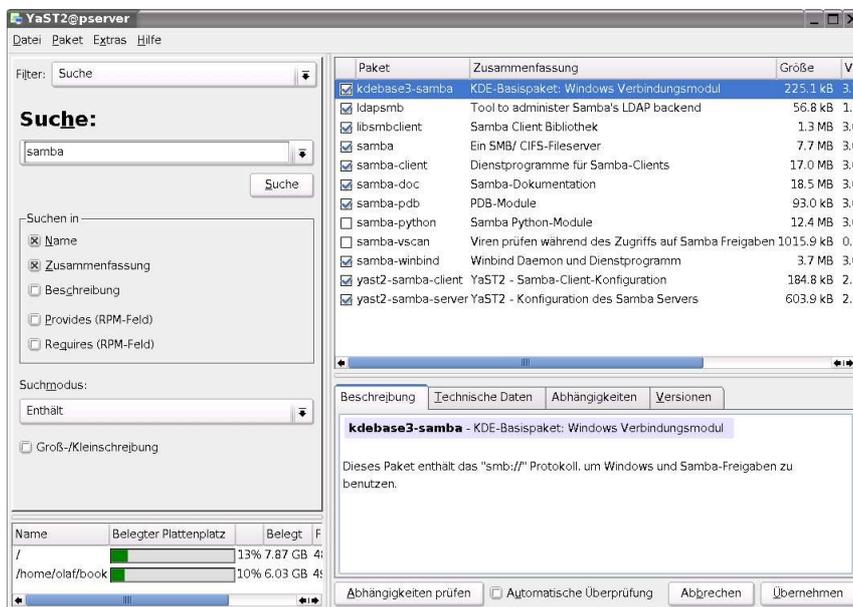
Abbildung 1.4: Hier sollte man *Software installieren oder Löschen* wählen.

Abbildung 1.5: Dann kann man hier die gewünschten Pakete anwählen.

den Fall, dass man **keine** RPM-Version findet, muss vorgegangen werden, wie in Kapitel 20 auf Seite 759 beschrieben.

Die SuSE-Edition sorgt dann dafür, dass die Pakete installiert und dass die entsprechenden Umgebungsvariablen eingestellt werden. Natürlich kann man auf diese Weise nur die etwas ältere Version 3.0.7 installieren, aber für die allermeisten Zwecke sollte das auch genügen. In Abschnitt 1.7 wird erklärt, wie man die neueste Version von Samba installiert, die man im Netz erhalten kann.

1.7 Wenn man kein SuSE Linux verwendet

Will man die neueste Version installieren (Version 3.0.9), steht man vor einem Problem, denn die Installationsmechanismen der verschiedenen Linux-Editionen haben sich in den letzten Jahren derart spezialisiert, dass ein Update so einfach nicht möglich ist. Fast alle Linux-Editionen verwenden heute den *Red Hat Program-Manager* und das bedeutet, dass ein zu installierendes Paket besser in einem solchen Format vorliegen sollte, will man nicht riskieren, dass das Betriebssystem **nach** der Installation eines Paketes ohne RPM-Format inkonsistent zurück bleibt.

Zunächst einmal müssen wir in diesem Fall die Frage klären, von wo man das Samba-Paket erhalten kann. Natürlich befindet sich die zum Zeitpunkt der Drucklegung aktuelle Version auf der beigelegten CD-ROM, aber sehr lange wird sie es nicht bleiben, daher sollte man schon wissen, von wo die neueste Version zu beziehen ist.

Die Beschaffung ist kein Problem; man findet entsprechende Links unter www.samba.org. Allerdings liegen gerade die neuesten Versionen des Samba-Paketes in den seltensten Fällen in einem RPM-Format vor. Meist findet man eine Source-Version, die für das Ziel-Betriebssystem erst angepasst und übersetzt werden muss.

Das Übersetzen funktioniert auch in den meisten Fällen ohne Probleme, danach stellt sich aber das Problem, wie man die neu übersetzte Version startet. Das ist nicht ganz trivial, denn ein einfaches Starten über `/etc/inetd.conf` ist bei gerade bei den neuen Versionen nicht ratsam und, will man alle Eigenschaften des neuen Servers nutzen, auch nicht möglich. Speziell der Daemon `nmbd` muss bereits beim Hochfahren des Systems gestartet werden und das legt nahe, dass man besser die vorgefertigten Startmechanismen der jeweiligen Linux-Variante verwendet, als einen eigenen zu stricken.

Den `nmbd`-Daemon erst zu starten, wenn das System bereits hochgefahren ist, ist nicht besonders klug, weil dieser Daemon im NetBIOS-Namensraum Namen für sich reservieren muss, die nach dem kompletten Hochfahren des Linux-Systems nicht mehr reklamiert werden können. Ansonsten hätten Windows-Rechner nämlich keine Möglichkeit den Samba-Server zu finden und das wä-

re gerade dann fatal, wenn dieser Server als Primary Domain Controller fungieren soll. Zudem kann sich der Daemon `nmbd` nicht selbst beenden, sobald er einmal gestartet ist. Für den Daemon `smbd`, das ist der Daemon, der die Share-Verwaltung übernimmt, gilt das nicht, der könnte schon über den Daemon `inetd` gestartet werden, aber es ist nicht sinnvoll, den aufwendigen Prozess des Superdaemons `inetd` anzuwerfen, der Gewinn in der beschleunigten Startzeit steht einfach nicht dafür.

In Kapitel 20 auf Seite 759 findet man weitere Informationen, die verschiedenen Installationsmöglichkeiten des Samba-Pakets betreffend.

1.8 Samba aktivieren

Wenn das Paket installiert ist, wird Samba zunächst nicht laufen (sollte Samba bei einer früheren Installation des Systems schon mitinstalliert worden sein, kann man auf Seite 13 sehen, wie man überprüft, ob Samba bereits aktiv ist).

Wenn der Samba-Server installiert, aber nicht aktiviert ist, muss man Linux übrigens nicht neu starten, um den Samba-Server zu starten! Das installierte Samba-Paket kann, wie jedes andere Netzwerkprogramm in einer Unix-Umgebung, auf der Kommandozeile manipuliert werden. Es ist ein Unding, dass gerade bei kritischer Serversoftware das Betriebssystem ständig neu gestartet werden muss, wenn eine Änderung der Konfiguration in Effekt treten soll. Das Neustarten von Komponenten während des Laufes ist einer der großen Vorzüge aller Unix-Betriebssysteme. Im Gegensatz zu anderen Betriebssystemen kann ein regelrechter Neustart praktisch immer vermieden werden.

Um sicherzugehen, dass der Samba-Server gestartet wird, kann man als *Superuser* mit den im Beispiel 1.1 gezeigten Kommandos starten, anhalten oder erneut starten. Bei dem Kommando `restart` wird der Samba-Server angehalten, falls er bereits läuft und dann wieder gestartet. Das macht man, um dem Server die Möglichkeit zu geben, die Konfigurations-Datei `/etc/samba/smb.conf` erneut einzulesen. Der Effekt ist aber derselbe, als würde man `stop` gefolgt von `start` eingeben.

Beispiel 1.1 Starten des Samba-Servers

```
/etc/init.d/smb start
/etc/init.d/smb stop
/etc/init.d/smb restart
```

Alternativ können die beiden Samba-Daemons auch mit

```
smbd -D
nmbd -D
```

gestartet werden. Zusätzlich existiert bei den meisten Linux-Editionen ein sogenannter *rc*-Befehl, mit dem der Samba-Server ebenfalls gestartet werden kann. Dazu gibt man die Befehle, wie in Beispiel 1.2 gezeigt, ein.

Beispiel 1.2 Starten des Samba-Servers mit *rc*-Befehl

```
r smb start
r smb stop
r smb restart
```

Natürlich kann man den Dienst wieder anhalten, indem man, wie unter Unix üblich, mit:

```
killall smbd
killall nmbd
```

die Daemons wieder anhält.

Als Systemadministrator sollte man wissen, wie man den einzelnen laufenden Prozessen Signale schickt. Wie wir gesehen haben, kann man den Prozessen mit `kill` numerische Signale senden, um sie zum Anhalten zu zwingen (`kill -9 <psid>`). Dieses Signal (9) besitzt als Alias den Namen `SIGKILL`. Das Signal `SIGHUP`, hat die niedrigste Signalnummer (1) und veranlasst nicht das Anhalten des Daemons, sondern das Signal veranlasst das Neueinlesen der entsprechenden Konfigurationsdatei, um die Daemons `smbd` und `nmbd` dazu zu zwingen, die veränderte `smb.conf`-Datei erneut zu interpretieren. Wie wir bereits auf Seite 7 gesehen haben, erzeugt Samba für jeden sogenannten *Share* einen neuen `smbd`-Prozess. Jeder dieser erzeugten Prozesse liest beim Start die `/etc/samba/smb.conf`-Datei ein und interpretiert sie.

Wenn bei laufenden `smbd`-Daemons die Konfigurationsdateien verändert wird und dann eine zweite Freigabe kontaktiert wird, so bedeutet das, dass zwei Shares aktiv sind, aber mit unterschiedlichen Konfigurationsdaten. Das Neueinlesen tangiert nämlich *nicht* bereits laufende Verbindungen. Die ursprüngliche `/etc/samba/smb.conf`-Datei, die für die erste Freigabe galt, bleibt für diese Freigabe gültig. Die gerade veränderte `/etc/samba/smb.conf`-Datei, die vom neu gestarteten Prozess eingelesen wurde, gilt nur für die neue Freigabe. Von der neuen `/etc/samba/smb.conf`-Datei haben aber alle bereits laufenden Prozesse überhaupt keine Kenntnis. Das ist ein Zustand, den man besser nicht toleriert. Schon allein deshalb nicht, weil die alte `/etc/samba/smb.conf`-Datei, wenn überhaupt, nur noch als Backup-Datei zur Verfügung steht. Das hängt vom jeweiligen Editor ab, nützt aber auch nicht viel, denn wenn man die Konfigurationsdatei zweimal sichert, wird die alte, für manche Prozesse noch gültige Konfigurationsdatei überschrieben. Man findet sich also in einer zweifelhaften Situation wieder, bei der die Randbedingungen bereits laufender Prozesse nicht mehr nachvollzogen werden können.

Diese Befehlsfolgen sind gerade im Experimentierstadium sehr wichtig, weil nur der Re- oder Neustart des Servers die zentrale Konfigurationsdatei `/etc/samba/smb.conf` neu einliest und deren Änderungen wirksam macht. Man braucht also nicht jedesmal den Rechner herunterzufahren und ihn dann neu zu starten, wenn man die Konfiguration von Samba verändern will. Das ist gerade für die Testphase, in der wir uns schließlich befinden, äußerst hilfreich, wie wir noch sehen werden.

Die beschriebene Art, den Samba-Server zu starten, dient nur für Testzwecke und ist natürlich nicht empfehlenswert für eine praktische Anwendung. Auf diese Weise konfiguriert, müsste man nach jedem Linux-Systemstart, den Server manuell erneut starten.

Um dem installierten Linux beizubringen, bei jedem Systemstart den Samba-Server neu zu starten, muss man nur im YaST-Menü *Samba Server* anwählen (Abb. 1.6), dann erhält man ein neues Fenster (Abb. 1.7 auf der nächsten Seite). Danach wird bei jedem Hochfahren der Linux-Workstation auch das Samba-System gestartet.

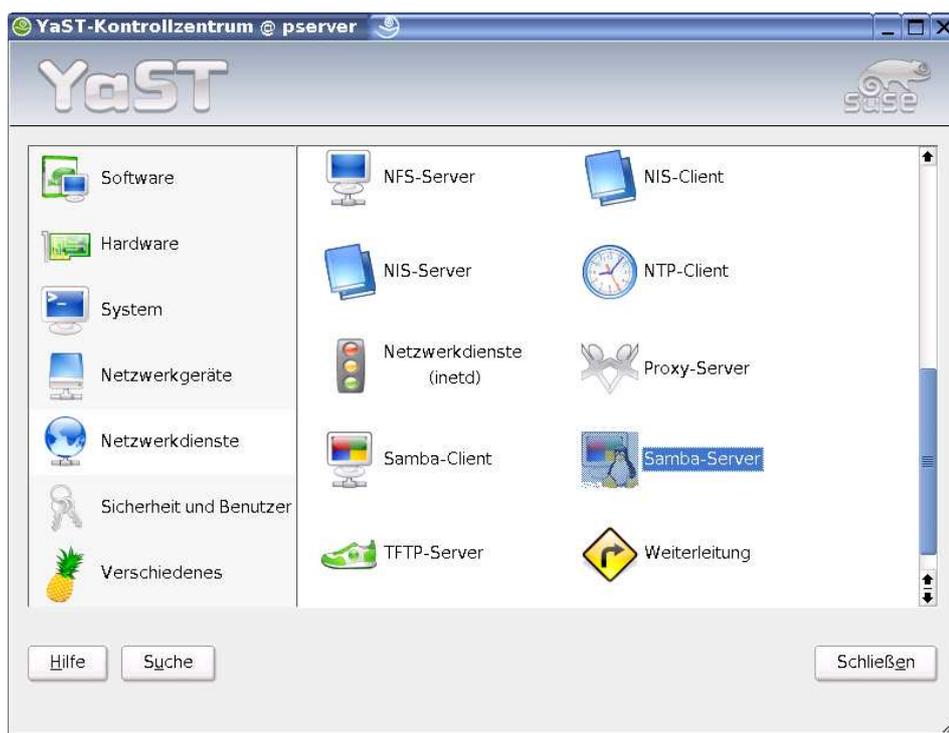


Abbildung 1.6: Samba Server

1 Die erste Installation

Wem das zu wenig transparent ist, der kann den Eintrag, wie in Beispiel 1.3 gezeigt, auch manuell mit einem Editor verändern. Die Datei befindet sich unter `/etc/rc.config`.

Beispiel 1.3 *Verändern der `/etc/rc.config`-Datei*

```
...  
#  
# start samba? ("yes" or "no")  
# Windows 95 / NT - File- and Printservices  
#  
START_SMB="yes"  
#  
...
```

Den Samba-Server dazu zu bringen, die `/etc/samba/smb.conf`-Datei neu einzulesen, kann auch mit dem Programm `swat` erzielt werden. Was sich hinter dem Programm `swat` verbirgt und wie man es einsetzt, wird erst im Absatz 18.3 auf Seite 671 genauer erläutert. Neben dem Programm `swat` werden mit dem Samba-Paket eine Fülle weiterer nützliche Programme mitgeliefert, die hier aber noch nicht besprochen werden sollen. Wer bereits jetzt schon etwas mit diesen Programmen herumexperimentieren möchte, findet die genaue Beschreibung der Programme im Kapitel 18 auf Seite 665 zusammengefasst.



Abbildung 1.7: Samba beim Neustart aktivieren

1.8.1 Einrichten des Servers

Bei dem in Beispiel 1.4 gezeigten Konfigurations-Code handelt es sich um den Inhalt der zentralen Konfigurationsdatei `/etc/samba/smb.conf`.

Diese Datei ist verantwortlich für das Verhalten des Samba-Servers im Netz. Die Datei kann sehr einfach strukturiert sein, sie kann aber auch extrem komplexe Konfigurationsdaten enthalten. Für den Anfang reicht freilich eine einfache Standard-Konfiguration aus. Beispiel 1.4 ist die unverändert übernommene `/etc/samba/smb.conf`-Datei von SuSE Linux. Unverändert, weil sie genauso auf dem eigenen Rechner implementiert ist, nachdem das Samba-Paket komplett installiert wurde. Das Beispiel sollte also von jedem Anwender sofort wiedergefunden werden können. Die `/etc/samba/smb.conf`-Datei befindet sich übrigens im Pfad `/etc`.

Für den Anfang scheint die Datei relativ komplex zu sein, aber das täuscht, da die meisten Eigenschaften auskommentiert sind. Kommentare werden entweder durch das `«;»`- oder `«#»`-Zeichen innerhalb der Zeile markiert. Jeder Text gilt als Kommentar, der sich nach einem dieser beiden Zeichen befindet. Das Kommentarzeichen gilt immer bis zum Ende der Zeile. Mehrere Zeilen an Kommentar müssen also jede für sich durch ein Kommentarzeichen am Anfang kenntlich gemacht werden.

Man sollte sich also nicht von den Kommentaren und der scheinbaren Komplexität des Beispiels 1.4 verunsichern lassen, und wir werden uns auch jetzt nicht um die Einzelheiten der `/etc/samba/smb.conf`-Datei kümmern. Bei dem folgenden Beispiel geht es nur darum, wie man den Samba-Server mit dieser Konfiguration *«zum Laufen»* bringt.

Beispiel 1.4 Die Standard-Installation von SuSE Linux

```
[global]
    workgroup = arbeitsgruppe
    guest account = nobody
    keep alive = 30
    os level = 2
    security = user

; Uncomment the following, if you want
; to use an existing NT-Server to
; authenticate users, but don't forget that
; you also have to create them locally!!!
; security = server
; password server = 192.168.1.10
; encrypt passwords = yes

printing = bsd
```

1 Die erste Installation

```
printcap name = /etc/printcap
load printers = yes

socket options = TCP_NODELAY

map to guest = Bad User

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

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

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

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

; Do you want samba to act as a logon-server for
; your windows 95/98 clients, so uncomment the
; following:
; logon script = %U.bat
; domain logons = yes
; domain master = yes
; [netlogon]
; path = /netlogon

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

; The following share gives all users access to the
; Server's CD drive, assuming it is mounted under
; /cd. To enable this share, please remove
```

```
; the semicolons before the lines
;
; [cdrom]
;   comment = Linux CD-ROM
;   path = /cd
;   read only = yes
;   locking = no

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

Die Erläuterung der einzelnen Einträge wollen wir uns an dieser Stelle ersparen. Die meisten der Parameterbelegungen werden auf Seite 82 ff. erläutert, dort steht dann das für das Verständnis nötige Hintergrundwissen zur Verfügung.

1.8.2 Clients auf Samba vorbereiten

Falls man in diesem Abschnitt auf Probleme auf der Client-Seite stoßen sollte, findet man im Kapitel 2.3 auf Seite 59 Hinweise zur Installation und Konfiguration diverser Clients aus der Microsoft-Welt. Dort werden auch ältere Client-Typen behandelt, wie OS/2-, Windows 95- und DOS-Clients.

Um Clients auf die Teilnahme an einem Samba-Netz vorzubereiten, müssen einige Grundlegende Begriffe geklärt werden, die häufig verwechselt werden.

1.8.3 Share – User

Aufgrund der besonderen Struktur der Windows-Betriebssysteme gilt es drei Begriffe genau zu unterscheiden:

□ Share-Name

Das ist der Name eines Directories, das exportiert werden soll. Dieser Name kann identisch mit dem User-Namen sein, muss es aber nicht! Dieses Directory wird auch als *Freigabe* bezeichnet. Eine Freigabe ist also eine Share, und eine Share ist wiederum ein Directory, das von einem Netzteilnehmer benutzt werden kann, als ob es sich auf seiner lokalen Maschine befände.

□ *User-Name*

Das ist der Name oder die User-ID eines Users, der eine Share für sich freigeben haben möchte. Eine Share auf einem Samba-Server kann den gleichen Namen tragen, wie der User. Dieser Umstand führt häufig dazu, dass die beiden Begriffe verwechselt oder gar gleichgesetzt werden.

□ *Password*

Das ist das Password, das einem User-Namen zugeordnet ist und das den Nachfrager nach einer Share berechtigt (oder auch nicht), die Share verwenden zu dürfen.

Besonders die Begriffe Share-Name und User-Name werden oft verwechselt oder gar als gleichbedeutend betrachtet. Die Abb. 1.10 auf Seite 36 erläutert bildlich diese Begriffe, in dieser Abbildung sieht man auch, dass das Pop-Up-Fenster auf dem NT-Server genau diese Eingaben verlangt. Der Share-Name und der User-Name können unter bestimmten Umständen identisch sein, trotzdem sind beide immer verschieden voneinander zu interpretieren. Etwas später, im Kapitel 5 auf Seite 183 werden wir weiteren Begriffen begegnen, die aber jetzt noch nicht von Bedeutung sind.

1.8.4 Samba mit nur einem Rechner

Wenn man nur über einen Linux-Rechner verfügt und dort den Samba-Server betreibt, kann man selbstverständlich auch Client-Dienste abfragen. Das Programm, welches man dafür verwendet, ist eigentlich dazu gedacht, sich als Linux-Client Ressourcen auf einer Windows NT-Maschine zu verschaffen, das Programm funktioniert auf *«jedem»* Server, also auch auf dem gerade eingerichteten. Wir werden diese Möglichkeit, einen Samba-Server zu benutzen, in dem Absatz 1.9 auf der nächsten Seite ausgiebig verwenden. Dort wird die grundlegende Funktionsweise eines Samba-Servers ohne Netz (und doppeltem Boden) an einem praktischen Beispiel erklärt.

Wenn man sich auf dem Samba-Server, wie in Beispiel 1.5 gezeigt, einloggt, so kann man auf dem Linux-Server gleichzeitig als Client und als Server in Erscheinung treten. Das ist besonders für Testzwecke äußerst hilfreich, wie wir gleich sehen werden.

Beispiel 1.5 Ein Client

```
olaf2@delcarlo:~ > smbclient //delcarlo/windir (↔)
Added interface ip=192.168.17.3 bcast=192.168.17.255 \
                    nmask=255.255.255.0
Password:
Domain=[DEVELOPMENT] OS=[Unix] Server=[Samba 2.2.1a]
smb: \>
```

Das Programm `smbclient` leistet auch dann sehr gute Dienste, wenn man einen Windows NT-Server im Netz betreibt. Dann kann nämlich auch ein Samba-Server Client auf einer Windows NT-Server-Maschine sein. Wenn man das Programm genauer kennenlernen will, so findet sich im Absatz 18.4 auf Seite 680 eine ausführliche Darstellung.

1.9 Ein einfaches Login

Für das nun folgende Beispiel eines einfachen Login benötigen wir weder einen weiteren Computer noch eine echte Netzinstallation. Ganz stimmt das nicht, denn ein zweiter Rechner wäre doch ganz hilfreich, aber er ist nicht zwingend notwendig.

Das Beispiel wird Login-Versuche unter verschiedenen Bedingungen ausführen und sollte ganz genau verstanden werden, um den Unterschied zwischen *Share* und *User* zu verstehen. Gerade diese beiden Begriffe werden häufig verwechselt, denn ein besonderer, aber sehr sinnvoller Mechanismus von Samba lässt anfänglich leicht Missverständnisse aufkommen. Es ist die Grundlage für jedes weitere Verständnis der Funktionsweise eines LAN-Netzes und eines Samba-Servers. Für das folgende Beispiel werden wir keine GUI verwenden, die graphischen Oberflächen sind zwar schön anzusehen, sie verdecken aber die Funktion unseres Beispiels unnötig und tragen nichts zum Verständnis bei.

Wir werden uns zwei verschiedene Einstellungen des Parameters `security` ansehen und versuchen, auf verschiedene Weise eine Freigabe auf dem Samba-Server zu erreichen. Der Parameter `security` ist entscheidend für das Verhalten von Samba, wenn man sich mit einem Server verbinden will. Der folgende Test bedarf eigentlich keines zusätzlichen Clients (obwohl das an einer Stelle des Beispiels doch von Nutzen wäre), wir werden einfach einen Client auf dem Samba-Server selbst installieren und über das Programm `smbclient` das Verhalten des Servers untersuchen.

Die Ausgangssituation:

Der Samba-Server ist auf einem Linux-Rechner mit dem *Host-Namen* **borkner** installiert. Auf diesem befindet sich ein *Account* mit Namen **olaf** und ein weiterer mit dem Namen **franca**.

Wir haben uns auf dem Rechner `borkner` bereits als User `olaf` angemeldet.

Die `/etc/samba/smb.conf`-Datei des folgenden Tests ist sehr einfach strukturiert, und zwar so wie Beispiel 1.6 auf der nächsten Seite sie zeigt.

In der `/etc/samba/smb.conf`-Datei sind nur zwei *Shares* auf dem Samba-Server definiert, von denen die `[printers]`-Section für dieses Beispiel nicht inter-

Kapitel 12

LDAP und Samba

Seit es Rechnernetze gibt, die mehrere Computer miteinander verbinden, haben die Dienste und Anforderungen an solche Netze drastisch zugenommen. Während zu Beginn der 80er Jahre des letzten Jahrhunderts, einige wenige gleichberechtigte Rechner in einem geschlossenen Netz miteinander Daten austauschten, wird heute der Begriff *Netz* viel weiter gefasst. Auf Grund der riesigen Datenmengen und der offenen Strukturen von Netzen müssen heute ganz andere Anforderungen an eine funktionierende Infrastruktur gestellt werden.

Eine Vielzahl von Diensten, die alle unabhängig voneinander konfiguriert und gewartet werden müssen, bestimmen die heute gebräuchlichen Netze, und es bedarf schon eines tiefen Fachwissens, um diese Dienste alle miteinander zu verbinden und zu koordinieren.

Um alle diese Anforderungen unter einen Hut zu bringen, wurden die Verzeichnisdienste entwickelt. Sie sollen es möglich machen, die Verwaltung von Netzdiensten zu vereinheitlichen und an eine Stelle zu delegieren.

12.0.1 Was sind Verzeichnisdienste?

Unter dem Begriff *Verzeichnisdienst* versteht man eine zentrale, oder auch verteilte Datenbasis, basierend auf dem bekannten *Client/Server*-Prinzip. Diese Datenbasis kann eine fast beliebige Informationsmenge verwalten; und durch den Einsatz von Klartextdateien können die verschiedenen Dienste konfiguriert und administriert werden.

12.1 Warum sollte man LDAP einsetzen?

Die gebräuchlichen Dienste, wie E-Mail, DNS und Samba verfügen natürlich schon von sich aus über derartige Konfigurations- und Administrationsmöglichkeiten. Der Vorteil, den LDAP bietet ist: die Verwaltung und Konfiguration kann

zentral von einer Instanz aus kontrolliert werden, obwohl die Dienste selbst dezentral eingerichtet sein können. Und letztlich sind es auch die immer stringenter werdenden Sicherheitsregeln, die eine zentrale Instanz für die Administration notwendig machen.

Das LDAP-Paket gehört nicht eigentlich zu Samba, allerdings kann ein Samba-Server nur dann alle Aufgaben in einer Windows-Domain übernehmen, wenn er in der Lage ist, sämtliche Attribute, die von Windows unterstützt werden, ebenfalls zu berücksichtigen. Und dies wiederum funktioniert nur in Verbindung mit dem LDAP-Paket.

Um das Zusammenspiel von Samba mit LDAP zu demonstrieren, ist dieses Kapitel der vierten Auflage hinzugefügt worden. Das kann aber nicht bedeuten, dass das LDAP-Paket eine umfassende Behandlung erfährt, denn das würde den Umfang eines Werkes über Samba nur unnötig aufblähen, deshalb beschränkt sich die folgende Darstellung auf die Aspekte von LDAP, soweit sie Samba betreffen.

Der Einsatz von LDAP erscheint auf den ersten Blick nur wenig sinnvoll zu sein, besonders dann, wenn man zu Hause nur über einen oder zwei Rechner verfügt, und die Anzahl der Accounts sehr beschränkt ist. Um den Sinn, der hinter LDAP steckt, einzusehen, muss man sich vor Augen führen, dass in einem großen Netz sehr viele Benutzer mit den unterschiedlichsten Profilen verwaltet werden müssen. Jeder Administrator eines auch nur mittelgroßen Netzes wird mir zustimmen, dass es extrem kompliziert sein kann, auf die Wünsche und Sonderbedingungen der einzelnen Accounts einzugehen. Und häufig kommt es sogar vor, dass bestimmte Sonderwünsche überhaupt nicht realisiert werden können, weil das System, sei es nun Windows oder Linux, das Geforderte einfach nicht möglich macht. Zudem ist das Informationsbedürfnis in einer vernetzten Welt sehr stark angestiegen. Wenn Clients mit den unterschiedlichsten Servern Daten austauschen wollen, so muss die Authentifizierung sichergestellt sein; und dazu gehört nicht nur ein UserID und ein Passwort.

Es müssen die unterschiedlichsten Sicherheitskriterien beachtet werden, verschiedene Verschlüsselungsmechanismen und die Gruppenzugehörigkeit sind nach einem einfachen *Ja/Nein*-Prinzip nicht mehr darzustellen. Zudem werden viele Dienste, die früher lokal verwaltet werden, von sogenannten Agents übernommen. Als Beispiel soll hier die Überwachung der Passwords dienen. Kein wirklich großes Netz überlässt die Authentifizierung den einzelnen Teilnehmern, sondern dieser Dienst wird von einem speziell dafür vorgesehenen Server übernommen.

Einen Samba-Server als Primary Domain Controller einzusetzen, ist prinzipiell kein großes Problem, das wird in Kapitel 9.1 auf Seite 337 ausführlich erklärt. Allerdings waren die Möglichkeiten eines solchen Servers bis zur Version 2.2 doch noch sehr eingeschränkt. So konnte die Gruppenzugehörigkeit eines Clients nicht an *die* Workstation weitergegeben werden, bei der sich der Client anmelden wollte. Die kompletten Profil-Daten, die ein Windows NT-Domain-Controller eines

einfachen Windows-Clients können nämlich in der `smbpasswd` nicht untergebracht werden. Das Format der `smbpasswd` lässt dies einfach nicht zu. Die Lösung, die sich bis zu der genannten Version anbot, war die Angabe eines globalen Pfades (siehe Referenz `logon home` auf Seite 572) für die Profildaten aller Clients. Das war aber nur eine provisorische Lösung, denn dieser globale Pfad erlaubt nicht die individuelle Verwaltung von Profildaten, für jeden einzelnen Client.

Um dieses Problem zu umgehen, müsste man den Ort der Profildaten eines Clients separat spezifizieren können. Seit der Samba-Version 2.2.3a ist es möglich, die Datei `smbpasswd` in einem LDAP-Verzeichnis abzulegen, in dem sich auch sämtliche Profildaten der unterschiedlichen Clients befinden können. Dadurch ist es möglich, einen Samba-Server als eigenständigen Primary Domain Controller zu konfigurieren und zwar so, dass auch individuelle Profildaten weitergegeben werden können.

12.2 Was bedeutet LDAP?

Ausgeschrieben bedeutet LDAP *Lightweight Directory Access Protocol*. Damit wird auch schon beschrieben, was LDAP eigentlich macht. Es ist ein Protokoll, das dem Zugriff auf eine Datenbank dient.

Das alltägliche Arbeiten mit einem Computer erfordert es, dass anfallende Daten strukturiert und manipuliert werden können. Diese Aufgaben übernimmt bei allen bekannten Betriebssystemen die Struktur des Directories. Ein Directory ist eine der einfachsten bekannten Datenbankstrukturen, welches von der ITU¹ im Rahmen der OSI-Protokolle definiert worden sind. Nach dieser Definition kann eine Directory-Struktur als ein Konglomerat von Einträgen definiert werden, welche die unterschiedlichsten Informationen enthalten können. So können beispielsweise Informationen über Personen gespeichert werden, aber auch solche, über verwendete Drucker, Gruppenzugehörigkeit. Diese Art eines Directories geht natürlich weit über die gewohnte Struktur eines Directories hinaus. Man kann sich einen LDAP-Eintrag als Eintrag in einem normalen Telefonbuch vorstellen, mit dem Unterschied, dass die Einträge in einer LDAP-Datei wesentlich komplexer sein und auch viel mehr Einträge enthalten können.

Eine eindeutige Zuordnung zwischen Eintrag und Name ist in einer modernen Directory-Struktur nur in den seltensten Fällen möglich. Gerade in der heutigen Zeit der intensiven Vernetzung spielen Sicherheitskriterien eine herausragende Rolle. So muss es möglich sein, die unterschiedlichsten Arten der Verschlüsselung pro Datei und pro Nutzer zu definieren. Aber auch die Attribute der Zugriffsberechtigten sind in einer netzorientierten Umgebung wesentlich komplexer, als das in einer Umgebung mit unnetzten Einzelrechnern war. So kann

¹International Telecommunication Union

beispielsweise ein einziger Zugriffsberechtigter über mehrere E-Mail-Adressen verfügen. Aber auch die Art und *Zugriffsberechtigung*, über die ein Client verfügen kann, unterscheiden sich in einer eher modernen Directory-Struktur.

Alle diese Gesichtspunkte lassen es nützlich erscheinen, den Begriff Directory unter vier verschiedenen Gesichtspunkten zu betrachten.

12.2.1 Sicherheitsmodell

Um ein Directory betreten zu können, muss man über die Rechte hierzu verfügen. Dabei werden verschiedene Arten der Rechtevergabe unterschieden, die wir uns im folgenden ansehen wollen.

Keine Authentifizierung

Es existieren verschiedene Gründe, warum Daten auch ohne Authentifizierung jedermann in einer Domain zugänglich sein sollten. Wenn beispielsweise eine Firma ein Adressbuch unterhält, so sollte dieses aus naheliegenden Gründen für jeden Netzteilnehmer ohne besonderen Aufwand einsehbar sein. Das beinhaltet natürlich auch Suchoperationen und eventuell Leseoperationen, um Netzteilnehmer und Daten über diese finden zu können. Einen solchen Zugang ermöglicht das Netz, indem es auf eine `bind`-Operation ohne Benutzernamen reagiert.

User/Password

Die gebräuchlichste Methode sich in einer Domain zu authentifizieren, ist die *Basic Authentication Method*. Bei diesem Verfahren wird während der `bind`-Operation ein eindeutiger Name und das dazugehörige Password übergeben. Der LDAP-Server sucht daraufhin in dem zugehörigen Verzeichnis nach dem mit DN (s. Abschnitt 12.2.2) bezeichneten Objekt. Dann vergleicht LDAP das Password mit dem Attribut `userPassword`, und dem User wird der Zugriff gestattet. Dieses Verfahren sollte *nicht* ohne eine verschlüsselte Datenverbindung zwischen Client und Server angewandt werden.

Simple Authentication and Security Layer

Betrachtet man das Challenge-Response-Verfahren, so erkennt man, dass für eine sichere Verbindung zwischen zwei Clients ein normales Authentifizierungsverfahren manchmal nicht ausreicht, um den Server von der Zugriffsberechtigung eines Clients zu überzeugen. Hierfür setzte man den sogenannten *Simple Authentication and Security Layer* ein. Allerdings beherrscht auch Samba-Version 3.0 nur die einfache Authentifizierung, daher sollte LDAP und Samba auf der gleichen Maschine installiert werden. Man kann allerdings auch zwischen der Samba-Maschine und der Maschine, auf der LDAP installiert ist, eine *SSL*-verschlüsselte Verbindung einrichten.

12.2.2 Namensmodell

Alle Einträge in einem Directory werden mit dem `distinguished name` gekennzeichnet. Dieser Name wird mit `DN` bezeichnet. Ein solcher Name kann auch als `full qualified Name` aufgefasst werden, das bedeutet, das Directory wird mit dem kompletten Pfad und seinem Namen identifiziert.

12.2.3 Informationsmodell

Der Inhalt traditioneller Directories, so wie sie seit langem üblich sind, zeichnete sich über eine lange Zeit durch einen Mangel an Informationsmöglichkeiten aus. Anfangs war es nur der Name, der einen Eintrag kennzeichnen konnte, und der war bei dem gebräuchlichsten Betriebssystem zusätzlich auf eine Anzahl von acht plus einer sogenannten Extension von drei Buchstaben beschränkt. Diese Einschränkung musste sein, da die Rechner der DOS-Zeit nur über sehr wenig Speicher verfügten. Jedes Zeichen zählte. Das war jedoch keinesfalls eine Limitierung, der nur einfache Betriebssysteme wie DOS oder CPM unterworfen waren, auch Unix hatte noch bis 1982 mit akutem Speichermangel zu leben. Die Kommandonamen wie `ls`, `cp` oder `mv` lassen das heute noch erkennen.

In einer nicht vernetzten Single-User-Umgebung war eine solcherart dürftige Bezeichnung sicherlich auch ausreichend, wenn freilich in einer vernetzten Umgebung auf den Inhalt eines Directories von *verschiedenen* Benutzern zugegriffen werden kann, so müssen wesentlich komplexere Strukturen zur Kennzeichnung eines Eintrags herangezogen werden. Das LDAP-Paket bietet hier sehr komplexe und aussagekräftige Möglichkeiten.

Ein Directory enthält heute Objekte, die einzeln Benutzern zugeordnet werden können. Diese Benutzerzuordnung bezeichnet man als Attribut. Dabei kann ein solches Objekt mit mehr als einem Attribut ausgestattet sein. So ist es zum Beispiel möglich, einem Berechtigten mehr als eine E-Mail zuzuweisen. Gegenüber dem Betriebssystem Unix freilich muss natürlich ein Eintrag eineindeutig identifizierbar sein. Das wird durch die Zuordnung einer UID gewährleistet, die aber bereits an anderer Stelle erläutert wurde.

Ganz konkret muss jeder Directory-Eintrag ein Attribut vom Typ `objectClass` besitzen. Eine solche Klasse definiert, welche Attribute ein Eintrag haben muss und welche er optional noch besitzen darf. Wenn, was in der Praxis häufig vorkommt, ein Eintrag mehreren Benutzern zugeordnet werden muss, so sollte klar sein, dass es dann auch mehr als ein Attribut vom Typ `objectClass` geben kann. Das bedeutet, wenn ein Eintrag mehreren Klassen angehören darf, so müssen auch mehrere Attribute vom Typ `objectClass` erklärt werden. Als Beispiel soll hier der Pfad des Benutzerprofils dienen. Das Benutzerprofil eines Accounts wird bei Unix *nicht* in der Definition des Accounts abgelegt, ist daher dem Objekt auch nicht kenntlich. Die Pfade eines solchen Accounts (`/home/franz` bei-

spielsweise) werden von Samba aber benötigt. Sie können in der Beschreibung des LDAP-Eintrags abgelegt werden. Wir werden später noch genau sehen, wie das zu geschehen hat.

Alle Einträge in den verschiedenen Directories müssen über eine Menge von Objektklassen verfügen, diese Zuordnungen (Objektklasse ↔ Attribute) werden in einem sogenannten *Schema* festgelegt, das nicht Teil der Verzeichnisdaten ist.

12.2.4 Funktionsmodell

Das Funktionsmodell beschreibt allgemein die Art des Zugriffs auf ein Directory. Dabei handelt es sich natürlich auch um die üblichen und gewohnten Funktionen, die auf einem Directory ausgeführt werden können, als da sind *Betreten*, *Suchen* und *Ändern*. LDAP bestimmt, welche Operationen einem Client zugänglich sind und welche nicht.

Aber bereits hier muss unterschieden werden:

12.2.4.1 Betreten

Diese Operation muss im Kontext des Sicherheitsmodells gesehen werden, denn LDAP überprüft nach bestimmten Kriterien die Zugangserlaubnis und die Art der Authentifizierung (siehe 12.2.1 auf Seite 386).

12.2.4.2 Suchen

Sehen wir uns als Beispiel das *Suchen* in einem Directory an. Beim Eintritt in ein Directory kann innerhalb der sichtbaren Einträge gesucht werden, es kann aber auch innerhalb der sichtbaren *und* der hierarchisch untergeordneten Bereiche gesucht werden. Dieses Verhalten kann man in der Konfigurationsdatei für LDAP festlegen.

12.2.4.3 Ändern

Die Operation *Ändern* erlaubt dem Benutzer, Einträge in ein Directory hinzuzufügen, zu löschen und zu verändern.

12.3 Installation und Einrichten

Die SuSE 9.2-Version von LDAP und hier speziell die Einrichtung ist zwar recht bunt, leider aber nicht besonders durchsichtig. Die Installation eines Servers ist über YaST überhaupt nicht möglich. Aus diesem Grund wird hier die Installation und die Inbetriebnahme eines LDAP-Servers beschrieben.

Leider ist das Einrichten in den allermeisten Dokumentationen nur sehr mangelhaft dargestellt, und das liegt zum Großteil an der verwendeten Terminologie. Sie ist, man kann durchgängig sagen, nicht transparent und speziell in einem lokalen Netz nicht nachvollziehbar dargestellt. Das gilt leider auch für die offizielle Dokumentation der Page www.openldap.org, die doch von den besten Experten auf diesem Gebiet unterhalten wird. Deshalb wird im Folgenden die Installation und das Einrichten eines LDAP-Servers kurz dargestellt.

12.3.1 Wo bekommt man die neueste Version?

Am einfachsten bezieht man die neueste Version unter folgender Adresse:

```
http://www.openldap.org/software/download
```

Dort finden sich einige sogenannte Mirrors, von denen man die neueste Version herunterladen kann. Welche Adresse am besten funktioniert, ist von Fall zu Fall verschieden. Im nun geschilderten Beispiel ließ sich die Version von der TU-Wien am schnellsten herunterladen. Man muss hierzu nicht die entsprechende Site kontaktieren, es genügt, wenn man das letzte Feld der Tabelle auswählt, dann wird automatisch das Herunterladen eingeleitet.

Diese erhaltene Version packt man dann aus und zwar mit dem folgenden Kommando:

```
gunzip -c openldap-stable-20040923.tgz | tar xvfB -
```

Dann wechselt man in das entstandenen Directory:

```
cd openldap-2.2.17
```

12.3.2 Übersetzen des Pakets

Nun muss man die Version übersetzen.

Um ein komplexes Programmpaket zu übersetzen, bedienen sich heute fast alle Unix- und Linux-Derivate der `./configure`-Datei. Diese Datei steuert einen sogenannten Preprocessor, der für den eigentlichen Übersetzungsvorgang eine `Makefile` produziert.

Die Plattformen und Anforderungen, auf der ein solcher Server laufen soll, sind recht unterschiedlich, deshalb gibt es die Möglichkeit, das spätere Übersetzen des Programmpakets an die eigenen, lokalen Bedürfnisse anzupassen. Welche Anpassungsmöglichkeiten möglich sind, erhält man, wenn man das Kommando:

```
./configure --help
```

eingibt.

Der folgende Auszug zeigt zur Demonstration nur die ersten und letzten Optionen:

```
Copyright 1998-2004 The OpenLDAP Foundation. All ...
    Restrictions apply, see COPYRIGHT and ...
Usage: configure [options] [host]
Options: [defaults in brackets after descriptions]
Configuration:
    --cache-file=FILE          cache test results in FILE
    --help                    print this message
    ...
    ...
Library Generation & Linking Options
    --enable-static[=PKGS]    build static libraries ...
    --enable-shared[=PKGS]   build shared libraries ...
    --enable-fast-install[=PKGS] optimize for fast...
    --with-gnu-ld             assume the C compiler u...
    --disable-libtool-lock    avoid locking (might br...
    --with-pic                try to use only PIC/non...

See INSTALL file for further details.
```

Im Regelfall sollte keine der vorbelegten Optionen verändert werden müssen. Nur in seltenen Ausnahmefällen kann die verwendete Datenbasis (hier Berkeley DB) unterschiedlich sein. Prinzipiell kann man natürlich jede Datenbasis verwenden, aber es ist einfacher, die geforderte Datenbasis zu installieren, anstatt LDAP mit einer anderen Datenbasis zu installieren.

Nun kann man daran gehen, mit dem Kommando `./configure` die Preprocessor-Datei `Makefile` zu erzeugen. Das ist ein langwieriger Prozess, sollte aber in der Regel anstandslos beendet werden können.

Mit den Kommandos:

```
make depend
make
```

kann die Software übersetzt werden. Nachdem `./configure` problemlos durchlaufen wurde, die nötigen Komponenten also vorhanden sind, sollte es hier keine Probleme geben.

```
make test
```

Wenn das erfolgreich durchlaufen wurde, kann man die übersetzte Version im System installieren:

```
su root -c 'make install'
```

Wie man sieht, geht das nur mit `root`-Rechten.

Nun muss die Konfigurations-Datei `/etc/openldap/slapd.conf` editiert werden:

```
database bdb
suffix "dc=<eigene-DOMAIN>,dc=<COM>"
rootdn "cn=Manager,dc=<eigene-DOMAIN>,dc=<COM>"
rootpw secret
directory /var/lib/ldap
```

Der Meta-Parameter `<eigene-DOMAIN>` muss dabei selbstverständlich durch die eigene Domain ersetzt werden (oder man benennt die eigene Domain in `eigene-Domain` um, was natürlich keine besonders schöne Lösung ist.)

Das angegebene Directory muss natürlich bereits existieren, wenn nicht, so muss man es vor der Inbetriebnahme erzeugen.

Nun kann man die Server-Software starten und das geschieht wie folgt:

```
rclldap start
```

Um zu prüfen, ob der Server funktioniert, gibt man folgendes Kommando ein (im Beispiel ist als Domain der Name `writers` verwendet worden):

```
ldapsearch -x -b "" -s base '(objectclass=*)' namingContexts
```

```
dn:
namingContexts: dc=writers,dc=com

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

Nun muss eine Beispiel-LDIF-Datei erzeugt werden. Am Besten macht man dies in dem angegebenen Directory `/var/lib/ldap` und nennt die Datei `beispiel.ldif`

```
dn: dc=writers,dc=<COM>
objectclass: dcObject
objectclass: organization
o: Writers
dc: writers

dn: cn=Manager,dc=writers,dc=<COM>
objectclass: organizationalRole
cn: Manager
```

Nun kann ein neuer Eintrag im Directory erzeugt werden und zwar mit:

```
-x -D "cn=Manager,dc=writers,dc=<COM>" -  
      / W -f beispiel.ldif
```

Mit:

```
ldapsearch -x -b 'dc=writers,dc=com' '(objectclass=*)'
```

kann man sehen, ob das funktioniert hat.

Damit hat man einen funktionierenden Server aufgebaut, dessen Syntax freilich ursächlich nichts mehr mit Samba zu tun hat und konsequenter Weise auch nicht mehr Gegenstand dieses Buches ist.

Was jetzt folgt, ist die Beschreibung, wie bestehende Daten in ein neu erzeugtes LDAP-Directory überführt werden können.

12.4 LDIF

Wenn ein LDAP-Directory eingerichtet werden soll, so wird es im Regelfall notwendig sein, bereits vorhandene Daten zu übernehmen. Hierzu dient das LDAP-Data-Interchange Format (LDIF). Der Inhalt einer LDIF-Datei besteht aus einer Liste von Zeicheneinträgen, wobei jeder Eintrag mindestens über ein Attribut vom Typ `objectClass` verfügen muss.

Die einfachste Form eines solchen Eintrags ist in Beispiel 12.1 gezeigt.

Beispiel 12.1 *Inhalt einer LDIF-Datei*

```
# comment  
dn: ou=Students,dc=sci,dc=univr,dc=it  
ou: Students  
objectclass: organisation  
objectclass: organizationalUnit  
o: Entwicklung  
dc: samba
```

Die Datei dient dem Transfer alter Directory-Strukturen in eine neue, LDAP-kontrollierte Umgebung. Deshalb wird eine solche Datei in fast jedem Fall notwendig sein (außer man setzt ein komplett neues Betriebssystem auf, welches sofort unter der Kontrolle von LDAP arbeitet).

Da der Inhalt dieser Datei sehr stark von der individuellen Struktur der Betriebssystem-Umgebung abhängt, kann man auch keine generelle Inhaltsangabe machen. Das Beste ist; man experimentiert etwas und sieht sich dann die entsprechenden Log-Dateien an. Dort findet man Hinweise darauf, welche Strukturen erkannt wurden und welche nicht.

Teil III

Samba-Referenz

Dieser Teil des Buches gibt eine komplette Übersicht aller Parameter, Variablen und Voreinstellungen des Samba-Pakets der Version Version 3.0.9. Er eignet sich «nicht» als Installations- und Konfigurationsanweisung für allgemeine Einstellungen der Datei `smb.conf`. Für diesen Zweck sollte der Teil II konsultiert werden. Aber auch Teil I kann hier gute Dienste leisten, freilich nur für einfache Netztopologien. Trotzdem erschöpft sich die folgende Referenz nicht nur in simplen Auflistungen der Voreinstellungen und spartanischen Erläuterungen. Dieser Teil III dient auch dazu, bestimmte Parameter hervorzuheben und zu beschreiben.

Es folgt nun der Referenzteil, der als Kompendium aller Parameter und vordefinierter Sections dient. Hier kann die Bedeutung der einzelnen Parameter nachgeschlagen werden. Meist finden sich dort auch kleine Beispiele oder zumindest Ausschnitte von Beispielen, die für die eigene Konfiguration brauchbar sind. Selbstverständlich stützt sich die Beschreibung der einzelnen Parameter auf die Informationen, die in den `man pages` zur Datei `/etc/samba/smb.conf` zu finden sind. Wie sollte das auch anders sein, schließlich sind diese Informationen die offiziellen, von der Gruppe der Samba-Entwickler herausgegebenen Erläuterungen. Auch wenn mancher meiner Leser der Meinung sein wird, man könne diese Informationen auch dort nachschlagen, so will ich hier kurz darlegen, warum ich diese Informationen als eigenständigen Teil III dieses Buches für sinnvoll halte:

- ❑ Es ist wenig praktikabel, bei der Lektüre eines Buches die Referenzen «online» zur Verfügung zu halten, noch dazu, wenn man ständig zwischen zwei Sprachen hin- und herschalten muss. Zudem kann nicht davon ausgegangen werden, dass die englische Sprache jedem geläufig ist, der sich ein Buch über Samba kauft.
- ❑ Viele der Parameterbeschreibungen in den `man pages` sind sehr spartanisch gehalten und vor allem ohne umgebenden Kontext kaum zu verstehen. Eine ausführlichere Erläuterung und vor allem Referenzierung ist dann sinnvoll und angebracht, besonders, wenn sie den Kontext des Parameters berücksichtigt.
Ich hätte die entsprechenden Parameter in ihrer natürlichen Umgebung beschreiben können, also im Kontext der Beispiele. Aber dann hätte die Parameterbeschreibung nicht komplett sein können, da viele Parameter für die Beispiele gar nicht benötigt werden. Außerdem käme man trotzdem ohne Verweise nicht aus, hätte man eine Doppelbeschreibung vermeiden wollen. Ich habe mich daher entschlossen, die Beschreibung der Parameter zentral und vor allem komplett zu halten.
- ❑ Es dürfte etwas schwierig sein, die `man pages` im Schwimmbad oder am Strand zu lesen. Ein Buch taugt dafür schon wesentlich besser.

Manche Parameterbeschreibungen sind nur übersetzt, besonders dann, wenn es sich um ganz einfache handelt (siehe Seite 582). Ist die Semantik eines Begriffs

sehr simpel, macht es wenig Sinn, eine neue Art der Beschreibung zu erfinden. Diese Parameter haben Eingang in den Referenzteil gefunden, da der Referenzteil komplett und vor allem konsistent sein soll.

Dieser Teil des Buches ist nützlich, weil man hier die einzelnen Voreinstellungen nachschlagen und gegebenenfalls in einer bestehenden Topologie entsprechend abändern kann. Er beschäftigt sich mit der Beschreibung der einzelnen Parameter und Variablen. Samba beschreibt eine Netzchnittstelle zwischen den verschiedenen Betriebssystemen. Da sich Samba um die Namensvergabe und Schreibweise von Variablen und Parametern, in Abhängigkeit der verschiedenen Betriebssysteme kümmert, ist ein Hinweis angebracht, der sich speziell auf die zwischen Unix- und Windows-artigen Betriebssystemen Unterschiede bezieht.

Die Namenskonventionen der einzelnen Betriebssysteme unterscheiden sich signifikant. Während alle Unix-Betriebssysteme strikt zwischen Groß- und Kleinschreibung unterscheiden, findet eine solche Differenzierung in anderen Betriebssystemen nicht statt. So können Bezeichner zwar im Betriebssystem OS/2 mit Groß-/Kleinschreibung eingegeben werden, deren maximale Länge auf 256 Buchstaben festgelegt ist, auch Windows 95 und Windows 98 erlauben diese Namensvergabe, aber leider berücksichtigen alle genannten Betriebssysteme die Schreibweise nicht. Während in einer Unix-Umgebung die Bezeichner `Borkner` und `borkner` als unterschiedlich interpretiert werden, sehen die genannten Betriebssysteme keinen Unterschied bei der Auswertung. Unix und Unix-artige Betriebssysteme lassen auch Leerzeichen bei der Namensvergabe einer Variablen zu, die von anderen Betriebssystemen als durch Leerzeichen getrennte Variablenmenge interpretiert wird. Das kann beim Einsatz von Samba als File- und Print-Server zu erheblichen Namenskonflikten führen. Aber auch für Unix ist die Verwendung von Namen, die Leerzeichen enthalten, nicht ratsam, weil es die Namensangabe erheblich kompliziert, da ein solcher Name immer von einem Paar Anführungszeichen eingeschlossen werden muss. Auch wenn das wirklich nützliche *«filename completion»* der Unix-Systeme hier vieles bei der Eingabe langer Datei- und Directory-Namen entschärft, sind generell Namen, die Leerzeichen enthalten, keine besonders gute Idee.

Schlimmer ist es bei relativ primitiven Single-User-Betriebssystemen wie beispielsweise DOS, das einerseits nur eine begrenzte Anzahl von Buchstaben pro Bezeichner zulässt (und interpretiert) und andererseits überhaupt keine Unterscheidung in der Schreibweise von Groß- und Kleinschreibung macht. So sind in einer DOS-Umgebung die Bezeichner `EineVariable` und `EINEVARI` völlig identisch.

Ein Samba-Server muss berücksichtigen, dass sich Clients in seiner Reichweite befinden, die nur über eingeschränkte Interpretationsmöglichkeiten der Namen verfügen.

Wenn auch Linux beispielsweise Leerzeichen in Service-Namen korrekt interpretiert, so ist es nicht ratsam, solche Service-Namen in einer inhomogenen Netzumgebung zu verwenden, da es sonst zu Problemen kommen kann. Leerzeichen werden ohnehin ignoriert, so dass die Bezeichnung der Service-Namen genauso gut ohne Leerzeichen vergeben werden kann. Auch sollte man die Service-Namen auf acht Buchstaben beschränken, für den Fall, dass sich DOS-Clients im Netz befinden oder optional angeschlossen werden können. Jeder Versuch eines DOS-Clients, sich mit einem abgeschnittenen Service-Namen bei einem Samba-Server anzumelden, der den Client mit mehr als acht Buchstaben angemeldet hat, muss fehlschlagen. Derartige Fehler sind äußerst schwer in einer laufenden Netzumgebung zu finden.

Die Begriffe «*Parameter*» und «*Variable*» werden im folgenden Kapitel synonym gebraucht. Mir ist klar, dass im mathematisch korrekten Sinne Unterschiede zwischen beiden Begriffen bestehen. Für die `/etc/samba/smb.conf`-Datei sind jedoch diese Unterschiede nicht von Belang, und sprachlich ist es einfach schöner, etwas Abwechslung in den Text zu bringen.

Kapitel 17

Samba Version 3.0.9-Referenz

17.1 Die Datei smb.conf

Dieses Kapitel gibt eine vollständige Übersicht der einzelnen Variablen. Einige davon haben Voreinstellungen (defaults), die immer dann gültig sind, wenn die entsprechende Variable «*nicht*» gesetzt wird. Andere haben keine Voreinstellung, das bedeutet, sie werden, wenn sie nicht oder falsch gesetzt werden, bei der Auswertung vom `nmbd` ignoriert.

Bei der Belegung der einzelnen Variablen kommt es stark auf das Zusammenwirken einzelner Variablen an. Alle möglichen und vor allem sinnvolle Kombinationen der Variablen anzugeben, ist aus rein technischen Gründen unmöglich und auch wenig hilfreich, da nur vergleichsweise wenige dieser Kombinationen auch wirklich nützlich sind und in der Praxis Verwendung finden. Zudem kommt es bei der Konfiguration sehr darauf an, welchen Typus von Clients der Server unterstützen muss. DOS-Clients sind mit dem unseligen 8×3 -Namensformat behaftet, während Windows for Workgroups-Clients die Angewohnheit haben, bei manchen Protokollen alle Passwords in Großbuchstaben umzuwandeln.

Alle diese Faktoren bestimmen den Inhalt einer gut konfigurierten Datei `/etc/samba/smb.conf`. Eine gute Mischung dieser Clients in einem gemeinsamen Netz macht die Konfiguration und Administration auch nicht einfacher. Auch wenn es vom Standpunkt des Administrators wünschenswert wäre, so lassen sich leider die zitierten Clients nicht alle aus einem bestehenden Netz entfernen, zu viele Applikationen stützen sich immer noch auf diese Plattformen.

Eine kleine Einschränkung gilt es noch zu erwähnen: Vollständig ist diese Übersicht schon, allerdings habe ich ganz wenige Parameter weggelassen, weil sie rein experimentellen Charakter haben. Das Buch wendet sich an Praktiker und Netzadministratoren. Da macht es wenig Sinn, zukünftige Parameterbelegungen zu beschreiben, deren korrekte Wirkungsweise noch keinesfalls garantiert werden kann.

Die zentrale Konfigurationsdatei für das Samba-Paket ist die `/etc/samba/smb.conf`-Datei, sie enthält alle Konfigurationsinformationen, die zur Laufzeit ausgewertet werden. Dies ist eine schier unübersehbare Menge von Parametern, deren Bedeutung in diesem Kapitel erläutert wird. Viele dieser Parameter sind uns bereits in den Beispielen begegnet und dort im Kontext erklärt worden.

Seit der Version 2.0.x existiert das Administrationswerkzeug `swat`, mit dem sich die `/etc/samba/smb.conf`-Datei über das HTTP-Protokoll konfigurieren und administrieren lässt. Das bedeutet, die Administration der Konfigurationsdatei kann, unabhängig vom Betriebssystem, mit einem HTTP-Browser (Netscape oder auch Internet Explorer) bewerkstelligt werden. Das Werkzeug `swat` sehen wir uns im Kapitel 18.3 auf Seite 671 genauer an. Dort werden auch Beispiele vorgestellt, wie eine solche `/etc/samba/smb.conf`-Datei administriert wird.

Die Erklärungen zu den Parametern stützen sich unter anderem auf die `man` pages der `/etc/samba/smb.conf`, die ursprünglich von *Karl Auer* zur Verfügung gestellt wurde. Wo sonst sollte man die nötigen Informationen alle herbekommen. Die komplette Liste der Autoren dieser `man` pages ist so lang, dass es wenig sinnvoll ist, auch nur einen von ihnen zu erwähnen. Jede Namensnennung wäre eine Hervorhebung, die allen Nichtgenannten nicht gerecht würde. Gedankt sei ihnen an dieser Stelle trotzdem. Aber auch den Samba-spezifischen News-Groups sei gedankt, denn der Dialog, den ich dort mit vielen kompetenten Administratoren führte, hat meinen Wissenstand über Samba enorm erweitert. Die Erfahrungen, welche mir dort vermittelt wurden, haben ebenfalls in großem Umfang den Inhalt des nun folgenden Kapitels bestimmt.

17.2 Kennzeichnung der Parameter

Für die folgende Darstellung stützen wir uns auf die in Absatz 4.4 auf Seite 118 gegebenen Erläuterungen, welche die Syntax und die Bedeutung der einzelnen Sections der `/etc/samba/smb.conf`-Datei behandelt.

Parameter spezifizieren die Attribute, die in den unterschiedlichen Sections vergeben werden. Manche dieser Parameter müssen zwingend in bestimmten hierfür vorgesehenen Sections definiert werden (Beispiel: `security` gilt nur in der Section `[homes]`), und andere können in jeder beliebigen Section verwendet werden (Beispiel: `create mode`). Um die Parameter unterscheiden zu können, werden sie, in Anlehnung an die `man` page `smb.conf`, wie folgt markiert:

- mit einem G für `[global]`-Section
- mit einem S für `[share]`-Section

Die auf diese Weise gekennzeichneten Parameter können auch in der `[global]`-Section verwendet werden, dann gelten sie freilich als Voreinstellung für alle angebotenen Dienste und für alle `shares`.

Alle nicht gekennzeichneten Parameter können überall verwendet werden.

Das Buch deckt die Samba-Server Version 3.0.9 ab. Seit der Herausgabe der Version 2.0.3, das ist die Version, die Anlass zur Erstausgabe dieses Buches gab, sind zwei weitere Subminor-Releases herausgegeben worden (2.0.4b und die 2.0.5a), die jeweils auch neue Parameter enthalten, daher werden beide Ausgaben, die 2.0.4b und die 2.0.5a, entsprechend gekennzeichnet. Für die Neuauflage des Buches wurde die zum Zeitpunkt der Drucklegung aktuelle Version Version 3.0.9 verwendet. Auch diese Version verfügt über neue Parameter, die entsprechend gekennzeichnet sind, so dass der Leser beurteilen kann, ob sich ein Upgrade für ihn lohnt. Die Anzahl der hinzugefügten Parameter lässt freilich vermuten, dass die Änderungen in dieser Version recht erheblich sind.

- ❑ Mit einem `2.0.4b` werden alle Parameter gekennzeichnet, die in der Version 2.0.4b hinzugefügt worden sind.
- ❑ Mit einem `2.0.5a` werden alle Parameter markiert, die der Version 2.2.1a zugeordnet werden müssen.
- ❑ Mit einem `2.0.6` werden alle Parameter markiert, die der Version 2.0.6 zugeordnet werden müssen. Die Version 2.0.6 enthält sechs neue Parameter, die auf den Seiten 526, 527, 609, 622 und 657 zu finden sind.
- ❑ Mit einem `2.0.7` werden alle Parameter markiert, die der Version 2.0.7 zugeordnet werden müssen. Die neue Windows-Version Windows 2000 bereitete einige Probleme, die noch bis zur Version 2.0.6 nur mühsam behoben werden konnten. Wenn man bis zu dieser Version Schwierigkeiten hatte, einen Windows 2000-Client dem Server bekannt zu machen, so sollte man unbedingt die gegenwärtig gültige Version des Samba-Servers verwenden, anstatt zu versuchen, die Clients mit der Version 2.0.6 zum Laufen zu bringen. Die Version 2.0.7 enthält optional fünf neue Parameter, die speziell für den professionellen Einsatz interessant sind (siehe Seiten 649, 650 und 664). Mit diesen Parametern kann eine Log-Umgebung definiert werden, die es dem Netzadministrator erlaubt, jeden Log-Versuch zu protokollieren, ob erfolgreich oder nicht. Diese Parameter sind aber nur verfügbar, wenn die Konfiguration des Samba-Pakets mit dem Kommandozeilenparameter `-with-utmp` erfolgte (siehe Seite 779).
- ❑ Mit einem `2.2.1a` werden alle Parameter markiert, die der Version 2.2.1a zugeordnet werden müssen. Hauptsächlich ist es die PAM-Unterstützung die verbessert wurde. Ob das allerdings von großem Nutzen ist, ist zweifelhaft, denn der PAM-Support kann nur in SMB-Netzen *ohne* verschlüsselte Passwörter eingesetzt werden und diese Netze werden immer seltener. Die Version 2.2.1a verhält sich bei der Namens-Interpretation genauso wie ein Windows NT-Server. Die Groß-/Kleinschreibung wird zwar beibehalten, aber nicht respektiert.

- ❑ Mit `2.2.2` werden alle Parameter gekennzeichnet, die in der Release-Version von Samba-2.2.2 hinzugekommen sind.
- ❑ Mit einem `3.0.9` werden alle Parameter gekennzeichnet, die zu der neuen Version Version 3.0.9 hinzugekommen sind. Bei dieser Version sind genau 329 Parameter implementiert, das heißt, es seit der Vorgänger-Version hat sich die Anzahl der Parameter fast verdoppelt. Das lässt schon auf eine tiefgreifende Änderung dieser Version schließen und ein Update sollte deshalb auf jeden Fall ins Auge gefasst werden.

Bemerkung:

Die Belegungen der Parameter mit den Werten *yes*, *on* und *true*, respektive *no*, *off* und *false* sind synonym gebraucht und können wechselseitig benutzt werden.

17.3 Erläuterung zu den Parametern

Es folgen nun Parameter der `/etc/samba/smb.conf`-Datei in alphabetischer Ordnung. Einige Parameter werden in den Beispielen sehr häufig vorkommen. Um redundante Informationen zu vermeiden, werden diese Parameter nicht für jedes Beispiel separat erläutert. Wenn man also einen Parameter findet, der im Kontext eines Beispiels «nicht» erklärt wird, so wird die Beschreibung in jedem Fall in diesem Kapitel zu finden sein. Wenn ein solcher Fall auftritt, findet man für gewöhnlich eine Referenz, die auf die entsprechende Seitenzahl verweist.

17.4 Die Parameter

`G`**abort shutdown script**`3.0.9`

Dieser Parameter akzeptiert als Belegung den qualifizierten Pfadnamen auf ein Script, welches vom `smbd`-Daemon aufgerufen wird, um die Ausführung eines Shutdown-Scripts anzuhalten.

Dieses Kommando wird im User-Mode ausgeführt.

Voreinstellung:

```
abort shutdown script =
```

Beispiel:

```
abort shutdown script = /sbin/shutdown -c
```

acl compatibilityS
3.0.9

Dieser Parameter legt fest, mit welcher ACL-Semantik des Betriebssystems der Samba-Server kompatibel sein soll (siehe [Glossar](#) [ACL](#)).

Mögliche Belegungen:

- winnt Für Windows NT 4
- win2k Für Windows 2000 und später
- auto

Die Einstellung bei der Belegung mit `auto` orientiert sich an der Betriebssystem-Version des Clientes.

Voreinstellung:

```
acl compatibility = Auto
```

Beispiel:

```
acl compatibility = winnt
```

add group scriptG
3.0.9

Als Belegung kann ein qualifizierter Pfadname angegeben werden, der vom `smbd`-Daemon als `root` ausgeführt wird, wenn eine neue Gruppe angefordert wird. Für den Gruppennamen wird der `%g`-Name expandiert. Dieses Script ist nur nützlich in Verbindung mit Windows NT Domain Administration Tools. Mit dem Script kann eine Gruppe mit wahlfreiem Namen erzeugt werden, um die Gruppennamen-Beschränkung von Unix zu umgehen. In diesem Fall muss das Script aber den numerischen Wert der Group ID-Nummer auf `stdout` ausgeben, da sonst eine Gruppenzuordnung nicht möglich ist.

Zu diesem Parameter existiert keine Voreinstellung.

add machine scriptG
3.0.9

Als Belegung kann ein qualifizierter Pfadname angegeben werden, der vom `smbd`-Daemon ausgeführt wird, wenn eine neue Maschine der Domain hinzugefügt wird.

Voreinstellung:

```
add machine script =
```

Beispiel:

```
add machine script = /usr/sbin/adduser -n -g machines -c Machine \  
-d /dev/null -s /bin/false %u
```

Index

Symbole

-valid 650
/etc/samba/smb.conf
80
/etc/user.map-Datei .
111
[global]-Section ... 80,
120
[homes]-Section 81, 121
#BEGIN_ALTERNATE ..
719
#DOM 719
#END_ALTERNATE 719
#PRE 719
Änderungen 841
Windows for Workgroups
63, 511
Windows 2000 69
LDAP 383
Password-Server 508
Password-Sicherheit . 455
Samba-Server nicht
sichtbar 444
16-Bit-Programme unter
NT 465

A

Abbilden von User-Name
456
abort shutdown script ...
504
Account 875
ACL 875

acl compatibility 505
Active Directory 875
add group script 505
add machine script .. 505
add printer command ...
506
add share command . 507
add user script 507
add user to group script .
508
addtosmbpass 715
admin users 509
ADSL 875
AFS 876
afs share 509
afs username map ... 509
algorithmic rid base . 510
Alias 876
allow trusted domains ...
510
Alpha-Version 876
Andere
Transportprotokolle .
189
announce as 511
announce version 511
Anpassen von Samba
Installieren von Samba
759
Anycast 876
API 876
Applet 876
Arbeitsgruppe 876

ARPA 877
ASCII 877
Attribute 793
auth methods 512
auto home 877
auto services 513
Automatische
Printer-Installation ..
453
automatisches Mouneten .
215
available 513

B

Backup Browser 298
Backup Domain
Controller ... V, 9, 11,
141, 201, 258, 313, 348,
353, 356, 366, 367, 372,
374, 379–382, 407, 411,
417, 418, 427, 485, 557,
877
bash 890
Basic Authentication
Method 386
bcast 592
BDC 878
BDC-Controller 379
Benutzer-ID-Nummer ...
201
Benutzerkonto 877
Benutzerprofil 877
Berechtigungen 214

Index

- Beta-Version 878
- Betriebssystem .. 878, 890
- Bibliotheken 200
- Bijektiv 878
- Bimodem 879
- Binary 879
- BIND 879
- bind interfaces only . 513
- Bindery 879
- block size 515
- blocking locks 515
- BOOTP 879
- Broadcast 194, 879
- browsable 516
- browse list 516
- browseable 516
- Browser 281, 879
 - Backup Browser ... 298
 - Domain Master
 - Browser 292
 - Domain-Controller 310
 - Samba und NT 310
- Browser-Einstellungen .. 482
- Browsing ... 289, 431, 458
- Byte Range Locking . 879

- C**
- C 749, 780
- C++ 674
- CA 880
- Caching 880
- CAP 522
- case sensitive 517
- case sensitive = yes/no .. 139
- casesignames 517
- Challenge 880
- change share command .. 518
- character set 518
- CIDR 880
- CIFS 880
- client lanman auth ... 519
- client ntlmv2 auth ... 520
- client plaintext auth . 520
- client schannel 521
- client signing 521
- client use spnego 522
- Clients 200, 353
- Close 867
- close socket 867
- codingsystem 522
- comment 523
- Computer-Account .. 880
- Computersuchdienst 880
- config file 523
- copy 524
- create mask 524
- create mode 525
- cron 880
- crontab 881
- csc policy 525

- D**
- Daemon 679, 864
- DAP 881
- Datagram-Dienste ... 206
- Datagramm-Kommunikation 868
- Datagramservice 881
- Dateirechte 214
- Dateistruktur 200
- Dateisystem 196
- deadtime 526
- debug hires timestamp .. 526
- debug leve 443
- debug level 527
- debug pid 527
- debug timestamp 527
- debug uid 527
- DECT 881
- default 529
- default case =
 - upper/lower 139
- default devmode 528
- delete group script .. 529
- delete printer command . 529
- delete readonly 530
- delete share command ... 530
- delete user from group
 - script 531
- delete user script 532
- delete veto files 533
- deny hosts 534
- DES 881
- dfree command 534
- DHCP 881
- Diagnose 825
 - ping 821
 - telnet 822
- Dictionary 881
- Die Printer-Section .. 131
- Die Unix-Lösung 457
- Die Samba-Lösung .. 457
- Dienste,
 - benutzerorientiert
 - Fremdrechner 840
 - Gruppen 840
 - Passwort-Kontrolle 840
 - Zielmaschine 840
 - Zugangsberechtigung . 840
- Dienste, geräteorientiert . 840
- Dienste,
 - maschinenorientiert . 840
- Digitale Signatur 882
- Directories 201
- directory 534
- directory mask 534
- directory mode 535
- directory security mask .. 535
- Directory-Struktur .. 201, 841
- Directory-Struktur,
 - verteilte 201
- disable netbios 536
- disable spoolss 536
- display charset 537
- DMB 882
- DNS 882
- dns proxy 537
- Domain 282, 283, 882

- domain logons 538
domain master 538
Domain Master 450
Domain Master Browser .
 292
Domain-Adressierung ...
 882
Domain-Controller .. 310,
 883
Domain-Member 883
dont descend 539
DOS 89, 259, 274, 275, 291
dos charset 539
dos filemode 540
dos filetime resolution ..
 540
dos filetimes 541
Druck-Kommando
 wechseln 238
- E**
E-Mail 383, 386, 387, 589,
 657, 881, 890
Editoren
 vi 861
Einleitung
 Passwort 39
einrichten
 Windows 59
Einrichtung 201
Einzelname 883
Emacs 200
enable rid algorithm . 541
encrypt passwords .. 542
Encryption 454
enhanced browsing .. 543
enumports command 543
Etwas Unix
 ping 858
EUC 522
exec 461
exec 544
exportieren 870
exports 871
- F**
fake directory create
 times 544
fake oplocks 544
FDDI 883
FIFO 884
Firewall 884
follow symlinks 545
force create mode 545
force directory mode 545
force directory security
 mode 546
force group 547
force user 548
Fortgeschrittene 115
 Grundkonfigurationen
 140
 Namensauflösung . 193
 Samba 115
 Server Level Security ..
 148
Freigabe 23, 884
Fremdrechner ... 840, 841
fstype 548
ftp .. 12, 93, 128, 150, 170,
 187, 198, 455, 472, 680,
 690, 697, 715, 810, 812,
 813, 840, 841, 856, 858,
 891
- G**
get quota command . 549
getwd cache 550
getwd cache = yes ... 442
[global]-Section ... 80,
 120
group 550
Grundkonfigurationen ..
 140
Gruppen 840
Gruppenname 884
guest account 551
guest ok 551
guest only 551
GUI .. 25, 48, 77, 738, 739,
 741, 743, 837, 884
- H**
Hashing 884
Herausforderung 885
Hetze Sebastian 864
HEX 522
hide dot files 552
hide files 552
hide special files 552
hide unreadable 553
hide unwriteable files 553
homedir map 553
[homes]-Section 81, 121
Hondel Dirk 864
Hop 885
host 592
Host Announcement . 885
hostname lookups ... 554
hosts allow 554
hosts deny 556
hosts equiv 556
hosts msdfs 554
- I**
IAB 885
idmap backend 557
idmap gid 557
idmap uid 557
importieren 870
include 558
inetd 679
inherit acls 558
inherit permissions .. 558
ink_absolute 872
Installation 201
Installieren von Samba ..
 759
interfaces 559
Internet . 66, 87, 189, 199,
 211, 258, 283, 286–288,
 310, 346, 379, 429, 466,
 469, 472, 542, 761, 765,
 773, 802, 803, 805, 808,
 867
Internet-Verbindung
 unterbinden 211
Internets ... 170, 287, 347
invalid users 559

Index

- IP-Adresse 191
- IPC 885
- IPng 885
- IPX-Protokoll 207

- J**
- J7BB 522
- J7BH 522
- J8BB 522
- J8BH 522
- JIS7 522
- JIS8 522
- JUBB 522
- JUBH 522
- JUNET 522

- K**
- Kämpf, Klaus 200
- KDE 738–741, 744
- keepalive 560
- Kerberos 886
- Kernel 886
- kernel change notify . 561
- kernel oplocks 561
- Konfiguration
 - Browser-Einstellungen
482
 - NetBIOS-Name ... 482
- Konfigurationsdatei
 - /etc/samba/smb.conf
80
- konfigurieren 793
- Konfigurieren 330
 - versehentliches
Löschen 793

- L**
- Löschen von Dateien 215
- LAN 869, 886
- lanman auth 561
- large readwrite 562
- LDAP 886
- ldap admin dn 562
- ldap delete dn 563
- ldap filter 563
- ldap group suffix 563
- ldap idmap suffix 563
- ldap machine suffix .. 564
- ldap passwd sync ... 564
- ldap port 564
- ldap replication sleep 565
- ldap server 565
- ldap ssl 566
- ldap suffix 566
- ldap user suffix 566
- Lendecke, Volker 463
- Lermen, Hans 864
- level2 oplocks 567
- LIFO 886
- link_relative 871
- Linux, Struktur von . 845
- Linux-Workstation 88, 95,
101, 103, 104, 110, 121,
126, 132
- Listening 865
- lm announce 568
- lm interval 568
- lmhosts 195, 592, 719
- load printers 569
- local master 569
- Local Master Browser 290
- lock dir 569
- lock spin count 570
- lock spin time 571
- locking 570
- Locking 458, 793
- log file 571
- log level 571
- Log-Dateien 135
- Login funktioniert nicht .
443
- Login unter WinNT 4.0 ..
464
- logon drive 572
- logon home 572
- logon path 572
- Logon Skripts 328
- Logon-Server 325
 - Konfigurieren 330
 - Logon Skripts 328
 - Profilverwaltung .. 327
 - System Policies 327
- Low Clients 452
- lppause command ... 573
- lpq cache time 574
- lpq command 574
- lpresume command . 575
- lprm command 576

- M**
- Müller Martin 864
- machine password
 - timeout 576
- magic output 576
- magic script 577
- mangle case 577
- mangle case = yes/no 139
- mangle prefix 580
- mangled map 578
- mangled names 578
- mangling char 580
- mangling method ... 580
- map acl inherit 581
- map archive 581
- map hidden 582
- map system 582
- map to guest 582
- map_daemon 872
- map_identity 872
- max reported print jobs ..
586
- max disk size 584
- max log size 584
- max mux 585
- max print jobs 585
- max protocol 586
- max smb processes . 587
- max ttl 587
- max wins ttl 587
- max xmit 588
- Max xmit 452
- maxopenfiles 585
- message command .. 588
- min passwd length .. 589
- min print space 590
- min protocol 590
- min wins ttl 591
- Mitglied einer Domain ..
349
- mount 201, 870

- mountd 870
 msdfs proxy 591
 msdfs root 591
 Multicast 886
- N**
- name cache timeout . 592
 name lookup 195
 name resolve order .. 592
 Name-Server 887
 Namensauflösung .. 190,
 193, 887
 Namensdienst 887
 Namensdienste 203
 Negotiate Protocol ... 887
 NetBEUI 62, 68, 185, 188,
 189, 887
 NetBIOS 187, 887
 Dienste blockieren . 805
 netbios aliases 593
 NetBIOS mit Samba . 208
 netbios name 593
 netbios scope 594
 NetBIOS und Samba . 189
 NetBIOS-Einzelname 204
 NetBIOS-Gruppenname .
 205
 NetBIOS-Name . 482, 684,
 887
 NetBIOS-Namenstyp 887
 Netz einbinden 201
 Netzdruckerdienste . 840
 Netzklassen 888
 Netztopologie 117
 nfs 394, 812, 840
 NFS ... 196, 200, 869, 888
 NFS-Dateisystem 869
 nfsd 870
 NIS 247–250, 349, 888
 nis homedir 594
 NIS+ 247, 248, 250
 nmbd 669
 no_root_squash 871
 NT 511
 nt acl support 595
 nt pipe support 595
 NT Server 511
- nt status support 596
 NT Workstation 511
 NT-Domain
 NIS 349
 NT-DOMAIN
 PAM 349
 ntlm auth 595
 null password 443
 null passwords 596
- O**
- obey pam restrictions 596
 ole locking compatibility
 597
 only user 597
 Open 867
 open socket 867
 oplocks 598
 Optimieren 442
 Optimieren der Clients ..
 450
 os level 599
 OS/2 470
 os2 driver map 599
 Overhead 868
- P**
- PAM VI, 348, 349,
 419–421, 503, 596, 599,
 600
 pam password change ...
 599
 panic action 600
 Parameter 504
 paranoid server security .
 600
 passdb backend 600
 passwd chat 602
 passwd chat debug .. 603
 passwd chat timeout 603
 passwd program 604
 password level 604
 password server 605
 Password-Server 212
 Passwords 165
 Passwort 39
 Passwort-Kontrolle .. 840
- path 607
 PDC 888
 Peer to Peer 346
 PGP 888
 PID 889
 pid directory 608
 ping 821, 858
 Pluggable Account Management
 596
 Polling 865, 889
 Port 866
 portmap 870
 posix locking 608
 postexec 461, 609
 PostScript ... 98, 100, 101,
 103–106
 preexec 461, 609
 preexec close 609
 preferred master 610
 preferred master 610
 preload 610
 preload modules 611
 preserve case 611
 preserve case = yes/no ..
 140
 Primary Domain
 Controller ... 74, 141,
 146, 283, 337, 355, 363,
 371, 379, 380, 382, 434,
 485, 877
 print command 613
 print ok 611
 Print-Kommando
 wechseln 238
 printable 611
 printcap 612
 printcap name 612
 printer 615
 Printer 131
 printer admin 615
 printer name 615
 printing 615
 private dir 616
 Probleme 825
 Diagnose 825
 Test 826

Index

- Probleme mit 100 MBit .. 472
Problemstellung 457, 463
Professionell
 Log-Dateien 135
Profile 48
profile acls 616
Profilverwaltung 327
Programmprojekte .. 200
Proxy 889
public 551
Public-Shares 209
- Q**
queuepause command ... 617
queueresume command . 617
- R**
read bmpx 618
read list 618
read only 618
read raw ... 443, 459, 618
read size 443
Read/Write 867
realm 619
receive- send to socket ... 867
Rechte 214
Referenz 501
 -valid 650
 Windows for Workgroups 511
 Password-Server .. 508
 abort shutdown script . 504
 acl compatibility .. 505
 add group script .. 505
 add machine script 505
 add printer command . 506
 add share command ... 507
 add user script 507
 add user to group script 508
 admin users 509
 afs share 509
 afs username map . 509
 algorithmic rid base ... 510
 allow trusted domains . 510
 announce as 511
 announce version . 511
 auth methods 512
 auto services 513
 available 513
 bcast 592
 bind interfaces only ... 513
 block size 515
 blocking locks 515
 browsable 516
 browse list 516
 browseable 516
 CAP 522
 case sensitive 517
 casesignames 517
 change share command 518
 character set 518
 client lanman auth 519
 client ntlmv2 auth . 520
 client plaintext auth ... 520
 client schannel 521
 client signing 521
 client use spnego .. 522
 codingsystem 522
 comment 523
 config file 523
 copy 524
 create mask 524
 create mode 525
 csc policy 525
 Daemon 679
 deadtime 526
 debug hires timestamp 526
 debug level 527
 debug pid 527
 debug timestamp .. 527
 debug uid 527
 default 529
 default devmode .. 528
 delete group script 529
 delete printer command 529
 delete readonly ... 530
 delete share command . 530
 delete user from group script 531
 delete user script .. 532
 delete veto files ... 533
 deny hosts 534
 dfree command ... 534
 directory 534
 directory mask 534
 directory mode 535
 directory security mask 535
 disable netbios 536
 disable spoolss 536
 display charset 537
 dns proxy 537
 domain logons 538
 domain master 538
 dont descend 539
 dos charset 539
 dos filemode 540
 dos filetime resolution 540
 dos filetimes 541
 enable rid algorithm ... 541
 encrypt passwords 542
 enhanced browsing 543
 enumports command .. 543
 EUC 522
 exec 544
 fake directory create times 544
 fake oplocks 544
 follow symlinks ... 545
 force create mode . 545

-
- force directory mode ... 545
 - force directory security mode 546
 - force group 547
 - force user 548
 - fstype 548
 - get quota command ... 549
 - getwd cache 550
 - group 550
 - guest account 551
 - guest ok 551
 - guest only 551
 - HEX 522
 - hide dot files 552
 - hide files 552
 - hide special files ... 552
 - hide unreadable ... 553
 - hide unwriteable files . 553
 - homedir map 553
 - host 592
 - hostname lookups . 554
 - hosts allow 554
 - hosts deny 556
 - hosts equiv 556
 - hosts msdfs 554
 - idmap backend 557
 - idmap gid 557
 - idmap uid 557
 - include 558
 - inherit acls 558
 - inherit permissions 558
 - interfaces 559
 - invalid users 559
 - JIS7 522
 - JIS8 522
 - JUBB 522
 - JUBH 522
 - JUNET 522
 - keepalive 560
 - kernel change notify ... 561
 - kernel oplocks 561
 - lanman auth 561
 - large readwrite 562
 - ldap admin dn 562
 - ldap delete dn 563
 - ldap filter 563
 - ldap group suffix .. 563
 - ldap idmap suffix . 563
 - ldap machine suffix 564
 - ldap passwd sync . 564
 - ldap port 564
 - ldap replication sleep .. 565
 - ldap server 565
 - ldap ssl 566
 - ldap suffix 566
 - ldap user suffix 566
 - level2 oplocks 567
 - lm announce 568
 - lm interval 568
 - lmhosts 592
 - load printers 569
 - local master 569
 - lock dir 569
 - lock spin count 570
 - lock spin timet 571
 - locking 570
 - log file 571
 - log level 571
 - logon drive 572
 - logon home 572
 - logon path 572
 - lppause command . 573
 - lpq cache time 574
 - lpq command 574
 - lpresume command ... 575
 - lprm command 576
 - machine password timeout 576
 - magic output 576
 - magic script 577
 - mangle case 577
 - mangle prefix 580
 - mangled map 578
 - mangled names ... 578
 - mangling char 580
 - mangling method . 580
 - map acl inherit 581
 - map archive 581
 - map hidden 582
 - map system 582
 - map to guest 582
 - max reported print jobs 586
 - max disk size 584
 - max log size 584
 - max mux 585
 - max print jobs 585
 - max protocol 586
 - max smbd processes ... 587
 - max ttl 587
 - max wins ttl 587
 - max xmit 588
 - maxopenfiles 585
 - message command 588
 - min passwd length 589
 - min print space 590
 - min protocol 590
 - min wins ttl 591
 - msdfs proxy 591
 - msdfs root 591
 - name cache timeout ... 592
 - name resolve order 592
 - netbios aliases 593
 - netbios name 593
 - netbios scope 594
 - nis homedir 594
 - NT 511
 - nt acl support 595
 - nt pipe support 595
 - NT Server 511
 - nt status support .. 596
 - NT Workstation ... 511
 - ntlm auth 595
 - null passwords 596
 - obey pam restrictions .. 596
 - ole locking compatibility ... 597
 - only user 597
 - oplocks 598

Index

- os level 599
- os2 driver map 599
- pam password change .
599
- panic action 600
- Parameter 504
- paranoid server
security 600
- passdb backend ... 600
- passwd chat 602
- passwd chat debug 603
- passwd chat timeout ..
603
- passwd program .. 604
- password level 604
- password server ... 605
- path 607
- pid directory 608
- posix locking 608
- postexec 609
- preexec 609
- preexec close 609
- preferred master ... 610
- preferred master .. 610
- preload 610
- preload modules .. 611
- preserve case 611
- print command 613
- print ok 611
- printable 611
- printcap 612
- printcap name 612
- printer 615
- printer admin 615
- printer name 615
- printing 615
- private dir 616
- profile acls 616
- public 551
- queuepause command
617
- queueresume command
617
- read bmpx 618
- read list 618
- read only 618
- read raw 618
- realm 619
- remote announce .. 619
- remote browse sync ...
620
- revalidate 620
- root 621
- root dir 621
- root directory 621
- root postexec 622
- root preexec 622
- root preexec close . 622
- Schrödingers Katze 651
- security 622
- security mask 627
- security=domain .. 532
- server schannel 628
- server signing 628
- server string 629
- set directory 629
- set primary group
script 630
- set quota command 630
- share modes 631
- short preserve case 631
- show add printer
wizard 632
- shutdown script ... 633
- SJIS Shift-JIS 522
- smb passwd file ... 633
- smb ports 634
- socket address 634
- socket options 634
- stat cache 635
- strict allocate 635
- strict locking 636
- strict sync 636
- sync always 637
- syslog 637
- syslog only 638
- template homedir . 638
- template primary group
638
- template shell 639
- time offset 639
- time server 639
- unicode 640
- unix charset 640
- unix extensions 640
- unix password sync ...
641
- Unterschiede 502
- update encrypted . 641
- use sendfile 649
- use spnego 649
- user 643
- username 643
- username level 645
- username map 646
- users 643
- utmp 649
- valid users 650
- veto files 651
- veto oplock files ... 652
- vfs object 653
- vfs objects 653
- volume 653
- wide links 654
- Win95 511
- winbind cache time 654
- winbind enable local
account 655
- winbind enum groups .
655
- winbind enum users ..
655
- winbind separator . 656
- winbind trusted
domains only ... 656
- winbind use default
domain 657
- wins 592
- wins hook 657
- wins proxy 661
- wins server 661
- wins support 662
- workgroup 662
- writable 662
- write cache size ... 663
- write list 663
- write ok 664
- write raw 664

- writeable 662
 ReferenzJ7BB 522
 ReferenzJ7BH 522
 ReferenzJ8BB 522
 ReferenzJ8BH 522
 Remote Procedure
 Call-Dienst 196
 remote announce 619
 remote browse sync . 620
 Ressourcen 841
 revalidate 620
 RFC 889
 RID 889
 rlogin .. 128, 715, 840, 841
 root 621
 root dir 621
 root directory 621
 root postexec ... 461, 622
 root preexec 461, 622
 root preexec close 622
 root_squash 871
 RPC ... 506, 529, 543, 632, 889
 RPC-Mount-Daemon 870
 RPC-NFS-Daemon .. 870
 RPC-Portmapper 870
 rpcclient 723
 RPM 16, 421, 422, 762, 763
 RTFM 890
 Rund um den Server 442
 Rund ums System ... 456
- S**
- SAM 140, 890
 Samba
 Attribute 793
 Browsing 431
 konfigurieren 793
 Zugriffsrechte 209
 Samba 3.0.0 426
 Samba anpassen 759
 Samba konfigurieren
 Locking 793
 Samba optimieren ... 811
 Samba und NT 310
 Samba, was ist
 NetBIOS 207
 Samba-Referenz 497
 Schrödingers Katze .. 651
 sec:smbmount 716
 secrets.tdb 698
 Section 80, 116, 118
 Sections 118
 security 622
 Security Account
 Manager 140
 security mask 627
 security=domain 532
 Semaphor 890
 Semaphore 757
 Server ... 86, 96, 121, 199, 200, 625, 627, 636, 818
 Server Level Security 148
 server schannel 628
 server signing 628
 server string 629
 Session Key 146
 Session-Dienste 206
 set directory 629
 set primary group script . 630
 set quota command .. 630
 sh 890
 Share 13, 29
 share modes 631
 Share Modes 452
 Shares 118
 Shell 890
 Shell- und
 Share-Verbindung ... 461
 short preserve case .. 631
 short preserve case =
 yes/no 140
 show add printer wizard 632
 shutdown script 633
 Sicherheit 840
 Sicherheitsmodell ... 140
 SID 890
 SJIS Shift-JIS 522
 SKey 146
 SMB 199, 890
 smb passwd file 633
 smb ports 634
 SMB und NFS 196
 smb.conf 501
 SMB/CIFS . 121, 122, 310, 336, 346, 665
 smbcalcs 698
 smbclient 680
 NetBIOS-Name ... 684
 smbcontrol 734
 close-share 737
 debuglevel 736
 force-election 735
 Nachrichten Typ .. 735
 Optionen 734
 Parameter 738
 ping 735
 printer-notify 737
 profile 736
 profilelevel 736
 Syntax 734
 Ziel 734
 smbd 665
 smbd-Daemon .. 16, 145, 504–506, 512, 515, 528, 530, 541, 544, 570, 571, 585, 587, 596, 602, 603, 608, 611, 616, 633, 737, 822, 823
 smbpasswd 706
 smbtar 727
 smbmount 718
 Smith, Neale 841
 SMTP 890
 socket address 634
 socket options 634
 spooling 120
 ssh 121, 128
 SSL 891
 stat cache 635
 Sticky Bit 891
 strict allocate 635
 strict locking 636
 strict sync 636
 Struktur der
 NetBIOS-Namen 190

Index

- Suchmaschine 891
swat
 inetd 679
 Listening 865
 Polling 865
 tcpwrapper 679
sync always 637
syslog 637
syslog only 638
System Policies 327
System-Error 1240 ... 460
- T**
T1 891
T3 891
tar 688, 689, 693, 732, 733
TCP 868
TCP-Stack 460
TCP/IP 868, 891
TCP/IP-Protokoll ... 207
tcpwrapper 679
tcsd 890
telnet 12, 93, 121, 128,
 158, 170, 187, 198, 455,
 536, 546, 547, 628, 715,
 822, 840, 841, 858
template homedir ... 638
template primary group .
 638
template shell 639
Test 826
testparm 698, 699
testprns 705
Theorie
 Andere
 Transportprotokolle .
 189
 Broadcast 194
 Datagram-Dienste . 206
 IP-Adresse 191
 IPX-Protokoll 207
 lmhosts 195
 name lookup 195
 Namensauflösung . 190
 Namensdienste 203
 NetBEUI 188
 NetBIOS mit Samba ...
 208
 NetBIOS und Samba ..
 189
 NetBIOS-Einzelname ..
 204
 NetBIOS-Gruppenname
 205
 Netz einbinden 201
 NFS 200
 NIS 247
 Password-Server .. 212
 Public-Shares 209
 Session-Dienste ... 206
 SMB 199
 SMB und NFS 196
 Struktur der
 NetBIOS-Namen 190
 TCP/IP-Protokoll . 207
 Unterschiede 197
 WINS-Server 193
 Zugriffsrechte 209
time offset 639
time server 639
Tips 441
 Password-Sicherheit ...
 455
 Samba-Server nicht
 sichtbar 444
 16-Bit-Programme
 unter NT 465
 Abbilden von
 User-Name 456
 Automatische
 Printer-Installation ..
 453
 Browsing 458
 debug leve 443
 Die Unix-Lösung .. 457
 Die Samba-Lösung 457
 Domain Master 450
 Encryption 454
 exec 461
 getwd cache = yes . 442
 Locking 458
 Login funktioniert nicht
 443
 Login unter WinNT 4.0
 464
 Low Clients 452
 Max xmit 452
 null password 443
 Optimieren 442
 Optimieren der Clients
 450
 postexec 461
 preexec 461
 Probleme mit 100 MBit
 472
 Problemstellung .. 457,
 463
 read raw 443, 459
 read size 443
 root postexec 461
 root preexec 461
 Rund um den Server ..
 442
 Rund ums System . 456
 Share Modes 452
 Shell- und
 Share-Verbindung ...
 461
 System-Error 1240 . 460
 TCP-Stack 460
 umount-Probleme . 460
 Unix-User-ID
 erzwingen 456
 Verschiedene
 Rechtevergabe .. 457
 Visual-C+ 465
 Wide Links 458
 Windows und
 AutoCAD 463
 winpopup 462
 write raw 443, 459
 Tips und Tricks 441
 Tools
 #BEGIN_ALTERNATE
 719
 #DOM 719

- #END_ALTERNATE .. 719
 #PRE 719
 lmhosts 719
 rpcclient 723
 smbpasswd 706
 Topologie 117
 Transfer 866
 Transportprotokolle . 189
- U**
 UDP 868, 892
 UDP/IP 868
 ugidd 872
 umount-Probleme ... 460
 UNC 892
 Unicast 892
 unicode 640
 Unicode 892
 Unix 837, 892
 Grundlagen 837
 Rechtevergabe 838
 unix charset 640
 unix extensions 640
 unix password sync . 641
 Unix-Rechte 838
 Unix-System 357
 Unix-User-ID erzwingen
 456
 Unix-Workstation ... 104
 Unterschiede ... 197, 502
 update encrypted 641
 use sendfile 649
 use spnego 649
 user 643
 User-Identifizierung . 840
 User-Profile 48
 username 643
 username level 645
 username map 646
 users 643
 utmp 649
 utmp dir 650
- V**
 valid users 650
 Variablensubstitution 136
- Verschiedene
 Rechtevergabe .. 457
 versehentliches Löschen .
 793
 Verzeichnisbaum 196
 Verzeichnisdienst 383
 veto files 651
 veto oplock files 652
 vfs object 653
 vfs objects 653
 vi 861
 Visual-C+ 465
 volume 653
- W**
 WAN 892
 WebNFS 892
 Werkzeuge
 #BEGIN_ALTERNATE
 719
 #DOM 719
 #END_ALTERNATE ..
 719
 #PRE 719
 lmhosts 719
 wide links 654
 Wide Links 458
 Win95 511
 winbind cache time .. 654
 winbind enable local
 account 655
 winbind enum groups ...
 655
 winbind enum users . 655
 winbind separator ... 656
 winbind trusted domains
 only 656
 winbind use default
 domain 657
 winbindd V, VI, 348,
 417–424
 Winbindd 417
 winbindd-Daemon .. 417,
 420, 421, 424, 638, 639
 Windows 59, 208
 Windows for Workgroups
 63
 Windows 2000 69
 einrichten 59
 Windows 2000 V, 59,
 69–72, 74, 75, 89, 153,
 175, 269, 348, 453, 503,
 506, 529, 536, 543, 632,
 642, 643, 760, 773, 774,
 899, 909
 Windows NT Domain ...
 337, 346
 Client 353
 Mitglied einer Domain
 349
 Peer to Peer 346
 Windows NT-Client . 737
 Windows NT-Server ... 9,
 173, 330, 331, 355, 356,
 430, 432
 Windows und AutoCAD
 463
 Windows 3.0 169
 Windows 3.1 . 59, 60, 119,
 129, 208, 259, 262, 268,
 269, 315, 461
 winpopup 462
 wins 592
 WINS . 194, 346, 417, 543,
 661, 892
 wins hook 657
 wins proxy 661
 wins server 661
 wins support 662
 WINS-Server ... 193, 194,
 321–324, 330, 346, 435,
 587
 workgroup 662
 Workgroup 283
 writable 662
 write cache size 663
 write list 663
 write ok 664
 write raw .. 443, 459, 664
 writeable 662
 wtmp directory 664
- X**
 X11 578, 588

Index

xftp 841

Y

YaST . 14, 19, 98, 104, 112,
128, 275, 388, 409, 413,
475, 671, 706, 713, 759,
763, 777, 778, 780, 787,
829, 832, 833, 839, 856,
858, 859

Yellow Pages 893

yes/no 117

YP 893

Z

Zielmaschine 840

Zugangsberechtigung ...
840

Zugriffsrechte 209

Samba 209