Tag 5

Die Datei smb.conf: Samba mitteilen, was es tun soll

Die Datei `smb.conf` ist sozusagen das Herzstück von Samba. Sie wird sowohl von `smbd` und `nmbd` als auch von vielen der anderen Tools benutzt, die in der Samba-Programmfamilie enthalten sind. Und obwohl sie wahrscheinlich mehr Parameter hat als Godzilla Zähne, ist sie nicht sehr schwer zu verstehen. Dieses Kapitel bietet eine vertiefte Darstellung der Datei `smb.conf`. Sie werden sich den allgemeinen Aufbau der Datei, Variablen, die während der Laufzeit zur Anwendung kommen, und einige der globalen Parameter ansehen, die das allgemeine Verhalten von Samba kontrollieren.

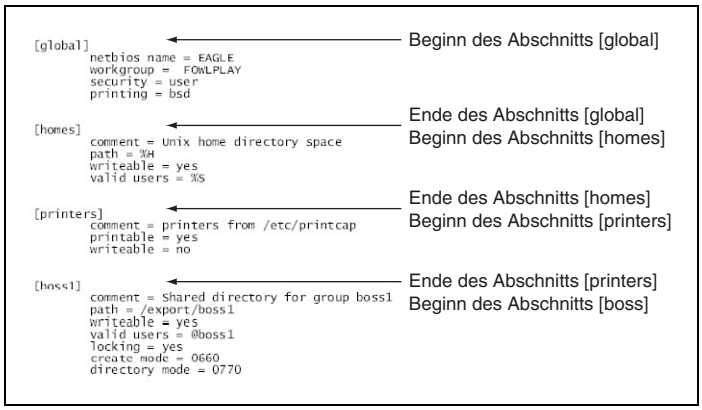
5.1 Aufbau

Eine Standard-`smb.conf`-Datei besteht aus verschiedenen Abschnitten, die jeweils mehrere Parameter enthalten. Diese Erklärung ist zwar richtig, aber nicht sehr hilfreich.

Die folgende Definition ergibt vielleicht mehr Sinn: Eine Samba-Konfigurationsdatei ist eine ASCII-Textdatei, die durch Abschnittsüberschriften logisch unterteilt ist, welche durch umschließende eckige Klammern (`[ ]`) gekennzeichnet sind. So wäre z.B. `[footbar]` eine gültige Abschnittsüberschrift. Die Namen der Abschnitte, Parameter und Werte sind nicht groß-/kleinsensitiv, es sei denn, sie gehören zum Betriebssystem, wie es für einen Verzeichnispfad der Fall ist. Jeder Abschnitt wird bis zur nächsten Abschnittsüberschrift fortgesetzt. Sambas `smb.conf` hat drei integrierte Abschnitte namens `[global]`, `[homes]` und `[printers]`.

Abbildung 5.1. stellt die integrierten Abschnitte und einen Beispielabschnitt dar. Weil Abschnittsüberschriften nicht groß-/kleinsensitiv sind, stehen die Bezeichnungen `[global]`, `[GLOBAL]` und `[Global]` alle für den gleichen Abschnitt. Die vier Einstellungen – `netbios name`, `workgroup`, `security` und `printing` – stehen für globale Parameter. Daher sind sie alle im Abschnitt `[global]` zu finden, der bei der nächsten Abschnittsüberschrift, `[homes]`, endet. Der letzte Abschnitt, `[boss1]`, steht für eine Festplattenfreigabe, die für diesen Server konfiguriert wurde.

Abb. 5.1: Der generelle Aufbau von smb.conf



5.1.1 [global]

Der Abschnitt `[global]` enthält Parameter, die für die allgemeine Funktionalität des Servers von Bedeutung sind. Die Parameter `netbios name` und `workgroup`, die in Kapitel 4, »Installation und Testen der Konfiguration«, kurz beschrieben wurden, sind Beispiele für globale Parameter. Diese und andere Parameter werden später in diesem Kapitel ausführlicher dargestellt.

5.1.2 [homes]

Der Abschnitt `[homes]` wurde bereits beim Beispiel-Samba-Server in Kapitel 4 kurz erwähnt. Diese spezielle Freigabe ermöglicht es Benutzern, auf ihre Home-Verzeichnisse zuzugreifen, ohne dass eine spezielle Freigabe für jeden Benutzer eingerichtet werden muss. Der Prozess läuft folgendermaßen ab:

1. Samba empfängt eine Verbindungsanfrage.
2. Die Datei `smb.conf` wird nach dem Namen der verlangten Freigabe durchsucht.

3. Wird der verlangte Name nicht gefunden und wurde die [homes]-Freigabe konfiguriert, durchsucht Samba die Datei /etc/passwd nach einem entsprechenden Benutzernamen.
4. Wird ein entsprechender Benutzername gefunden, erzeugt Samba eine Kopie der [homes]-Freigabe und ändert den Namen homes in den gefundenen Benutzernamen um. Ist kein Pfad angegeben, wird dieser auf das Home-Verzeichnis des Benutzers eingerichtet, wie er im /etc/passwd-Eintrag definiert ist.
5. Wird kein entsprechender Benutzername gefunden, gibt Samba die Fehlermeldung Invalid resource in tree connection request an den Client zurück.

5.1.3 [printers]

Der dritte integrierte Abschnitt, [printers], ist [homes] ähnlich. Der Unterschied liegt in der Art der Ressource, die hier verfügbar gemacht wird. [homes] erzeugt Home-Verzeichnisse aus der /etc/passwd, während [printers] Drucker aus der /etc/printcap freigibt. Wenn Sie ein anderes Drucksystem als BSD benutzen, müssen Sie eine Hilfs-printcap-Datei für Samba erstellen, um Druckernamen zu authentifizieren. Weitere Informationen hierzu finden Sie in Kapitel 8, »Drucker«.

5.1.4 Die restlichen Abschnitte von smb.conf

Jeder Abschnitt außer [global] wird als freigegebene Ressource (kurz *Freigabe*) angesehen, daher müssen die Abschnitte generellen Benennungskonventionen für Freigaben folgen.

Um benutzerdefinierte Freigaben einzurichten, brauchen Sie nur eine Abschnittsüberschrift, wie z.B. [foo], und die notwendigen Parameter eingeben, die in den Kapiteln 6, »Sicherheitsebenen und Passwörter«, und 7, »Dateifreigaben«, dargestellt werden. Der SMB-Client (z.B. ein Windows-PC) kann dann über den Netzwerkpfad \\Servername\FOO auf die Freigabe zugreifen. Ich habe vorher erwähnt, dass die Abschnittsüberschriften in smb.conf nicht groß-/kleinsensitiv sind. Daher beziehen sich [foo] und [FOO] auf die gleiche Freigabe. Darum kann der PC-Client \\Servername\FOO mounten, wenn die Freigabe als [foo] definiert ist.

Je nachdem wie sehr Sie es mögen, Ihre Arbeit zu dokumentieren – ich hoffe für die arme Person, die nach Ihnen kommt und Ihre Kreation übernehmen muss, dass Sie es sehr mögen –, können Sie freizügig Kommentare

einfügen, indem Sie ein Semikolon (;) oder eine Raute (#) als erstes Zeichen einer Zeile einsetzen. Kommentare werden beim Zeilensprung beendet:

```
; Dies ist ein Kommentar
# und dies auch
```

Tabelle 5.1 ist eine Zusammenfassung der Formate für die `smb.conf`-Inhalte, die ich dargestellt habe.

Tabelle 5.1: Zusammenfassung der smb.conf-Formate

Eingabe	Format
Abschnitt	Zeile, die einen Zeichenstring enthält, der in eckige Klammern eingeschlossen ist, z.B. `[foo]`.
`[global]`	Spezieller Abschnitt, der Parameter enthält, die für die generellen Samba-Einstellungen und die Einstellungen der Standardfreigaben gelten.
`[homes]`	Dynamische Freigabe, die Namen aus der `/etc/passwd` holt.
`[printers]`	Dynamische Freigabe, die Druckernamen aus einer spezifizierten `printcap`-Datei holt.
Kommentar	Zeile, der das Zeichen ; oder # vorangestellt ist.
Parameter	Konfigurationsparameter, gefolgt von = und einem Wert, z.B. `writable = yes`.

5.2 Variablen

Sie können in `smb.conf` verschiedene Variablen benutzen. Diese Makros, die durch das Zeichen % gekennzeichnet sind, werden während der Analyse der Konfigurationsdatei beim Ablauf ersetzt. Wenn z.B. Benutzer `jdoe` eine Anfrage für die Aufnahme einer Arbeitssitzung überträgt, analysiert Samba die `smb.conf` und ersetzt alle Entsprechungen der Variable `%U` durch `jdoe`. Tabelle 5.2 listet alle verfügbaren `smb.conf`-Variablen auf.

Tabelle 5.2: smb.conf-Variablen

Variable	Beschreibung
`%a`	Die Architektur des entfernten Rechners. Zuverlässigkeit wird nicht hundertprozentig garantiert, aber in der Regel ist es in der Praxis gut genug. Derzeit unterstützte Werte sind `Samba`, `WfWg`, `WinNT` und `Win95`. Windows 2000 ist tatsächlich Windows NT 5.0 und wird daher als `WinNT` erkannt
`%d`	Die Prozess-ID des aktuellen Server-Prozesses
`%g`	Die primäre Gruppe des Benutzernamens `%u`

Tabelle 5.2: smb.conf-Variablen (Fortsetzung)

Variable	Beschreibung
%G	Die primäre Gruppe des Benutzernamens %U
%h	Der Name des Internet-Hosts, auf dem Samba läuft
%H	Das Home-Verzeichnis für den Benutzernamen %u
%I	Die IP-Adresse des Client-Rechners in Dezimalpunktschreibweise
%L	Der NetBIOS-Name des Servers
%m	Der NetBIOS-Name des Client-Rechners
%M	Der Internet-Host-Name des Client-Rechners
%N	Der Name Ihres NIS-Home-Directory-Servers, wie er in der auto.home-Map spezifiziert ist. Wenn Sie Samba ohne AUTOMOUNT-Unterstützung kompiliert haben, ist dies das Gleiche wie %L
%p	Der Pfad zum Home-Verzeichnis des Benutzers, wie er in auto.home definiert ist. Es wird vorausgesetzt, dass der NIS-Map-Eintrag durch einen Doppelpunkt getrennt und als %N:%p aufgeteilt ist
%P	Das Root-Verzeichnis des aktuellen Dienstes
%R	Das Protokoll, das während der Protokollabstimmungsphase bei Verbindungsaufnahme ausgewählt wurde
%S	Der Name der aktuellen Freigabe
%T	Aktuelles Datum und Zeit
%u	Benutzername der aktuellen Freigabe
%U	Der Benutzername, den der Client bei Aufnahme der Arbeitssitzung verlangt hat. Dies ist nicht unbedingt der gleiche wie der, der benutzt wurde
%v	Samba-Versionsnummer

Diese Variablen können auf vielfache Art und Weise benutzt werden. Eine Variable kann überall dort eingesetzt werden, wo ein Textstring zugelassen ist. Der folgende [global]-Parametereintrag z.B. würde Samba veranlassen, Verbindungsinformationen in eine Datei namens /var/log/log.*NetBIOS-Name* zu schreiben, wobei *NetBIOS-Name* durch den NetBIOS-Namen des Clients ersetzt wird.

```
log file = /var/log/log.%m
```

Hier ist noch ein Beispiel, das Samba mitteilt, abhängig vom Betriebssystem des sich verbindenden Clients ein anderes Domain-Logon-Skript zu benutzen:

```
logon script = %a.bat
```

Die Namen der verfügbaren Logon-Skripte wären `WfW.bat`, `Win95.bat` und `WinNT.bat`. Domain-Logons werden ausführlich in Kapitel 21, »Windows-9x-Domänenkontrolle«, und Anhang A, »Experimentelle PDC-Unterstützung«, dargestellt.

Um die Variablen ausführlich zu erklären, ist vielleicht die `[homes]`-Freigabe, die ich für das Beispiel in Kapitel 4 benutzt habe, besser geeignet:

```
; Freigabename
[homes]
   comment = Unix-Home-Verzeichnisbereich
   path = %H
   writable = yes
   valid users = %S
   create mode = 0600
   directory mode = 0700
   locking = no
```

Der Eintrag `valid users = %S` schränkt Verbindungen auf den Benutzer ein, dessen Benutzername dem der Freigabe entspricht. Denken Sie an meine frühere Erklärung der `[homes]`-Freigabe. Findet Samba eine Entsprechnung des Freigabenamens in der Datei `/etc/passwd`, wird eine Freigabe mit den Parametern aus der `[homes]`-Definition erzeugt, die in den entsprechenden Benutzernamen umbenannt wird. Daher ist der einzige Benutzer, dem die Verbindung erlaubt wird, der Eigentümer des Home-Verzeichnisses.

Noch ein letztes Beispiel, bevor es weitergeht. In meinem Beruf verwalte ich etwa 30 verschiedene Samba-Server, die unter verschiedenen Betriebssystemen laufen. In der Regel werden alle Server gleichzeitig auf die gleiche Version von Samba aktualisiert, aber es gibt immer einige Ausnahmen. Um schnell die installierte Version auf einem Server feststellen zu können, hat jede `smb.conf` im Abschnitt `[global]` einen Eintrag, der dem folgenden ähnelt:

```
server string = samba print server for administration [%v]
```

Mit dem Parameter `server string` legen Sie den Text fest, der neben dem Rechnernamen in Browse-Listen angezeigt wird, die über Tools wie die Netzwerkumgebung verfügbar sind. `%v` wird dynamisch auf die Version des aktuell laufenden `nmbd`-Prozesses aktualisiert. Um also festzustellen, welche Samba-Version auf einem Server läuft, benutze ich einfach den Befehl `net view \\Servername` von einem Windows-Rechner und untersuche den ausgegebenen String des Servers.

5.3 Parameter

Ein schnelles `grep` durch die `smb.conf`-2.0-Manpage legt über 130 einzelne globale Parameter und etwa 100 weitere offen, die mit Freigaben zu tun haben. Die `smb.conf`-2.0-Manpage ist etwa 8.500 Zeilen lang. Es erübrigt sich zu sagen, dass für die Konfiguration Ihres Servers ziemlich viele Optionen zur Verfügung stehen. In diesem Abschnitt lernen Sie einige der gebräuchlichsten Optionen kennen. Ich hebe die Darstellung einiger `[global]`-Optionen für spätere Kapitel auf, in denen der Kontext besser zur Funktion des Parameters passt. Eine komplette Auflistung der aktuellen `[global]`-Parameter finden Sie, wie immer, in der `smb.conf`-Manpage.

Die Werte für Parameter lassen sich, mit wenigen Ausnahmen, in drei Kategorien aufteilen:

- ✘ Bei der ersten wird der Wert als Zeichenstring eingegeben, wie z.B. `jerryc` oder `samba server`. Groß-/Kleinschreibung wird in Textstrings beibehalten.
- ✘ Die zweite ist ein boolescher Parameterwert, der `yes/no`, `true/false` oder `1/0` akzeptiert. Boolesche Werte sind nicht groß-/kleinsensibel, also sind `YES`, `Yes` und `yes` für Samba identisch.
- ✘ Die dritte Kategorie der Parameter akzeptiert einen numerischen Wert. Sie müssen jeden Parameter überprüfen, um zu bestimmen, ob es sich um eine ganze Zahl oder eine Basis handelt, wie z.B. einen Erstellungsmodus, der eine Oktalzahl ist.

Parameter haben die Form *`Name`* = *`Wert`*, z.B.:

```
netbios name = EAGLE
```

Nur das erste Gleichheitszeichen wird für die Analyse des Parameters und seines Werts benutzt. Der Wert beginnt beim ersten nicht leeren Zeichen nach dem Gleichheitszeichen und endet mit dem ersten Zeilenumbruch, dem kein `\`-Zeichen vorangestellt ist. Daher hat die folgende Einstellung die gleiche Bedeutung wie das obenstehende Beispiel:

```
netbios name        =        EAGLE
```

netbios name

Sie haben den Parameter `netbios name` oben schon kurz gesehen. Über diesen Parameter richten Sie NetBIOS-Rechnernamen des Samba-Servers ein. Wie die meisten anderen Parameter hat auch dieser einen Standard-

wert, nämlich den Hostnamen des Servers. Es ist möglich, diesen Parameter nicht einzurichten und den Standardwert zu benutzen, aber ich persönlich ziehe es vor, den Rechnernamen explizit zu definieren.

```
Standard:     netbios name = Internet-Hostname des Rechners
```

Ich möchte die Themen Namensauflösung und Browsing an dieser Stelle nicht zu sehr vertiefen, aber ich habe die Erfahrung gemacht, dass es einfacher ist, wenn NetBIOS-Rechnername und Internet-Hostname gleich sind, es sei denn, Sie haben einen sehr guten Grund, verschiedene Namen zu verwenden. Wenn z.B. der Hostname des Servers `eagle` ist, würde ich den NetBIOS-Namen explizit wie folgt einrichten:

```
netbios name = EAGLE
```

Alle gültigen DNS-Namen, die nicht länger als 15 Zeichen sind, sind auch gültige NetBIOS-Namen. Das Gegenteil trifft nicht zu, da einige Zeichen, wie z.B. eine Tilde (~), für Rechnernamen benutzt werden können, aber nicht gültig sind, wenn es um DNS geht.

netbios aliases

In Kapitel 2, »Windows-Netzwerke«, habe ich im Abschnitt »NetBIOS-Überblick« erwähnt, dass es in einer NetBIOS-Verbindung einen »Anrufernamen« auf Seiten des Clients und einen vom Client verlangten »angerufenen« Namen gibt. Ein NetBIOS-Server antwortet nur auf Anfragen, die seinem angerufenen (*called*) Namen entsprechen. Der Parameter `netbios aliases` ermöglicht Samba, auf mehrere angerufene Namen zu antworten. Das heißt, Sie können den gleichen Server in einer Arbeitsgruppe unter mehreren Namen sehen, wenn Sie auf einem Windows-Client durch die Netzwerkumgebung browsen. Jeder Server-Name könnte verschiedene Freigaben zur Verfügung stellen, die sich alle auf dem gleichen Rechner befinden. Standardmäßig sind keine `netbios aliases` eingerichtet.

```
Standard:     netbios aliases =
```

Abbildung 5.2 zeigt einen Server in der Windows-95-Netzwerkumgebung, wenn folgende Einstellung aktiviert ist:

```
netbios aliases = admin acct business
```

Der primäre NetBIOS-Name des Samba-Servers, `BILBO`, erscheint ebenfalls in der Auflistung. Sie sollten beachten, dass nur der primäre Name (d.h. `netbios name = ...`) benutzt wird, wenn auf Anfragen für Domänen-Logons reagiert werden soll oder wenn der Server als Browse-Server konfiguriert ist.

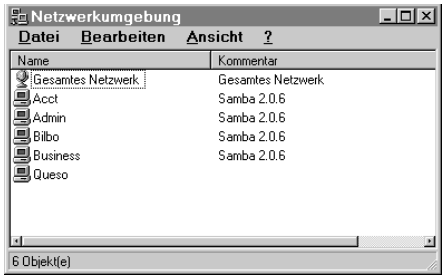


Abb. 5.2: Ein Beispiel für einen einzelnen Samba-Server mit mehreren NetBIOS-Aliasen

workgroup

Der Parameter `workgroup` entscheidet, zu welcher Arbeitsgruppe der Server gehört, wenn er auf Anfragen von Clients reagiert. Die Zugehörigkeit zur Arbeitsgruppe beeinflusst auch andere Einstellungen wie z.B. Domänen-Logons, Domänenzugehörigkeit und Browse-Dienste.

Der Standardwert für diesen Parameter wird während der Kompilierung durch das Makro `WORKGROUP` eingerichtet, das in dem Makefile definiert ist.

```
Standard:     workgroup = während Kompilierung definiert
```

Ein Arbeitsgruppenname ist ein NetBIOS-Gruppenname und muss daher den Standardnamenskonventionen folgen (siehe Kapitel 2). Ein Beispiel:

```
workgroup = FOWLPLAY
```

server string

Der Parameter `server string` definiert den Textstring, der im Kommentarabschnitt des Windows-Druckmanagers angezeigt wird. Er wird auch mit dem NetBIOS-Namen des Rechners angezeigt, wenn Sie z.B. in der Netzwerkumgebung durch das Netzwerk browsen. Der Parameter akzeptiert, wie andere auch, `smb.conf`-Variablen. So können Sie hier z.B. über die Variable `%v` die aktuell laufende Version von Samba überprüfen.

Die Standardeinstellung ist:

```
Standard:     server string = Samba %v
```

Ich gebe normalerweise eine etwas umfassendere Beschreibung ein, die den Standort eines Rechners bestimmt:

```
server string = Drucker-Server in Abteilung Einkauf [%v]
```

Folgendes Beispiel zeigt die Server in der aktuellen Arbeitsgruppe von einem Windows-NT-4.0-SP5-Rechner gesehen:

```
H:\>net view
Server-Name       Beschreibung
-------------------------------------------------------------
\\BURRITO         Drucker-Server in Abteilung Einkauf ↵
     [2.1.0-prealpha]
\\PIZZA           Samba [1.9.18p7]
Der Befehl wurde erfolgreich ausgeführt.
```

log file

Über diesen Parameter können Sie den während der Kompilierung festgelegten Standardstandort der `smbd`-Logdateien außer Kraft setzen.

```
Standard:    log file = bei Kompilierung eingerichtet
```

Es gibt einige Besonderheiten für diesen Parameter. Sie sollten wissen, in welcher Reihenfolge die Dinge geordnet sind.

1. Wird während des Starts über den Parameter `-l` eine Datei spezifiziert, schreibt `smbd` die ersten Log-Einträge in die Datei, die in der Befehlszeile angegeben ist. Wird während des Starts kein Standort festgelegt, protokolliert `smbd` die Log-Informationen in der Datei, die während der Kompilierung angegeben wurde.
2. Wird bei einer Analyse der Konfigurationsdatei der `log-file`-Parameter gefunden, werden alle zukünftigen Log-Einträge in die Datei geschrieben, die durch den Wert des Parameters spezifiziert sind.

Weil Samba also zunächst nichts über den in der `smb.conf` definierten `log-file`-Standort weiß, schreibt es einige Startinformationen in die Logdatei, die es beim Starten kennt.

Sie können den während der Kompilierung festgelegten Standardstandort für die Logdatei von `nmbd` nur überschreiben, indem Sie beim Start den Parameter `-l` verwenden.

Dieses Beispiel erzeugt eine separate Logdatei für jeden Benutzer, der sich mit dem Server verbindet (oder versucht, sich zu verbinden):

```
log file = /var/log/log.%U
```

max log size

Der Parameter `max log size` nimmt als Wert eine ganze Zahl an, mit der die maximale Größe für die Logdatei in Kilobyte spezifiziert wird. Samba überprüft regelmäßig die Größe der Logdateien. Hat eine Logdatei die definierte maximale Größe überschritten, benennt Samba die Datei mit der Erweiterung `.old` um und erstellt eine neue. Existiert bereits eine Datei mit gleichem Namen (*Logdatei*`.old`), wird sie überschrieben. Der Standardwert ist auf 5 Mbyte gesetzt.

```
Standard:    max log size = 5000
```

Sie können hier einen beliebigen Wert einfügen. Der folgende Eintrag richtet die maximale Größe der Logdatei auf 2 Mbyte ein:

```
max log size = 2000
```

syslog

Damit dieser Parameter in Kraft treten kann, müssen Sie während der Kompilierung die `syslog`-Unterstützung aktivieren:

```
./configure -- with-syslog
```

Der Parameter `syslog` nimmt als Wert eine ganze Zahl an und gleicht die Samba-Debug-Prioritäten mit den `syslog`-Log-Prioritäten ab. Die Entsprechungen finden Sie in Tabelle 5.3. Nur Samba-Debug-Meldungen mit einer Priorität, die kleiner als der definierte Wert ist, werden an den `syslogd`-Daemon übertragen. Standardmäßig werden daher nur Debug-Meldungen mit der Priorität 0 an `syslog` gesendet, obwohl der Wert auf `1` eingestellt ist.

```
Standard:    syslog = 1
```

Tabelle 5.3 listet die verschiedenen Debug-Prioritäten und ihre `syslog`-Entsprechungen auf.

Samba-Debug-Level	syslog-Level
0	LOG_ERR
1	LOG_WARNING
2	LOG_NOTICE
3	LOG_INFO
>3	LOG_DEBUG

Tabelle 5.3: Samba-Debug-Level und entsprechende syslog-Prioritäten

Sollen alle Meldungen, die der Priorität LOG_NOTICE entsprechen, an den syslogd-Prozess übertragen werden, fügen Sie folgenden Eintrag in smb.conf ein:

```
syslog = 3
```

syslog only

Dieser boolesche Parameter bestimmt, ob Meldungen nur an den syslog-Daemon gesendet werden und nicht an die normalen Debug-Logdateien. Dieser Parameter wird zusammen mit dem Parameter syslog verwendet und setzt voraus, dass während der Kompilierung die syslog-Unterstützung aktiviert wurde. Standardmäßig werden Debug-Einträge zusätzlich zu den syslog-Dateien auch an die Standard-smbd- und -nmbd-Logdateien übertragen. Durch folgende Einstellung können Sie Samba veranlassen, Logging-Informationen nur an den syslog-Daemon weiterzugeben:

```
syslog only = yes
```

debug level

Über den Parameter debug level, der auch log level genannt wird, können Sie den maximalen Grad (Level) der Debug-Meldungen einstellen, die auf die Festplatte geschrieben werden. Der Parameter hat einen Standardwert von 2.

```
Standard:    debug level = 2
```

Der Parameter debug level gilt sowohl für smbd als auch für nmbd. Sie werden Samba-Logs zu Debugging-Zwecken ausführlich in späteren Kapiteln verwenden. Hier ein Beispiel, in dem der Log-Level auf 5 eingestellt ist:

```
debug level = 5
```

Je höher der Debug-Level eingestellt ist, umso ausführlicher werden Meldungen in die Log-Dateien geschrieben.

Wenn Sie über die Option -d in der Befehlszeile einen Debug-Level definieren, setzt dieser Wert die Einstellung für den Parameter debug level außer Kraft.

lock directory

Mit diesem Parameter legen Sie einen Pfad für das Verzeichnis fest, in das Samba seine freigegebene Speicherdatei, Statusdatei, Browse-Liste, WINS-Datenbank (wenn WINS-Unterstützung aktiviert ist) und Lock-Dateien

schreibt, die für die Implementierung des Parameters `max connections` verwendet werden. Der Parameter `max connections` wird in Kapitel 7 dargestellt, in dem Sie erfahren, wie Sie Samba für die Freigabe von Verzeichnissen konfigurieren. Der Zweck der Parameter besteht darin, die Anzahl der Benutzer einzuschränken, die sich gleichzeitig mit einer Freigabe verbinden können.

Während der Kompilierung wird normalerweise das Lock-Verzeichnis `/usr/local/samba/var/locks` als Standard festgelegt:

```
Standard:    lock directory = während der Kompilierung festgelegt
```

In der Praxis sollten Sie die Standardeinstellung für das Lock-Verzeichnis z.B. dann ändern, wenn Sie mehreren Servern die Benutzung der gleichen Samba-Binärdateien ermöglichen möchten, indem Sie Tools auf einem NFS-gemounteten Dateisystem bereitstellen. Viele Unternehmen mounten solch ein Dateisystem in `/usr/local/`, um netzwerkspezifische Tools und Utilities freizugeben. Zwar können Sie Binärdateien zwischen Samba-Servern gemeinsam nutzen, aber es ist nicht möglich, ein Lock-Verzeichnis freizugeben. Daher sollten Sie für jeden Server ein lokales Verzeichnis festlegen, in das Samba die notwendigen Dateien platzieren kann.

```
lock directory = /var/spool/locks/samba
```

name resolve order

Der Parameter `name resolve order` entspricht der Datei `/etc/nsswitch.conf` auf Plattformen wie Linux, Solaris und IRIX. Mit diesem Parameter können Sie die Reihenfolge festlegen, in der versucht wird, Namen aufzulösen. Der Parameterwert ist eine durch Leerstellen getrennte Liste, für die vier Einträge zulässig sind. Tabelle 5.4 listet die möglichen Werte und Besonderheiten auf.

Wert	Beschreibung
`lmhosts`	Die Samba-Datei `lmhosts` wird auf eine Entsprechung des verlangten Namens durchsucht.
`hosts`	Dieser Wert weist Samba an, eine Standard-Hostname-/IP-Adresse-Auflösung durchzuführen und dafür die auf dem System zur Verfügung stehenden Mittel zu benutzen, z.B. Durchsuchen von `/etc/hosts`, DNS-Anfragen oder NIS/NIS+-Entsprechungen. Bedenken Sie, dass diese Methode nur benutzt wird, wenn der aufzulösende NetBIOS-Name die Server-Ressourcenkennung (`<20>`) hat.
`wins`	Ist über die Parameter `wins server` oder `wins support` (siehe Kapitel 18, »WINS«) ein WINS-Server definiert, kann der NetBIOS-Name über eine Anfrage an den WINS-Server aufgelöst werden.

Tabelle 5.4: Zugelassene Einträge für den Parameter name resolve order

Tabelle 5.4: Zugelassene Einträge für den Parameter name resolve order (Fortsetzung)

Wert	Beschreibung
`bcast`	Führt die normale NetBIOS-Namensauflösung per Broadcast durch, die voraussetzt, dass sich der in Frage kommende Host im gleichen Broadcast-Subnetz befindet (oder es vielleicht einen WINS-Proxyserver gibt).

Standardmäßig wird zuerst die lokale `lmhosts`-Datei durchsucht. Eine `lmhosts`-Datei ist die NetBIOS-Entsprechung zur Unix-Datei `/etc/hosts`. Danach versucht Samba, über Standardmethoden wie das Durchsuchen von `/etc/hosts` oder Anfragen an den DNS den Namen aufzulösen. Waren diese beiden Methoden erfolglos, kontaktiert der Server einen WINS-Server, falls in `smb.conf` einer spezifiziert wurde. Bleibt auch dies erfolglos, versucht Samba, den Namen über Broadcast-Anfragen aufzulösen.

```
Standard:    name resolve order = lmhosts hosts wins bcast
```

Mit folgender Einstellung verwendet Samba für die Namensauflösung keine Broadcasts:

```
name resolve order = lmhosts wins hosts
```

deadtime

Über diesen Parameter können Sie die Anzahl von Minuten festlegen, die eine Verbindung (wie z.B. ein `smbd`-Prozess) inaktiv sein darf, bevor sie als terminiert angesehen und abgebrochen wird. Eine Verbindung gilt dann als inaktiv, wenn keine Aktivität erkannt wird und sie keine offenen Dateien enthält. Dies kann auf einem Server hilfreich sein, der viele Verbindungen handhabt, die nicht immer benutzt werden. Meine Benutzer haben die Angewohnheit, sich einzuloggen und niemals wieder auszuloggen, selbst wenn sie in den Urlaub gehen. Die meisten Clients haben eine Funktion, die die Verbindung automatisch wiederherstellt und diese Einstellung für den Benutzer transparent macht.

Die Standardeinstellung 0 bestimmt, dass die Verbindung niemals gekappt wird.

```
Standard:    dead time = 0
```

Auf den Servern an meinem Arbeitsplatz habe ich diesen Wert auf 15 Minuten eingestellt:

```
dead time = 15
```

smbrun

Dieser Parameter definiert den absoluten Pfad zur Binärdatei `smbrun`, einem kleinen Programm, das vom `smbd`-Daemon zur Ausführung von Shell-Befehlen benutzt wird. Wenn Sie Samba über den Standardbefehl `make install` installiert haben, sollten Sie diesen Parameter nicht gebrauchen. Haben Sie die Samba-Binaries aber manuell an einem anderen Standort installiert als dem durch `$prefix` in der Makefile definierten, müssen Sie diesen Parameter einrichten. Wenn `smbd` die `smbrun`-Binärdatei nicht finden kann, protokolliert es entsprechende Debug-Meldungen in der Datei `log.smb`. Der Standardwert wird durch die Variable `$prefix` in dem Makefile bestimmt.

```
Standard:    smbrun = während Kompilierung eingerichtet
```

Haben Sie das Tool in ein anderes Verzeichnis installiert, wie z.B. `/usr/bin`, müssen Sie die Pfadangabe entsprechend ändern:

```
smbrun = /usr/bin/smbrun
```

message command

Der Parameter `message command` bestimmt die Aktion, die Samba durchführt, wenn es eine WinPopup-artige Nachricht empfängt. Sie wissen von der Darstellung der NetBIOS-Namen in Kapitel 2, dass Namen mit der Ressourcenbezeichnung `<03>` den Messenger Server darstellen. Die WinPopup-Meldungen werden an diesen Namen gesendet. Abbildung 5.3 zeigt das WinPopup-Windows-95-Utility, das eine Nachricht an den Samba-Server mit dem Namen `BILBO` sendet.

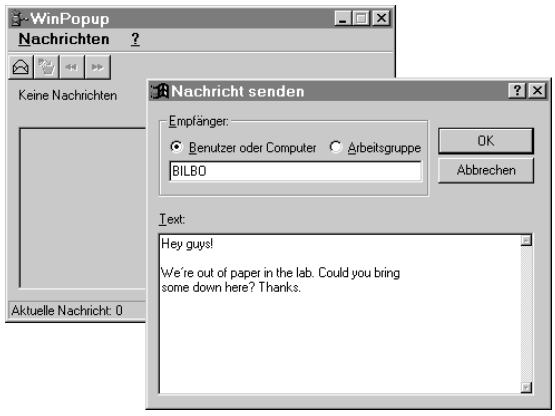


Abb. 5.3: Windows 95 OSR2 WinPopup.exe sendet und empfängt Nachrichten

Standardmäßig verwirft Samba WinPopup-artige Nachrichten.

```
Standard:    message command = none
```

Es gibt viele Möglichkeiten, die Nachricht zu senden. Ich habe das folgende Beispiel benutzt, um die Nachricht, die vom WinPopup-Client gesendet wurde (siehe Abbildung 5.3) auf meinem Linux-Rechner anzuzeigen (siehe Abbildung 5.4):

```
message command = /bin/bash -c '/usr/X11R6/bin/xterm -T ↵
"WinPopup Nachricht" \
-e /usr/bin/vim %s; rm %s' &
```

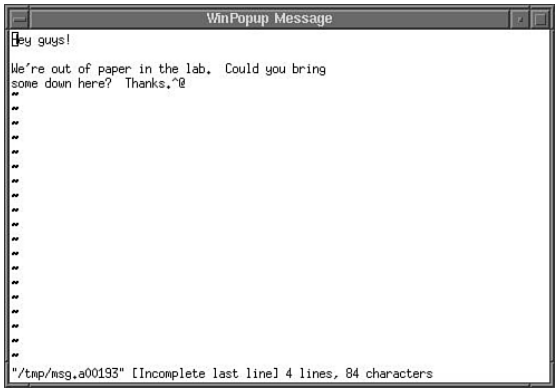


Abb. 5.4: Ausgeführtes message command bei Erhalt einer WinPopup-Nachricht

Sie könnten auch ein Befehlszeilen-Mail-Utilitiy wie z.B. `/bin/mailx` benutzen, um die Nachricht über SMTP zu übertragen.

Die WinPopup-Nachricht wird mit dem globalen `guest account` übertragen (in der Regel der Account `nobody`). Der Befehl kann neben den Standardmakros zusätzliche Variablen enthalten. Diese sind in Tabelle 5.5 aufgelistet.

Tabelle 5.5: Zusätzliche Variablen für den Parameter message command

Variable	Beschreibung
`%s`	Name der Datei, die den Textteil der Nachricht enthält.
`%t`	Name des Empfängers, an den die Nachricht gesendet wurde. Normalerweise ist dies der Name des Servers.
`%f`	Name des Clients, von dem die Nachricht stammt.

Es gibt einige Dinge, die Sie beim Einrichten des Parameters `message command` beachten sollten:

- ✘ Sie müssen absolute Pfade zu den Binaries benutzen, es sei denn, die von Ihnen verwendeten Befehle sind im Standardsuchpfad für die ausgeführte Shell enthalten.
- ✘ Sie müssen die erhaltene Nachricht explizit entfernen, sonst bleibt sie nach Ausführung des Befehls stehen.
- ✘ Das Programm sollte sofort wieder die Kontrolle an Samba zurückgeben, sonst kann der übertragende Client hängen, bis es zu einem Timeout kommt.

auto services

Dieser Parameter akzeptiert eine Liste aller Freigabenamen, die automatisch in der Browse-Liste für den Samba-Server sichtbar sein sollen. Dies ist wahrscheinlich in Bezug auf dynamisch erzeugte Dienste wie `[homes]` und `[printers]` am sinnvollsten. Die Standardeinstellung (keine Freigaben automatisch sichtbar) würde es nicht ermöglichen, die erweiterte Version dieser Freigaben zu sehen.

```
Standard:    auto services = none
```

Die folgende Einstellung lässt die Home-Verzeichnisse für die Benutzer `jerryc` und `peteh` in einer Browse-Liste auftauchen. Dieser Parameter gibt nicht mehr Zugriffsrechte auf die Dateien in der Freigabe als ein Benutzer normalerweise hat.

Nehmen wir an, dass `jerryc` und `peteh` Benutzernamen in der lokalen `/etc/passwd` sind und dass die `[homes]`-Freigabe korrekt definiert wurde. Diese Freigaben sind normalerweise nicht verfügbar, bis der Benutzer die Verbindung zum Server aufgebaut hat. Die folgende Beispieleinstellung führt dazu, dass die Freigaben in der Browse-Liste gezeigt werden, egal welcher Benutzer sich mit dem Server verbindet.

```
auto services = jerryc peteh
```

Das heißt jedoch nicht, dass Benutzer sich mit diesen bestimmten Freigaben verbinden können. Sie können lediglich sehen, dass die Freigaben auf dem bestimmten Server verfügbar sind.

protocol

Während der Verhandlungsphase beim Aufbau einer SMB-Verbindung sendet der Client eine Liste der Protokolldialekte, die er unterstützt. Der Server wählt daraus den höchsten aus, der ihm bekannt ist. Wenn Sie sich dies noch einmal ansehen wollen, blättern Sie zurück zu Kapitel 2.

Mit dem Parameter `protocol` können Sie den höchsten SMB-Dialekt spezifizieren, den Samba beherrscht. Normalerweise sollten Sie diese Option nicht verwenden, damit Samba die Protokollauswahl automatisch handhaben kann. Die Standardeinstellung ermöglicht `smbd`, den höchsten möglichen SMB-Dialekt, NT1, auszuwählen.

```
Standard:    protocol = NT1
```

In Tabelle 5.6 sind die zugelassenen Namen und eine kurze Beschreibung aufgelistet.

Tabelle 5.6: SMB-Dialekte

Name	Beschreibung
CORE	Die früheste Version von SMB, die keine Unterstützung für Benutzernamen bietet.
COREPLUS	Im Wesentlichen eine schnellere Version von CORE.
LANMAN1	Die erste moderne Version des Protokolls, die auch Unterstützung für lange Dateinamen bietet.
LANMAN2	Version 2 ist eine verbesserte Version des LANMAN1-Protokolls.
NT1	Dies ist die aktuellste Version des in Samba implementierten Protokolls, die auch vom Windows-NT-4.0-Service-Pack-3 verwendet wird. Die Version 2 dieses Protokolls wurde mit dem Service-Pack-4 für Windows NT freigegeben. Windows-NT-SP4-Clients arbeiten auch noch korrekt mit Samba mit der Version 1 des Protokolls.

time server

Ist dieser Parameter auf `true` gesetzt, kündigt `nmbd` sich als Zeitserver für Windows-Clients an und ermöglicht Ihnen daher, folgenden Befehl auf einem Windows-Client über die (MS-DOS-)Eingabeaufforderung auszuführen und die entsprechenden Ergebnisse zu erhalten:

```
C:\WINDOWS> net time
Aktuelle Zeit auf \\BILBO ist 1-27-1999 9:39 P.M.
Der Befehl wurde erfolgreich ausgeführt.
```

Auch wenn Sie diesen Parameter nicht einrichten, können Sie einen bestimmten Server immer nach der aktuellen Zeit befragen, indem Sie folgenden Befehl ausführen:

```
C:\WINDOWS> net time \\<Servername>
Standard:    time server = no
```

Standardmäßig antwortet Samba nicht auf `time-server`-Anfragen.

```
Standard:    time server = no
```

5.4 Zusammenfassung

Zwar gibt es relativ viele `smb.conf`-Parameter, aber Sie müssen nur die einrichten, die Sie verwenden wollen, bzw. die, die Sie explizit definieren wollen, wenn Sie genau so übervorsichtig sind wie ich. Das folgende Beispiel ist wahrscheinlich die einfachste funktionierende `smb.conf`, die ich mir vorstellen kann. Sie implementiert einen einfachen Home-Verzeichnis-Server:

```
[global]
   workgroup = MYGROUP
[homes]
   writeable = yes
```

Es liegt an Ihnen und den Bedürfnissen Ihres Netzwerks, wie ausführlich Sie Ihre `smb.conf` gestalten.

5.5 Frage & Antwort

Wenn ich die Samba-Konfigurationsdatei ändere, muss ich dann die beiden Samba-Daemons beenden und neu starten?

Für die meisten Änderungen an der Konfiguration müssen Sie nichts tun. Samba überprüft regelmäßig, ob die Konfigurationsdatei geändert wurde. Wenn ja, wird sie neu geladen. Aber es gibt hier einige Ausnahmen. Erstens, wenn Sie die Definition einer Freigabe ändern, können aktuell damit verbundene Benutzer die Änderungen erst sehen, wenn Sie die Verbindung beendet und die Freigabe neu gemountet haben. Zweitens, einige Änderungen erfordern, dass Samba neu gestartet wird, z.B. die Parameter `netbios name` oder `workgroup`.

Kann Samba mehr als einer Arbeitsgruppe gleichzeitig angehören?

Nein. Samba kann nur Mitglied einer Arbeitsgruppe sein.

Tag 6

Sicherheitsmodi und Passwörter

Als ich nach meiner Woche Urlaub in Kapitel 4, »Installation und Testen der Konfiguration«, an meinen Arbeitsplatz zurückkam, nahm meine Chefin mich beiseite. »Diese Netzwerklaufwerksdinger, die du eingerichtet hast, sind großartig. Die Produktivität ist in ungeahnte Höhen geschnellt. Ich würde diese Methode gern unternehmensweit einsetzen, aber bevor ich sie dem Management empfehlen kann, brauche ich einige genaue Fakten zur Sicherheit des Ganzen. Kannst du mir erklären, wie Samba mein Passwort überprüft, wenn ich mich einlogge?«

Einen Moment stand ich da und dachte nach. Dann sagte ich: »Ich setze mich gerne hin und erkläre, wie das alles abläuft, aber erst brauche ich eine Stunde, um mir einen Kaffee zu holen und einige Dinge zu erledigen.«

»Hört sich gut an«, sagte meine Chefin. »Ich sehe dich dann in einer Stunde in meinem Büro.«

Ich ging den Flur hinunter zum Testlabor und versuchte mich daran zu erinnern, wo ich meine Lieblingstasse und meine Kopie dieses Buches gelassen hatte.

Konnte ich die Informationen rechtzeitig finden? Würde Samba unternehmensweit eingesetzt werden?

Bleiben Sie dabei und finden Sie es heraus!

Im vorangegangenen Kapitel habe ich den allgemeinen Aufbau der Samba-Konfigurationsdatei und einige der generellen und verschiedenen `[global]-Parameter` dargestellt. In diesem Kapitel geht es um die Parameter, die für die Authentifizierungsmethoden relevant sind, wenn sich Clients mit

dem Samba-Server verbinden. Außerdem werde ich Ihnen etwas über Passwortsicherheit, Verschlüsselung und die Benutzung der IP-Adresse eines Clients für die Entscheidung, ob die Verbindung überhaupt authentifiziert wird, erzählen.

6.1 Sicherheitsmodi und der Parameter security

Das SMB-Protokoll bietet zwei grundsätzliche Modi für die Authentifizierung von Verbindungen. Der von Samba benutzte Modus wird durch die Einstellungen für den Parameter `security` im Abschnitt `[global]` der `smb.conf` definiert.

Der Eintrag für den Parameter `security` in der `smb.conf`-Manpage listet vier mögliche Eingaben auf, die verwendet werden können. Ich sprach aber von zwei grundlegenden Methoden, die das SMB-Protokoll für die Authentifizierung bietet. In der Realität sind von den vier Modi, die Samba unterstützt – `share`, `user`, `server` und `domain` – nur `share` und `user` fundamental unterschiedlich und stellen damit die zwei SMB-Sicherheitsmodi dar. Die anderen von Samba unterstützten Werte sind Variationen des User-Sicherheitsmodus:

```
security = [share|user|server|domain]
```

Vor Version 2.0 war die Standardeinstellung für den Sicherheitsparameter `share`. Mit der Version 2.0 wurde diese Einstellung auf den User-Modus geändert.

Mit Samba 2.0 wurde ein Passwortdatenbank-API eingeführt, über das Entwickler durch Definition einer Sammlung von Funktionen verschiedene Authentifizierungsmethoden einbetten können. Das heißt, dass Sie mehrere Methoden zur Auswahl haben, wenn Sie entscheiden, wie Sie die Informationen zu Ihren Bentuzer-Accounts speichern wollen.

Abbildung 6.1 zeigt die möglichen Backends, die derzeit benutzt werden oder sich in der Entwicklungsphase befinden. Der Client verlangt eine Verbindung zum Server, und der Server kontaktiert die Account-Datenbank über ein definiertes Interface. Es ist nicht unbedingt notwendig zu wissen, welches Backend benutzt wird. Was Samba betrifft, bietet die Datenbank die benötigten Benutzerinformationen.

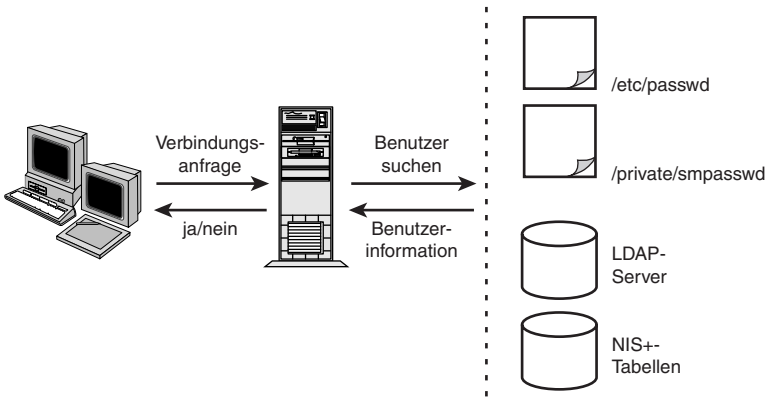


Abb. 6.1: Samba ermöglicht mehrere, voneinander unabhängige Benutzer-Account-Datenbanken

Derzeit wird experimentelle Unterstützung für den Zugriff auf NIS+-Tabellen und einen *Lightweight-Directory-Access-Protocol-(LDAP-)*Server entwickelt. Von den in Abbildung 6.1 dargestellten Möglichkeiten werden derzeit die Samba-`private/smbpasswd`-Datei und die Standard-Unix-`/etc/passwd`-Datei unterstützt. Beide Methoden werden später in diesem Kapitel ausführlich dargestellt. Jetzt sollten Sie nur wissen, dass beide Methoden für die Authentifizierung einen Benutzernamen und ein Passwort verlangen.

6.1.1 security = share

Im Share-Modus (Freigabeebene) sendet der Client während der Verbindungsanfrage ein Passwort. Ein zugehöriger Benutzername ist nicht erforderlich. Dies unterscheidet sich etwas von der Beschreibung, die ich in Kapitel 2, »Windows-Netzwerke«, gegeben habe, die eher auf den User-Modus zutrifft. Abbildung 6.2 verdeutlicht die zwei Schritte, die im Share-Modus für eine Verbindung zu einem SMB-Server benutzt werden.

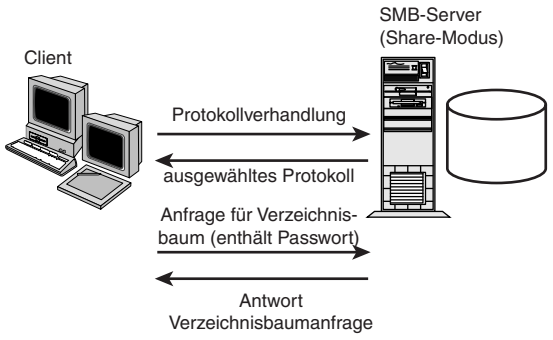


Abb. 6.2: Während der Verbindungsanfrage wird im Share-Modus das Passwort übertragen

Möglicherweise kennen Sie bereits ein Beispiel für einen SMB-Server im Share-Modus. Der Share-Modus ist die Standardeinstellung für einen Datei- oder Drucker-Server unter Windows 95 (siehe Abbildung 6.3). Abbildung 6.4 zeigt das Dialogfeld in der Netzwerkfreigabe, über das Sie den Sicherheitsmodus ändern können.

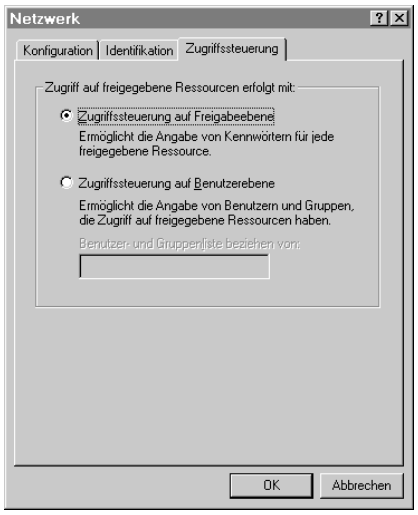


Abb. 6.3: Zugriffssteuerung auf Freigabeebene (Share-Modus) unter Windows 95

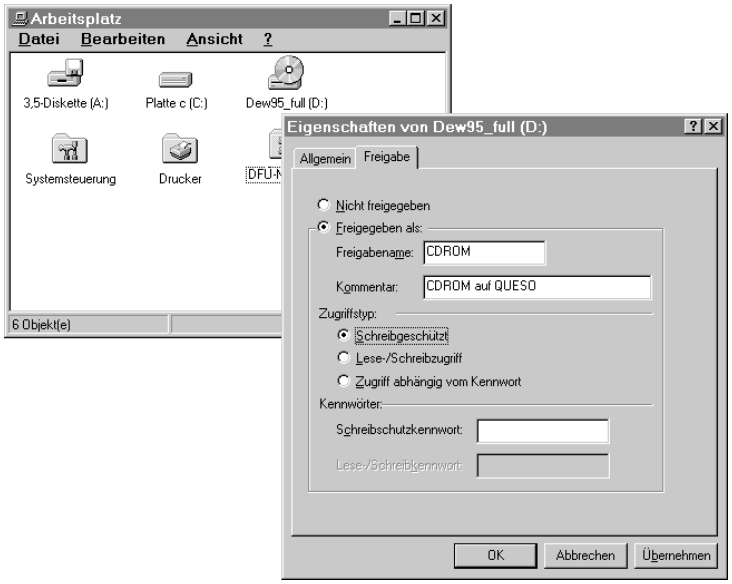


Abb. 6.4: Die Sicherheitsebene unter Windows 95 wählen

Vielleicht denken Sie mittlerweile: »Die Zugriffskontrolle auf Freigabeebene scheint dem Authentifizierungsmodell Benutzername/Passwort unter Unix zu widersprechen.« Sie haben Recht. Das Konzept des Share-Modus funktioniert in einer Multiuser-Umgebung wie Unix nicht gut, aber Samba versucht weitestgehend, das Unix-Sicherheitsmodell nicht zu beeinträchtigen.

Obwohl der Client erwartet, dass der SMB-Server im Share-Modus jeder Freigabe ein Passwort zuordnet, benutzt Samba das Standard-Unix-Benutzername/Passwort-Schema. Trotz der Tatsache, dass Clients der Verbindungsanfrage an einen Server im Share-Modus ein Passwort beifügen, übertragen viele Clients außerdem eine Sitzungsanfrage, die einen Benutzernamen enthält. Samba fügt diesen Namen einer Namensliste hinzu und versucht, ihn über das übertragene Passwort zu authentifizieren. Sie können andere Benutzernamen spezifizieren, die in die Liste aufgenommen werden sollen, indem Sie für die Freigabe den Parameter `user` definieren:

```
user = jerryc, smbguest, jdoe
```

Samba versucht die Verbindung zu authentifizieren, indem es jeden Benutzernamen mit dem Passwort vergleicht, bis eine Entsprechung gefunden wird oder auch nicht; dann wird die Verbindung zur Freigabe abgelehnt. Da Samba sowieso immer versucht, eine Kombination aus Benutzername und Passwort zu authentifizieren, wird die Zugriffskontrolle im Share-Modus nicht empfohlen. Es ist generell besser, eine Form der Authentifizierung auf Benutzerebene, die ich im nächsten Abschnitt darstelle, zu verwenden.

6.1.2 security = user

In Kapitel 2 (im Abschnitt »Protokollüberblick«) versucht ein Client, sich mit einem Server im User-Modus zu verbinden. Werden unverschlüsselte Passwörter benutzt, überträgt der Client während der Sitzungsaufnahme einen Benutzernamen und ein Passwort.

Da es in diesem Buch um die Benutzung von Samba und nicht um die Entwicklung eines SMB-Servers geht, werde ich Sie nicht sehr oft mit Paketausgaben langweilen. Aber ich denke, es ist recht nützlich, die Account-Informationen zu sehen, die während dieser Phase der Verbindung übertragen werden. Nachfolgend finden Sie eine Sitzungsanfrage von einem Windows-95-OSR2-Client, der sich mit einem Samba-Server verbindet. Die Pakete wurden über eine SMB-aktivierte Version von `tcpdump` abgefangen:

```
C:\\WINDOWS> net use h: \\bilbo\boss
Der Befehl wurde erfolgreich ausgeführt.
```

`tcpdump` ist ein Netzwerkpaket-Sniffer, der mit Source-Code verteilt wird. Die SMB-aktivierte Version können Sie sich unter `http://samba.org` herunterladen. Weitere Informationen über Paket-Sniffer finden Sie in Kapitel 11, »Troubleshooting«.

Nach Ausführung des Befehls generiert `tcpdump` folgende Ausgabe:

```
SMB PACKET: SMBsesssetupX (REQUEST)
SMB Command = 0x73
Error class = 0x0
Error code  = 0
Flags1      = 0x10
Flags2      = 0x0
Tree ID     = 0
Proc ID     = 28754
UID         = 1
MID         = 3586
Word Count  = 13
Com2=0x75
Res1=0x0
Off2=125
MaxBuffer=2920
MaxMpx=50
VcNumber=0
SessionKey}0xBE
CaseInsensitivePasswordLength=
[000] 54 45 53 54 50 41 53 53  00 00 00 00 00 00 42 4F  ⏎
   TESTPASS ......BO
[010] 53 53 00 00 00 00 00 00  42 4F 53 53 00 43 48 49  ⏎
   SS...... BOSS.CHI
[020] 50 53 4E 44 49 50 53 00  57 69 6E 64 6F 77 73 20  ⏎
   PSNDIPS. Windows
[030] 34 2E 30 00 57 69 6E 64  6F 77 73 20 34 2E 30 00  ⏎
   4.0. Windows 4.0
```

Sie können erkennen, dass das Passwort `testpass` und der Benutzername `boss` während der Sitzungsanfrage übertragen werden. Wenn Sie genau hinsehen, werden Sie auch feststellen, dass der Benutzername und das Passwort in Großbuchstaben umgewandelt werden. Dies kann ärgerlich sein und wird im Abschnitt »Passwortverschlüsselung« später in diesem Kapitel dargestellt.

Im User-Modus akzeptiert Samba die übermittelte Kombination aus Benutzername und Passwort und versucht, diese unter Benutzung seiner Account-

Datenbank zu authentifizieren. Dieser Prozess ist unabhängig vom Backend des Benutzer-Accounts (z.B. verschlüsselte Passwörter, LDAP und `/etc/passwd`) immer gleich, obwohl der Beweis der Identität auch ein abgeleiteter Wert statt des eigentlichen Passworts selbst sein kann. Bitte beachten Sie die Hinweise zur SMB-Challenge/Response-Verschlüsselung im Abschnitt »Passwortverschlüsselung« später in diesem Kapitel. Ist die Authentifizierung der Sitzungsanfrage erfolgreich, braucht der Client während nachfolgender Verbindungsanfragen keine Benutzer-Account-Informationen zu übertragen.

In Abbildung 6.5 sind die drei Schritte für eine Verbindung zu einem SMB-Server im User-Modus dargestellt. Zunächst wird der Protokolldialekt ausgewählt, dann eine Sitzung zwischen dem Client und dem Server aufgebaut und schließlich die Verbindung zur Ressource konfiguriert.

6.1.3 security = server

Der Server-Modus von Samba ist im Prinzip eine Variante des User-Modus. Samba teilt dem Client mit, dass es sich im User-Modus befindet, und der Client führt eine normale Sitzungsanfrage durch. Samba nimmt dann die Informationen und sendet eine Sitzungsanfrage an den Rechner, der als Passwort-Server fungiert. Befindet sich der Passwort-Server im User-Modus und akzeptiert die Sitzungsanfrage, akzeptiert Samba die ursprüngliche Sitzungsanfrage des Clients.

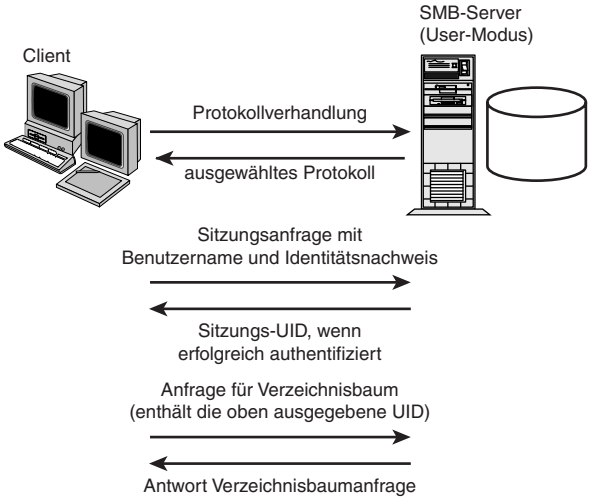


Abb. 6.5: Verbindungsanfrage im User-Modus

Abbildung 6.6 illustriert diesen Prozess. Die chronologische Reihenfolge des Diagramms ist von oben nach unten. Wenn Sie den Pfeilen folgen, sehen Sie, dass an dem Punkt, an dem der Client eine Sitzung mit dem Server verlangt, dieser eine Sitzungsanfrage an den Passwort-Server sendet. Erst wenn der Server eine Antwort vom Passwort-Server erhalten hat, wird die Anfrage des Clients akzeptiert oder abgelehnt.

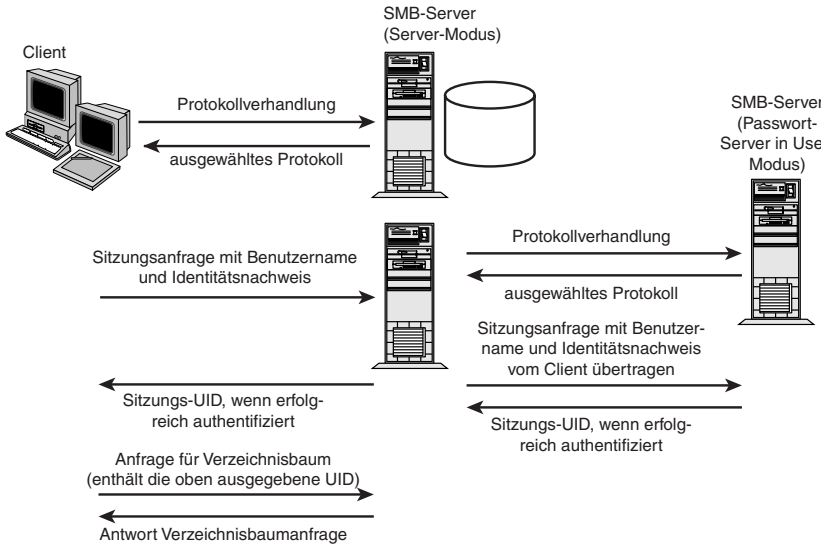


Abb. 6.6: Ein Client, der sich mit einem Samba-Server im Server-Modus verbindet

Der Parameter `password server` hat folgende Syntax:

```
password server = NetBIOS-Name des SMB-Servers
```

Sie können mehrere NetBIOS-Namen auflisten, z.B.

```
password server = DOMAINPDC DOMAINBDC1 DOMAINBDC2
```

Mit dieser Einstellung kann Samba versuchen, jedem aufgelisteten Server nacheinander eine Sitzungsanfrage zu senden, bis ein Server antwortet. Das heißt, der nächste Rechner in der Liste wird nur dann kontaktiert, wenn der vorstehende Rechner nicht erreichbar war. Es heißt nicht, dass Samba versucht, die anderen aufgelisteten Rechner zu kontaktieren, wenn die Verbindungsanfrage an den ersten Rechner scheitert.

Sie müssen den NetBIOS-Namen des Passwort-Servers benutzen (nicht die IP-Adresse), und Samba muss eine Möglichkeit haben, den Namen in eine IP-Adresse aufzulösen, um eine Verbindungsaufnahme zu versuchen.

Sie können jeden SMB-Server im User-Modus als Passwort-Server verwenden, aber die Sicherheit Ihres Samba-Servers entspricht dann nur der des augewählten Passwort-Servers. Ich habe Sie gewarnt! Übliche Wahlen für einen Passwort-Server sind Ihr Windows NT Primary Domain Controller (PDC) oder ein anderer Samba-Rechner.

Der Server-Modus hat eine Besonderheit. Nachdem Samba die Sitzungsanfrage für den Client gewährt hat, muss es eine Methode haben, eine Unix-UID für den Benutzer zu bekommen, um den Zugriff auf Dateien kontrollieren zu können. Das heißt, dass zwar keine lokalen Accounts für die Authentifizierung der Verbindung verwendet werden, aber der Benutzer muss eine UID auf dem lokalen Server haben. Es gibt zwei mögliche Lösungen für dieses Problem:

- ✘ Sie können einen lokalen Account für alle Benutzer einrichten, die auf den Samba-Server zugreifen und einfach das Passwortfeld in der `/etc/passwd` (oder der Datei, in der die Passwörter gespeichert sind) deaktivieren. Normalerweise erreichen Sie dies, indem Sie das Sternchen (*) in das Feld `password` setzen.
- ✘ Sie können eine der Methoden benutzen, über die Benutzernamen innerhalb einer Freigabe zugeordnet werden, z.B. `force user`, die in Kapitel 7, »Dateifreigaben«, beschrieben werden. Oder verwenden Sie den Parameter `username map`, den ich später in diesem Kapitel darstelle.

Abbildung 6.7 sieht Abbildung 6.6 sehr ähnlich, da beide Abbildungen den gleichen Prozess darstellen. Abbildung 6.7 wurde aber um den Punkt erweitert, an dem Samba versucht, eine gültige Unix-UID für den in der Sitzungsanfrage angegebenen Benutzernamen zu erhalten. In einem gewissen Sinne transparent ist, dass die Suche nach dem Benutzernamen durch den Parameter `username map` gefiltert werden kann, bevor der Name tatsächlich in der `/etc/passwd` gesucht wird.

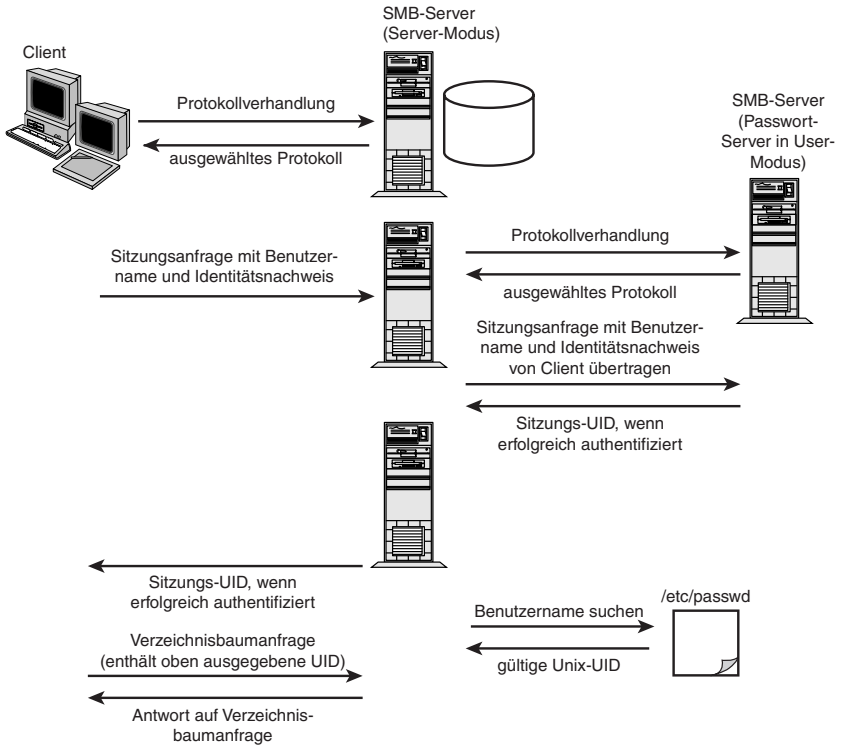


Abb. 6.7: Einem authentifizierten Benutzernamen wird eine Unix-UID zugeordnet

6.1.4 security = domain

Der Domain-Modus von Samba entspricht im Wesentlichen dem Konzept des Server-Modus, mit der Ausnahme, dass der Samba-Server Mitglied einer Windows-NT-Domäne wird. Das heißt, der Samba-Server kann an Dingen wie vertrauten Beziehungen teilnehmen. Es gibt einige weitere Vorteile für die Benutzung von `security = domain` statt `security = server`. Diese Darstellung möchte ich auf Kapitel 12, »Fallstudie: Einen NT-Datei- und Drucker-Server ersetzen«, verschieben, in dem ich beschreibe, wie ein Windows-NT-Datei- und Drucker-Server durch einen Samba-Rechner im Domain-Modus ersetzt wird. Bis dahin betrachten Sie bitte die zwei Modi als äquivalent.

6.2 Benutzernamen und Passwörter

Jetzt wissen Sie, wie `smbd`-Verbindungen authentifiziert werden, und können sich auf die Details von Benutzernamen und Passwörtern konzentrieren.

6.2.1 Username Level

Wie Sie bereits in der Paketausgabe von einer Sitzungsanfrage gesehen haben, übertragen einige Clients den Benutzernamen komplett in Großbuchstaben. Standardmäßig versucht Samba, diesen Benutzernamen klein geschrieben zu suchen und dann nur mit dem ersten Buchstaben groß geschrieben, z.B. `boss` und `Boss`. Wenn Sie einen seltsamen Unix-Benutzernamen haben, wie z.B. `BobAcct`, der Bob in der Abteilung *Accounting* zugewiesen ist, kann Samba mit dieser Methode den Benutzernamen nicht finden.

Darum gibt es einen Parameter, über den die maximale Anzahl von Großbuchstaben in dem Benutzernamen bestimmt werden kann. Samba versucht dann, über eine Brute-Force-Methode den Benutzernamen zu finden, indem es alle Abwandlungen mit Großbuchstaben von 1 bis zum definierten Wert ausprobiert.

Stellen Sie den Parameter `username level` auf `4` ein und wenden Sie diese Einstellung auf Bobs Account-Namen an:

```
username level = 4
```

Sie können voraussetzen, dass als Benutzername während der Sitzungsanfrage `BOBACCT` übertragen wird. Samba versucht, folgende Namen in der Systempasswortdatei (oder anderen benutzten Backends) zu finden:

```
bobacct
Bobacct
bObacct
boBacct
bobAcct
bobaCct
bobacCt
bobaccT
Bobacct
BoBacct
BobAcct
```

Die Suche wird beendet, sobald der Benutzername gefunden ist. Je höher der Wert für `username level`, um so mehr Kombinationen von Groß-/

Kleinbuchstaben werden ausprobiert und um so länger dauert es, bis die Suche erfolgreich oder auch nicht abgeschlossen ist. Sind alle Unix-Account-Namen im Standardformat klein geschrieben, wird dieser Parameter unnötig.

6.2.2 Username Map

Eines der Hauptprobleme bei der Integration von Unix- und PC-Betriebssystemen besteht in der Synchronisierung der Informationen zu den Benutzer-Accounts. Einige Unix-Varianten lassen nur die Benutzung von acht Zeichen oder weniger für den Benutzernamen zu, während einige Windows-Clients eine beliebige Zeichenkette mit Leerzeichen erlauben. Oft finden sich Administratoren mit der Aufgabe beschäftigt, zwei bereits etablierte Systeme mit existierenden Account-Namen zu integrieren. Mit dem Parameter `username map` können Sie eine Datei spezifizieren, die den während einer Verbindungsanfrage übertragenen Benutzernamen einem lokalen Benutzernamen zuordnet.

Diese Option ist standardmäßig nicht aktiviert.

```
Standard:    username map = none
```

Um Zuordnungen zu verwenden, müssen Sie den Standort der Datei spezifizieren, in der die Zuordnungen enthalten sind:

```
Beispiel: username map = /usr/local/samba/lib/users.map
```

Jeder Eintrag in der Datei sieht so aus:

Unix Benutzername = *Client-Benutzername ...*

Wenn Sie z.B. den Benutzernamen `Administrator` oder `Admin` dem Account `sysadmin` zuordnen wollen, definieren Sie folgenden Eintrag:

```
sysadmin = Administrator Admin
```

Wenn ein Benutzer versucht, sich als `Administrator` mit einer Freigabe zu verbinden, muss er das Passwort für den Account `sysadmin` übermitteln.

Die Zuordnung wirkt sich auf alle Beispiele des Client-Benutzernamens aus, mit Ausnahme der Sitzungsaufnahme zu einem Passwort-Server, wenn `securitiy = server` gesetzt ist. Um bei obigem Beispiel zu bleiben: Wenn ein Benutzer versucht, sich mit dem Home-Verzeichnis von `Administrator` zu verbinden, würde er sich tatsächlich mit `\\server\sysadmin` verbinden.

Es ist möglich, Unix-Gruppen einem einzelnen Account zuzuordnen. Diese Zeile ordnet jeden Benutzer in der Gruppe `staff` dem Account `staffsmb` zu:

```
staffsmb = @staff
```

Es steht außerdem eine Wildcard zur Verfügung, über die Sie jeden vom Client übertragenen Namen zuordnen können. Dieser Eintrag ordnet alle Benutzer dem Account `guest` zu:

```
guest = *
```

In Hinsicht auf Abbildungen gibt es einen Punkt, den Sie beachten sollten: `smbd` analysiert die Datei Zeile für Zeile und führt eventuelle Abbildungen bis zum Ende der Datei durch. Dies kann zu mehrfachen Zuordnungen führen, von `Benutzername` zu `neuer_Benutzername1` zu `neuer_Benutzername2`. Wenn Sie die Analyse der Datei stoppen wollen, nachdem eine Zuordnung erfolgt ist, sollten Sie der Zeile ein Ausrufungszeichen (`!`) voranstellen. Wird keine Zuordnung in der Datei gefunden, benutzt Samba den ursprünglichen Benutzernamen.

6.2.3 Password Level

Werden für die Authentifizierung Klartextpasswörter benutzt, tauchen ähnliche Probleme auf wie die, die ich im Abschnitt über Benutzernamen in Bezug auf die Groß-/Kleinschreibung erwähnt habe. Dieser Ausschnitt aus der vorher gezeigten `tcpdump`-Ausgabe erinnert Sie daran, dass das Passwort, `testpass`, in Großbuchstaben übertragen wird:

```
[000] 54 45 53 54 50 41 53 53  00 00 00 00 00 00 42 4F  ↵
   TESTPASS ......BO
[010] 53 53 00 00 00 00 00 00  42 4F 53 53 00 43 48 49  ↵
   SS...... BOSS.CHI
[020] 50 53 4E 44 49 50 53 00  57 69 6E 64 6F 77 73 20  ↵
   PSNDIPS. Windows
[030] 34 2E 30 00 57 69 6E 64  6F 77 73 20 34 2E 30 00  ↵
   4.0. Windows 4.0
```

Der Parameter `password level` hat in etwa die gleiche Funktion wie der Parameter `username level`. Der Unterschied liegt darin, dass standardmäßig zwei Möglichkeiten für die Verwendung des Passworts ausprobiert werden: so, wie es vom Client übertragen wird, und komplett klein geschrieben.

Wie der Parameter `username level` nimmt auch `password level` als Wert eine ganze Zahl an, die die maximale Anzahl von Großbuchstaben definiert, die für das Passwort zugelassen sind. Samba versucht dann, den Benutzer-

namen zu authentifizieren, indem es Abwandlungen der Großbuchstaben im Passwort benutzt.

Je größer der Wert und je mehr Kombinationen Samba ausprobiert, um so länger dauert die Authentifizierungsphase. Sie müssen bestimmen, was für Ihren Server akzeptabel ist. Ein `password level = 8` bedeutet auf den meisten Systemen, dass beim Passwort nicht mehr zwischen Groß- und Kleinschreibung unterschieden wird. Ich habe festgestellt, dass die Einstellung 4 in der Regel akzeptabel ist und nicht allzu viele Umstände für existierende Passwörter macht. Es ist jedoch auch hilfreich, die Richtlinien für die Benutzung von Passwörtern dahingehend zu ändern, dass nicht mehr als vier Großbuchstaben verwendet werden dürfen.

6.2.4 Passwortverschlüsselung

Samba unterstützt sowohl die LanManager- als auch die Windows-NT-SMB-Passwort-Verschlüsselungsalgorithmen. Das heißt, Samba kann Benutzer auf die gleiche Art und Weise authentifizieren wie Microsoft-Server es können.

Wenn Sie mit der Verschlüsselung von Passwörtern unter Unix vertraut sind, erscheinen Ihnen einige Punkte vielleicht ähnlich. Z.B. sind die LanMan- und NT-Passwort-Hashwerte unwiderruflich, ebenso wie Unix-Passwörter, die in `/etc/passwd` (oder `/etc/shadow`) gespeichert sind. Unwiderruflich heißt, dass Sie nur dann feststellen können, ob ein Benutzer das korrekte Passwort eingegeben hat, wenn Sie das eingegebene Passwort verschlüsseln und diesen Wert mit der verschlüsselten Version vergleichen, die auf der Festplatte gespeichert ist. Es gibt keine Möglichkeit, einen LanMan/NT-Hash zu entschlüsseln außer über Brute-Force-Methoden, wie z.B. einen Wörterbuchangriff.

Sie sollten einen großen Unterschied zwischen den Unix- und LanMan/NT-Verschlüsselungsalgorithmen beachten. Der Algorithmus zur Erzeugung eines LanManager- oder NT-Passwort-Hashwerts produziert bei gleicher Eingabe immer das gleiche Resultat. Das heißt, wenn Sie das Passwort `testpass` zweihundertmal verschlüsseln, ist das verschlüsselte Passwort immer gleich. Dieser Prozess erzeugt, was als *Klartextentsprechung* bekannt ist.

Ich hoffe, dass das folgende Beispiel dies klarmacht. Sie können den Prozess in Abbildung 6.8 verfolgen.

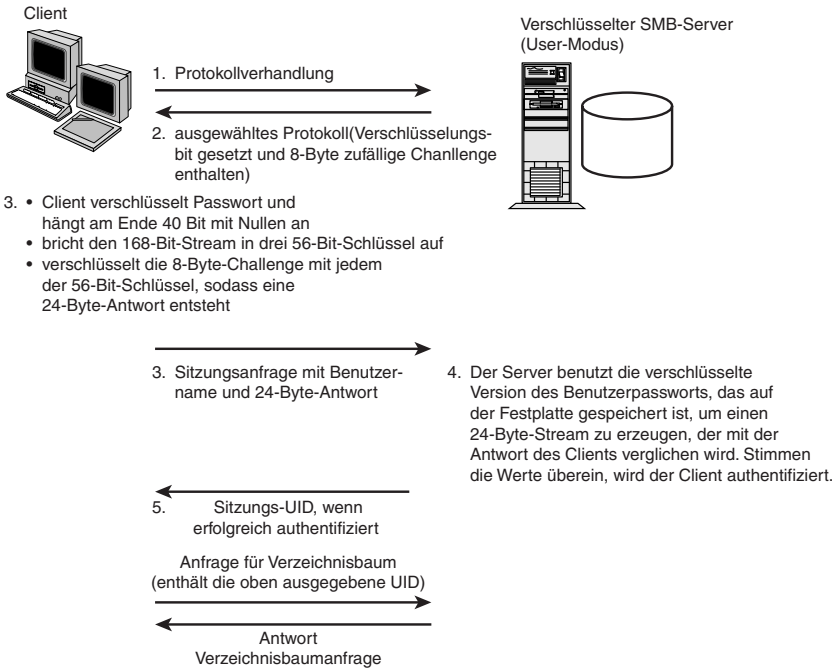


Abb. 6.8: Beispiel für eine Challenge/Response-Authentifizierung zwischen einem Client und einem Server

1. Der Client sendet eine Anfrage für eine Protokollverhandlung an den Server.
2. Unterstützt der Server verschlüsselte Passwörter, wird das entsprechende Bit im Response-Paket übertragen, und der Server fügt dem Paket eine 8-Bit-Challenge hinzu. Diese Aufforderung wird zufällig erzeugt und ist für jeden Client verschieden.
3. Abbildung 6.9 illustriert die Generierung der Client-Antwort. Der Client benutzt das verschlüsselte Passwort, das dem verhandelten Protokolldialekt entspricht (entweder LanMan oder NT) und hängt fünf Null-Bytes an (dies erzeugt einen 168-Bit-Stream), um drei verschiedene 56-Bit-DES-Schlüssel zu erzeugen, die dann jeweils für die Verschlüsselung der 8-Byte-Challenge benutzt werden. Die drei 8-Byte-Resultate werden zusammengefasst und bilden die 24 Byte lange Antwort, die an den Server übertragen wird.

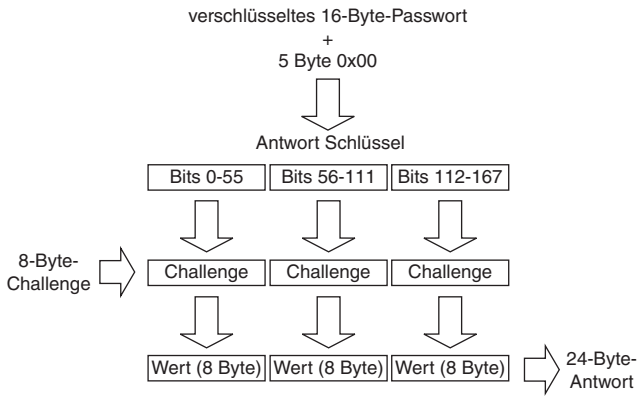


Abb. 6.9: Die 24 Byte lange Antwort generieren

4. Der Server führt dann unter Benutzung der verschlüsselten Version des Benutzerpasswords, die auf der Festplatte gespeichert ist, die gleichen Schritte durch. Das Resultat wird mit dem vom Client übertragenen Wert verglichen, um zu überprüfen, ob der Client das korrekte Passwort benutzt hat.

5. Stimmen der 24-Byte-Wert des Servers und die vom Client übermittelte Antwort überein, wird die Sitzungsanfrage (oder Freigabeverbindung im Falle des Share-Modus) akzeptiert. Stimmen sie nicht überein, hat der Client nicht das korrekte Passwort übertragen.

Machen Sie sich keine Gedanken, wenn Sie den Prozess nicht Wort für Wort wiederholen können. Ich habe ihn hier nur dargestellt, um einen Punkt zu beweisen. Das Passwort des Benutzers wird niemals über das Netzwerk übertragen. Das sorgt für erhöhte Sicherheit. Es werden nur die Daten übertragen, die aus dem Passwort generiert wurden.

Kommen wir nun zurück zu meinem vorangegangenen Kommentar über Klartextentsprechungen von Passwörtern. Der Server muss das verschlüsselte Passwort irgendwo speichern, damit er den 24-Byte-Wert generieren kann, um die Antwort des Clients zu authentifizieren. Denken Sie daran, dass das Passwort immer auf den gleichen Wert verschlüsselt wird. Wenn also jemand die verschlüsselte Version des Passworts kennt, kann diese Person an dem vorher beschriebenen Prozess teilnehmen, ohne das Passwort jemals kennen zu müssen!

Sind das zu viele Informationen auf einmal? Vielleicht können einige dieser Punkte helfen, sich für verschlüsselte oder Klartextpasswörter zu entscheiden:

✘ Mit Klartextpasswörtern kann Samba die gleiche Passwortdatenbank (d.h. die `/etc/passwd`) benutzen wie andere Unix-Dienste, z.B. login

und FTP. Diese Dienste übermitteln Passwörter oft auch in Klartext über das Netzwerk. Samba überträgt also keine Benutzer-Account-Informationen, die nicht sowieso schon über das Netzwerk geschickt werden.

- ✘ Wenn Sie Klartextpasswörter benutzen, brauchen Sie nichts anderes als normale Unix-Systemdateien, die auf der Festplatte gespeichert werden.
- ✘ Windows NT ab SP3 mag keine Klartextpasswörter, und Sie können keinen Server browsen, der Verschlüsselung nicht unterstützt. NT fordert außerdem die Eingabe eines Passworts, wenn Sie sich mit nicht verschlüsselten Freigaben verbinden wollen, was bei häufigen Freigabeverbindungen extrem ärgerlich werden kann.
- ✘ Die Abstimmung zwischen `smbpasswd` und `unix passwd` kann schwierig sein. Weitere Informationen hierzu finden Sie in Kapitel 16, »Passwortsynchronisierung«.
- ✘ Verschlüsselte Passwörter können nicht von jemandem gelesen werden, der Zugriff auf die zwischen Client und Server übertragenen Pakete hat. Wenn Sie Klartextpasswörter benutzen, können die übertragenen Informationen über einfache Netzwerkanalyse-Tools wie `tcpdump` eingesehen werden.

Wenn Sie sich für verschlüsselte Passwörter entscheiden, die standardmäßig nicht aktiviert sind, können Sie die Funktion über folgende Einstellung in Ihrer `smb.conf` aktivieren:

```
encrypt passwords = yes
```

Haben Sie die Verschlüsselung von Passwörtern aktiviert, müssen Sie nun eine zweite Benuter-Account-Datei im Auge behalten. In dieser Datei, die normalerweise `smbpasswd` genannt wird und sich in einem Unterverzeichnis namens `private` unterhalb des Samba-Installationsverzeichnisses befindet, speichert Samba die LanMan- und NT-Hashwerte der Benutzerpasswörter. Das Format ist dem von `/etc/passwd` sehr ähnlich:

```
username:uis:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX:
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX:account: flags:lastset:
```

Die Felder `username` und `uid` erklären sich von selbst. Die nächsten zwei Felder enthalten die zwei 16-Byte-Hashwerte des Benutzernamens, die von LanMan bzw. NT generiert wurden. Das Feld `account flags` bestimmt den Typ des Accounts wie z.B. Benutzer-Account oder Rechner-Account (Rechner-Accounts werden in Anhang A, »Experimentelle PDC-Unterstützung«, näher dargestellt). Das Feld `lastset` zeichnet den Zeitpunkt der letzten Passwortänderung auf.

Hier ein Beispieleintrag:

```
jerryc:1009:AAD3B435B51404EEAAD3B435B51404EE: ↵
     31D6CFE0D16AE931B73C59D7E0C089C0:[U      ]:LCT-36918AD9:
```

Wollen Sie die Datei, die die verschlüsselten Passwörter enthält, an einem anderen Ort speichern oder sie umbenennen, können Sie die neuen Werte über den Parameter `smb passwd file` definieren. Der Wert sollte ein absoluter Pfad zur SMB-Passwortdatei wie der folgende sein:

```
smb passwd file = /etc/smbpasswd
```

Das Erzeugen der ursprünglichen SMB-`passwd`-Datei und das Einrichten der Passwörter kann eine extrem erschreckende Aufgabe sein, wenn Sie viele existierende Unix-Accounts haben.

Es gibt hier zwei übliche Lösungen. Beide verlangen, dass Sie zunächst einen ersten `smbpasswd`-Eintrag für jeden Benutzer einrichten. Dies kann ganz einfach gemacht werden, wenn Sie eines der Skripte benutzen, die in der Samba-Distribution enthalten sind:

```
cat /etc/password | mksmbpasswd.sh >
/usr/local/samba/private/smbpasswd
```

Erhält der Unix-Rechner Account-Informationen von NIS oder NIS+, können Sie den oben stehenden Befehl `cat` je nach System entweder durch `ypcat` oder `niscat` ersetzen. Das Shellskript `mksmbpasswd.sh` befindet sich im Unterverzeichnis `source/script` der Samba-Distribution. Die resultierende `smbpasswd`-Datei enthält alle Benutzer aus `/etc/passwd` mit ihren LanMan- und NT-Hash-Passwortwerten, die auf 32 X eingestellt sind. Samba authentifiziert einen Benutzer, dessen Passworteintrag auf diesen Wert eingestellt ist, nicht.

Wollen Sie den Wert auf ein leeres Passwort setzen, müssen Sie

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX:
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

durch

```
NO PASSWORDXXXXXXXXXXXXXXXXXXXX:
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

ersetzen.

Geben Sie als `root` den folgenden Befehl aus:

```
/usr/local/samba/bin/smbpasswd -n Benutzername
```

Ersetzen Sie *`Benutzername`* durch den entsprechenden Benutzernamen. Samba speichert die verschlüsselten Passwörter in einer Datei namens

smbpasswd und fügt ein Utility hinzu, das ebenfalls smbpasswd heißt, um die Einträge in der Datei zu manipulieren. Lassen Sie sich durch den Namen nicht verwirren.

Alternativ können Sie die smbpasswd-Datei manuell über einen Texteditor bearbeiten und die Eingabe selber ändern. Wenn Sie die smbpasswd-Datei tatsächlich manuell bearbeiten, sollten Sie jedoch sicherstellen, dass die LanMan- und NT-Passwortfelder 32 Zeichen enthalten, nicht mehr und nicht weniger. Haben die Felder nicht genau 32 Zeichen, kann Samba diesen Benutzer niemals authentifizieren.

Nachdem Sie den smbpasswd-Eintrag geändert haben, müssen Sie den folgenden Parameter null passwords im Abschnitt [global] der smb.conf auf yes setzen:

```
null passwords = yes
```

Nach Erzeugen der smbpasswd-Datei bleibt die Frage: »Wie fülle ich das Passwortfeld für jeden Eintrag?«

Lösung 1

Wenn Sie Samba derzeit mit Klartextpasswörtern benutzen, können Sie die verschlüsselten Passwortfelder nach und nach für jeden Benutzer füllen, indem Sie den booleschen Parameter update encrypted benutzen. Der Standardwert für diesen Parameter ist no. Um die Unterstützung zu aktivieren, müssen Sie im Abschnitt [global] der smb.conf folgenden Eintrag einfügen:

```
update encrypted = yes
```

Wenn Sie den Wert dieses Parameters auf yes setzen, müssen Sie sicherstellen, dass der Parameter encrypt passwords auf no gesetzt ist.

```
encrypt passwords = no
```

Ist der Parameter update encrypted auf yes gesetzt, schreibt Samba jedes Mal, wenn ein Benutzer erfolgreich eine Sitzungsaufnahme verlangt, die verschlüsselte Version des Klartextpassworts, das für diesen Benutzer übermittelt wurde. Die einzige Voraussetzung ist, dass der Benutzer einen gültigen Eintrag in der vorhandenen smbpasswd-Datei hat. Was immer der vorherige Wert des passwd-Felds war, es ist jetzt auf das aktuelle Passwort des Benutzers eingestellt. Natürlich hat diese Methode nur im User-Modus Sinn.

Mit dieser Lösung kann der Samba-Server einige Tage oder Wochen, wie lange auch immer er braucht, laufen, Passwörter abfangen und die smbpasswd-Datei ausfüllen. Enthält smbpasswd genügend Einträge, können Sie

einfach die folgenden Parameter in `smb.conf` ändern und auf die Benutzung verschlüsselter Passwörter umschalten:

```
encrypt passwords = yes
update encrypted = no
```

Die meisten Ihrer Benutzer werden niemals erfahren, dass sich etwas geändert hat.

Abbildung 6.10 verdeutlicht, wie der Parameter `update encrypted` funktioniert. Zunächst wählen der Client und der Server den Protokolldialekt, dann sendet der Client in seiner Sitzungsanfrage den Benutzernamen und das Passwort in Klartext. Kann der Benutzer erfolgreich über `/etc/passwd` authentifiziert werden, verschlüsselt der `smbd` das Passwort und schreibt die Informationen in die `smbpasswd`-Datei. Sie sollten wissen, dass der Eintrag des Benutzers in der `smbpasswd` an diesem Punkt des Authentifizierungsprozesses niemals benutzt wird. Das verschlüsselte Passwort wird hier nur aufbewahrt.

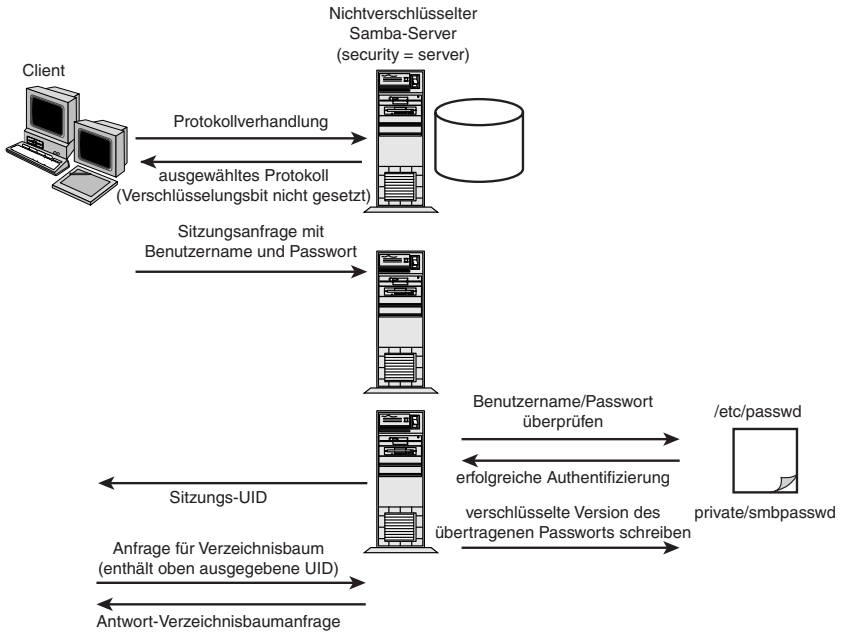


Abb. 6.10: Über den Parameter update encrypted wird smbpasswd nach und nach gefüllt

Lösung 2

Ich habe vorher schon erwähnt, dass Samba ein Utility namens `smbpasswd` enthält, mit dem Sie Einträge in der `smbpasswd`-Datei manipulieren können. Dieses Tool finden Sie im Unterverzeichnis `/bin`. Es ist die Entsprechung des Unix-Programms `/bin/passwd`.

Wenn ein Benutzer einen neuen Unix-Account erhält, weisen die meisten Unternehmen ein zufälliges Passwort zu und zeigen dann dem Benutzer, wie er das Passwort ändern kann, damit es persönlicher oder leichter zu merken ist. Wenn Sie eine neue Samba-Infrastruktur aufbauen – das Wort könnte doch glatt aus einem Dilbert-Comic stammen, oder? –, könnten Sie Benutzern ganz einfach gleichzeitig mit ihrem neuen Unix-Account auch ein SMB-Passwort zuweisen. Zusammen mit den Standardanweisungen für die Änderung ihres Unix-Passworts über `/bin/passwd` könnten Sie Ihren Benutzern gleich die Anweisungen für die Änderung ihres SMB-Passworts über den Befehl `/usr/local/samba/bin/smbpasswd` mitgeben. Dies ist sicher die einfachste Lösung, da so die Verantwortung an den Benutzer übergeben wird, die Passwörter synchron zu halten, wenn dies gewünscht ist. Dies könnte aber je nach Kaliber Ihrer Benutzer zu vermehrten Anrufen bei den Supportleuten führen. Es ist sehr leicht durcheinander zu bringen, welches Passwort zu welchem Logon gehört, wenn die Accounts nicht mehr synchron sind. Ein Benutzer kann sehr unnachgiebig in seinem Glauben sein, dass er das korrekte Passwort für seinen Account eingegeben hat! Sie müssen selber entscheiden, welche Lösung für Sie die beste ist.

Nachfolgend finden Sie eine Beispielsitzung, in der ich mein SMB-Passwort über den Befehl `smbpasswd` ändere. Ich habe Kommentare in spitzen Klammern eingefügt, die Ihnen erklären, was eingegeben werden muss. (Sie haben doch nicht wirklich gedacht, dass ich Ihnen mein Passwort verrate, oder?)

```
[jerryc]@bilbo jerryc]408: /usr/local/samba/bin/smbpasswd
Old SMB password: <geben Sie das alte SMB-Passwort ein und ⏎
   drücken Sie Enter>
New SMB password: <geben Sie das neue SMB-Passwort ein und ⏎
   drücken Sie Enter>
Retype new SMB password: <geben Sie das neue SMB-Passwort ⏎
   noch einmal ein und drücken Sie Enter>
... Password changed for user jerryc
```

6.2.5 Windows-9x- und Windows-NT-Clients und verschlüsselte oder Klartextpasswörter

Ich möchte hier einen kurzen Hinweis zur Benutzung von Klartextpasswörtern und neueren Microsoft-Clients geben. Dieses und andere Themen, die für die Microsoft-32-Bit-Clients spezifisch sind, werden ausführlicher in Kapitel 14, »Windows 9x und Windows NT«, dargestellt.

Mit dem Service-Pack-3.0 für Windows NT 4.0 hat Microsoft den Standard geändert, so dass nur noch verschlüsselte Passwörter benutzt werden. Wenn Sie daher versuchen, sich mit einem nicht verschlüsselten Samba-Server zu verbinden, werden Sie folgende Fehlermeldung sehen:

```
Server ist nicht verfügbar. Mit diesem Konto kann mann sich
nicht von dieser Station aus anmelden.
```

Sie werden das gleiche Verhalten bei Windows-95-Clients feststellen, die das SMB Network Redirector Update (`vrdrupd.exe`) haben. Windows 95 wird Sie aber einfach weiterhin auffordern, ein Passwort einzugeben. Dieser Patch aktualisiert das System:

```
\windows\system\Vredir.vxd
\windows\system\Vnetsup.xvd
```

Es ist möglich, mit diesen Clients einen nicht verschlüsselten Samba-Server zu benutzen. Dafür müssen Sie nur einen Wert in der Windows-Systemregistrierung einstellen. Die Details für diese Lösung finden Sie in Kapitel 14.

6.3 Zugriffskontrollen

Samba bietet außer der Standardauthentifizierung über Benutzername/Passwort einige zusätzliche Optionen für die Kontrolle von Verbindungsanfragen. Über diese Optionen können Sie auf Basis der IP-Adresse des Clients die Verbindungen kontrollieren, was sehr hilfreich sein kann, wenn Ihr Netzwerk mit einem größeren LAN (oder dem Internet) verbunden ist.

hosts allow

Sie können den Parameter `hosts allow` benutzen, um eine Liste von Hosts zu definieren, die sich mit einer bestimmten Freigabe verbinden können. Wird der Parameter im Abschnitt `[global]` der `smb.conf` benutzt, gilt er unabhängig von den einzelnen Freigabeeinstellungen für alle Freigaben.

Der Parameter nimmt als Wert eine Liste von IP-Adressen in Dezimalschreibweise an, wobei die Adressen vollständige Adressen oder Subnetz-Netzwerkadressen sein können. So würde z.B. `192.168.1.73` einem be-

stimmten Host die Verbindung gestatten, während `192.168.1.` Verbindungen von jedem Host im Class-C-Subnetz `192.168.1.` zulassen würde. Sie können Hostnamen statt IP-Adressen benutzen, wenn Samba die Namen auflösen kann. Dies heißt gewöhnlich, dass Sie als Wert den *Fully Qualified Domain Name (FQDN)* angeben. Es ist ebenfalls möglich, über das Schlüsselwort `EXCEPT` Hosts auszuschließen. Standardmäßig werden Verbindungen von jeder IP-Adresse akzeptiert.

Hier sind einige Beispiele:

```
hosts allow = 192.168.1.73 queso.my.net 191.168. EXCEPT 191.168.2.
```

Diese Einstellung ermöglicht Verbindungen von zwei bestimmten Hosts, `192.168.1.73` und `queso.my.net`, sowie Verbindungen von jedem Host im Class-B-Subnetz `191.168.` außer denen, die sich im Class-C-Subnetz `191.168.2.` befinden.

Hier ist ein Beispiel, für das eine Kombination aus IP-Adresse und Subnetzmaske verwendet wird:

```
hosts allow = 192.168.1.32/255.255.255.224
```

Dies ermöglicht Verbindungen von Hosts im Bereich `192.168.1.33` bis `192.168.1.63`. Die Broadcast-Adresse für das Subnetz ist `192.168.1.64`.

hosts deny

Der Parameter `hosts deny` ist die Ergänzung zum Parameter `hosts allow`. Er bietet die gleiche Funktion wie das Schlüsselwort `EXCEPT` innerhalb des Wertes von `hosts allow`, aber zu einem höheren Grad. Die Syntax ist die gleiche wie die von `hosts allow`. Standardmäßig werden keine Verbindungen abgelehnt:

```
hosts deny = 192.168.3. 192.168.1.72
```

hosts equiv und user hosts

Die nächsten zwei Parameter erwähne ich nur der Vollständigkeit halber und empfehle Ihnen, sie nicht zu benutzen, weil beide Methoden Möglichkeiten bieten, über die sich Benutzer mit Freigaben verbinden können und ohne Passwort authentifiziert werden. Dies kann ein ernsthaftes Sicherheitsloch in Ihrem Server darstellen. Seien Sie vorsichtig!

Über den Parameter `hosts equiv` können Sie den Standort einer Datei festlegen, die eine Liste von Hosts oder Benutzern, einen pro Zeile, enthält, die auf einen Dienst zugreifen können soll, ohne ein Passwort angeben zu müssen. Hier ein Beispiel:

```
hosts equiv = /etc/hosts.equiv
```

Der boolesche Parameter `user hosts` veranlasst Samba, die Unix-Benutzerdatei `~/.rhosts` zu verwenden, um spezielle Hosts zu bestimmen, die ohne Angabe eines Passworts auf Freigaben zugreifen können. Wie beim Parameter `hosts equiv` ist auch diese Option standardmäßig nicht aktiviert. Wollen Sie sie aktivieren, müssen Sie folgende Einträge in den Abschnitt `[global]` der `smb.conf` einfügen:

```
use rhosts = yes
```

6.4 Verschiedenes

Die letzen zwei Parameter, die ich darstellen werde, haben einen Bezug zu Sicherheit, passen aber nicht zu den anderen bereits behandelten Themen.

map to guest

Ohne zu sehr ins Detail zu gehen, können Sie über den Parameter `map to guest` festlegen, was Samba tun soll, wenn eine Verbindungsanfrage ungültige Benutzerinformationen enthält (z.B. ein ungültiges Passwort). Es gibt drei mögliche Antworten:

- `Never` – Samba lehnt Verbindungsanfragen mit einem ungültigen Passwort ab. Dies ist die Standardeinstellung.
- `Bad User` – Wenn der Client ein ungültiges Passwort überträgt, wird die Verbindung abgelehnt, es sei denn, der Benutzername ist nicht bekannt. In diesem Fall wird die Verbindung akzeptiert, und der Benutzer wird in den `guest account` aufgenommen, der in `smb.conf` spezifiziert ist.
- `Bad Password` – Diese Einstellung führt dazu, dass Kombinationen aus falschem Benutzernamen und Passwort als Gastverbindungen akzeptiert werden. Der verbindende Benutzer merkt hiervon jedoch nichts und beschwert sich möglicherweise, dass er nicht auf seine Dateien zugreifen kann, weil er als `guest account` verbunden ist.

Ich empfehle Ihnen, die Standardeinstellungen bestehen zu lassen, es sei denn, Sie haben einen guten Grund, sie zu ändern. Wenn Ihnen selbst kein guter Grund einfällt, ist dies wahrscheinlich Grund genug, die Einstellungen in Ruhe zu lassen.

root directory

Dies ist ein weiterer Parameter, der nicht häufig benutzt wird. Er weist Samba an, ein `chroot()` zum angegebenen Verzeichnis auszuführen, ganz ähnlich wie es bei anonymen FTP-Verbindungen gehandhabt wird. Dies ist nicht unbedingt notwendig, da Samba standardmäßig Zugriff auf Dateien außerhalb der Freigabe ablehnt. Allerdings wird hiermit eine zusätzliche Sicherheitsebene eingefügt, aber sie müssen sicherstellen, dass sich alle notwendigen Skripte, Systemdateien und Binaries unterhalb des Root-Verzeichnisses befinden. Um das Standard-Root-Verzeichnis `/` zu überschreiben, können Sie ganz einfach das Verzeichnis Ihrer Wahl spezifizieren:

```
root directory = /export/smb
```

6.5 Abschließende Kommentare

Ich dachte, es wäre besser, mit diesem Punkt zu schließen, statt ihn innerhalb eines Kapitels zu vergraben. Wenn Sie durch eine Firewall von anderen Netzwerken getrennt sind und nicht wollen, dass Clients, die sich hinter der Firewall befinden, auf Ihre internen SMB-Server zugreifen können, sollten Sie die eingehenden Ports 137, 138 und 139 blockieren. Dies ist insbesondere dann wichtig, wenn Sie Benutzer haben, die gern ihre gesamte Festplatte freigeben, weil »es so praktisch ist«. Gerade Windows 95/98-Clients bieten dies per Voreinstellung an.

In Kapitel 7 kommen Sie zu den Details der Freigabenkonfiguration, so dass Ihre Benutzer tatsächlich auf ihre Dateien zugreifen können. Oh, ich habe vergessen, die kleine Geschichte zu beenden, die ich am Anfang dieses Kapitels begonnen habe, oder?

Nachdem ich etwas weniger als eine Stunde gebraucht hatte, trank ich den letzten Schluck von meinem Kaffee (der mittlerweile lauwarm geworden war) und machte mich auf den Weg zu meiner Chefin. Nachdem ich ihr meine Entscheidung für verschlüsselte Passwörter erklärt und einen Strategieplan – noch ein Dilbert-Wort – für die Änderung existierender Benutzerpasswörter in verschlüsselte Passwörter auf relativ harmlose Art und Weise definiert hatte, gratulierte sie mir zu einer weiteren gut erledigten Aufgabe. Dann unterschrieb sie einen Kaufvertrag für einen neuen Laptop, damit ich während meines von der Firma bezahlten Urlaubs mit ihr in Kontakt bleiben konnte. (Es könnte passieren!)

6.6 Zusammenfassung

Das SMB-Protokoll unterstützt zwei Modi für die Authentifizierung von Verbindungen. Samba unterstützt sowohl den Share-Modus (Freigabeebene) als auch den User-Modus (Benutzerebene). Zusätzlich bietet Samba zwei weitere Varianten der Authentifizierung auf Benutzerebene: Server-Modus und Domain-Modus.

Sie können für alle `security`-Optionen in Samba entweder Klartext- oder verschlüsselte Passwörter benutzen. Klartextpasswörter werden über die Standard-Unix-Account-Datenbank `/etc/passwd` (oder ihre Netzwerkentsprechung) authentifiziert. Die Passwortverschlüsselung dagegen verlangt, dass Samba eine separate Datei verwaltet, in der die Hashwerte der verschlüsselten Passwörter gespeichert werden.

6.7 Frage & Antwort

Brauche ich externe Libraries, um die Passwortverschlüsselung in Samba zu aktivieren?

Es ist richtig, dass der Administrator für ältere Versionen von Samba eine externe DES-Library besorgen musste, aber neuere Versionen von Samba benötigen diese nicht mehr. Der komplette benötigte Source-Code ist in der Samba-Distribution enthalten.

Kann ein Samba-Server so konfiguriert werden, dass er sowohl Klartext- als auch verschlüsselte Passwörter gleichzeitig unterstützen kann?

Nein. Ein einzelner Samba-Server kann nicht gleichzeitig Klartext- und verschlüsselte Passwörter für die Authentifizierung von Benutzern verwenden. Es gibt jedoch eine Methode über den Parameter `netbios aliases`, mit der Sie dies umgehen können. Die nötigen Funktionen für die Implementierung einer derartigen Lösung werden in Kapitel 10, »Automatisierung auf Server-Seite«, dargestellt.

Können einige Freigaben für den Share-Modus und andere auf dem gleichen Server für den User-Modus konfiguriert werden?

Nein. Der Samba-Parameter `security` ist eine `[global]`-Option.

6.8 Neue Begriffe

Klartextentsprechung eines Passworts – Diese wird generiert, wenn der benutzte Verschlüsselungsalgorithmus bei gleicher Eingabe immer den gleichen Byte-String erzeugt. Anders gesagt, ein Passwort wird immer zum gleichen Wert verschlüsselt. Kann ein Eindringling die verschlüsselte Version des Passworts abfangen, kann er erfolgreich am Challenge/Response-Authentifizierungsschema teilnehmen, das von SMB-Servern wie Samba und Windows NT benutzt wird.

Tag 7

Dateifreigaben

von Richard Sharpe

In den Kapiteln 5, »Die Datei smb.conf: Samba mitteilen, was es tun soll«, und 6, »Sicherheitsmodi und Passwörter«, haben Sie sich das grundlegende Format für die Parameter in `smb.conf`, die Samba-Sicherheitsmodi und die Handhabung der Passwörter angesehen. Samba wurde entwickelt, um Dateien auf verschiedenen Rechnern gemeinsam nutzen zu können, und genau das ist es, was wir uns jetzt näher ansehen werden.

Die Freigabe von Dateien ermöglicht Ihnen, Dateien von verschiedenen Rechnern aus gemeinsam zu benutzen (siehe Abbildung 7.1). Normalerweise verfügt ein Datei-Server über mehr Festplattenspeicher als ein Client (Windows für Workgroups, Windows 95/98, Windows NT usw.). An den Server können auch die meisten Drucker angeschlossen sein, aber das Drucken wird im nächsten Kapitel ausführlich beschrieben.

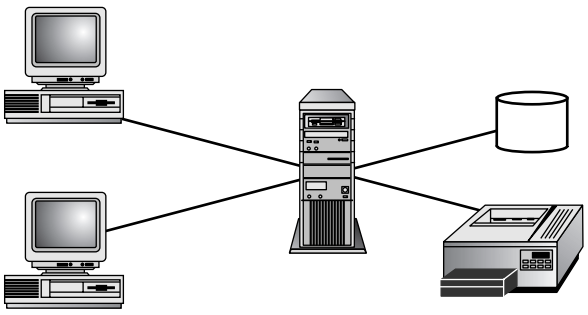


Abb. 7.1: Ein Datei-Server, der Dateien und Drucker freigibt.

In diesem Kapitel führe ich Sie durch alle notwendigen Schritte, damit Sie Dateifreigaben auf einem Samba-Server einrichten können. Sie werden außerdem die meisten Freigabe- und globalen Parameter kennenlernen, die für die Freigabe von und den Zugriff auf Dateien relevant sind.

Wenn Sie meinen Ausführungen Schritt für Schritt folgen, sollten Sie sicherstellen, dass Ihre Clients keine verschlüsselten Passwörter benutzen, da diese zu unnötigen Komplikationen führen können. Weitere Informationen zur Deaktivierung verschlüsselter Passwörter auf Ihren Clients finden Sie in den Kapiteln 6 und 14, »Windows 9x und Windows NT«. Sie sollten sich über den Account `boss` in Ihren Client-Rechner einloggen. Ist Ihr Samba-Server der erste SMB-Server in Ihrem Netzwerk, erhalten Sie möglicherweise während des Einloggens eine Fehlermeldung, die besagt, dass Sie nicht über einen Logon-Server authentifiziert werden konnten. Ignorieren Sie diese Meldung fürs Erste.

7.1 Eine smb.conf-Datei aufbauen

Bevor Sie Dateien freigeben können, brauchen Sie eine funktionierende `smb.conf`-Datei, die Samba benutzen kann. In Kapitel 4, »Installation und Testen der Konfiguration», haben Sie sich eine `smb.conf`-Datei angesehen, hier werden Sie eine nun von Beginn an aufbauen.

Wie ich bereits erwähnt habe, hat die Datei `smb.conf` einen globalen und einen Freigabeabschnitt. Im Folgenden benutzen Sie den untenstehenden globalen Abschnitt und fügen Abschnitte für Dateifreigaben hinzu, während Sie verschiedene Methoden für die Kontrolle und Verwaltung von Dateifreigaben kennenlernen:

```
[global]
   workgroup = FOWLPLAY
   netbios name = EAGLE
   server string = My first server
   guest account = pcguest
   security = user
   password level = 8
```

Als Erstes sollten Sie bei dieser `smb.conf`-Datei bemerken, dass sie keine Freigaben definiert, aber trotzdem funktioniert. Wie auch schon in Kapitel 5 ist in dieser `smb.conf`-Datei die Arbeitsgruppe, der Samba angehört, `FOWLPLAY` und der NetBIOS-Name des Servers, `EAGLE`.

Wenn Sie diese `smb.conf` auf Ihrem Server installieren und Samba neu starten (Sie müssen sich als `root` in Ihren Samba-Server einloggen, um das zu

tun), sollten Sie den neuen Server in der Netzwerkumgebung in Windows 9x oder Windows NT 4.0 (für Windows für Workgroups 3.11 benutzen Sie den Dateimanager und wählen *Laufwerke*, *Netzlaufwerk verbinden*) sehen können.

Wenn Sie Samba bereits benutzen, ersetzen Sie Ihre vorhandene `smb.conf`-Datei nicht einfach durch obiges Beispiel. Machen Sie zuerst eine Backup-Kopie.

Nachdem Sie Samba neu gestartet haben (siehe Kapitel 4 für Details darüber, wie Sie Samba auf verschiedenen Plattformen starten), sollten Sie unter Windows 9x oder Windows NT die Netzwerkumgebung sehen, wie sie in Abbildung 7.2 dargestellt ist.

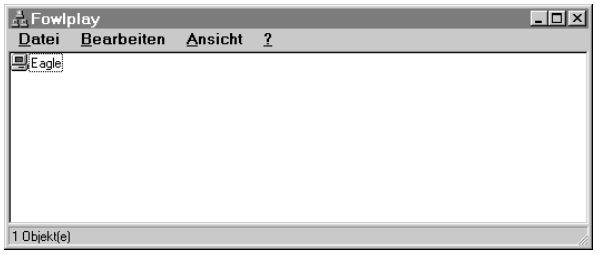


Abb. 7.2: Netzwerkumgebung für die Arbeitsgruppe FOWLPLAY

Hier sehen Sie, dass Ihr Server als `EAGLE` auftaucht (so haben Sie ihn in `smb.conf` genannt) und dass es sich um einen Samba-2.0.0Beta4-Server handelt. Wenn Sie den Parameter `netbios name` aus der vorstehenden `smb.conf` entfernen, wird Ihr Server mit einem Namen angezeigt, der aus den ersten Bestandteilen seines DNS-Namens besteht. Dies ist möglicherweise das, was Sie wollen, aber es wird nicht zu den Beispielen in diesem Kapitel passen.

Wenn Sie jetzt auf den oben stehenden Server doppelklicken, wird das Fenster in Abbildung 7.3 geöffnet.

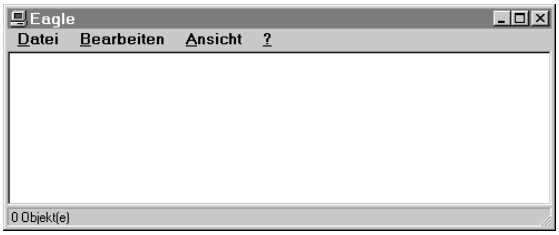


Abb. 7.3: Auflistung der Freigaben auf dem Dateiserver EAGLE

Damit wird bestätigt, dass Ihr Samba-Server keine Freigaben hat oder zumindest keine, die browsebar sind (dies werde ich später darstellen).

Ihr nächster Schritt besteht darin, eine Freigabe einzurichten und zu sehen, was sich ändert.

7.2 Eine Freigabe einrichten

Um eine Freigabe einzurichten, müssen Sie Ihrer `smb.conf`-Datei im Definitionsbereich für Freigaben einen entsprechenden Abschnitt hinzufügen. Da Samba Verzeichnisse und die darin liegenden Dateien freigibt, sollten Sie zunächst ein entsprechendes Verzeichnis, das freigegeben werden soll, auf Ihrem Server suchen oder erstellen.

Hier erstelle ich ein Verzeichnis mit dem Namen `/home/first-share` und weise Samba an, es freizugeben:

```
mkdir /home/first-share
```

Sie können natürlich auch einen anderen Namen benutzen, wenn Sie zu den abenteuerlustigen Menschen gehören, aber dann müssen Sie im Folgenden alle Pfadnamen entsprechend ändern, und die Beispiele in diesem Buch sehen möglicherweise anders aus als das, was Sie sehen.

Dann fügen Sie unserer oben stehenden `smb.conf`-Datei Folgendes hinzu:

```
[first-share]
   comment = Meine erste Freigabe
   path = /home/first-share
   browsable = yes
```

Nachdem Sie nun Ihrer `smb.conf` eine Freigabe hinzugefügt haben, starten Sie Samba neu. Sie sollten (nach einer Weile) die neue Freigabe in der Netzwerkumgebung sehen können, wie in Abbildung 7.4 dargestellt.

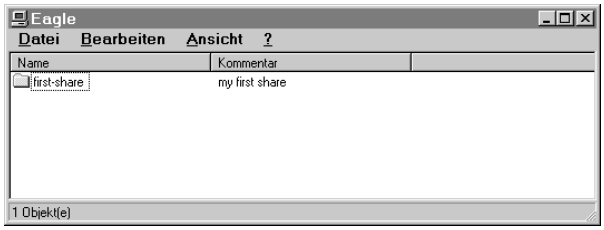


Abb. 7.4: Die Netzwerkumgebung zeigt Ihre erste Freigabe!

Großartig, nun können Sie eine Freigabe auf Ihrem Samba-Server sehen. Was können Sie noch tun? Nun, bevor Sie sich die Freigabe ansehen, sollten Sie erst einmal einige Dateien einfügen:

```
cat > /home/first-share/file-1.txt
Now is the time for all good men
To come to the aid of their country
^D
cat > /home/first-share/file-2.txt
The time has come the walrus said
To talk of many things
^D
todos /home/first-share/file-2.txt
```

Beachten Sie, dass Sie möglicherweise das `todos-Utility` nicht auf Ihrem Server haben. In diesem Fall sollten Sie sich eine Kopie besorgen oder das Perl-Skript (das sich auf der diesem Buch beiliegenden CD-ROM befindet) eingeben, da Sie es brauchen werden. Das Skript ist bei einigen Distributionen auch unter dem Namen `u2dos` oder ähnlich verfügbar.

Nachdem Sie einige Dateien in die Freigabe auf Ihrem Server gestellt haben, schauen Sie sich an, was Ihr Client Ihnen zeigt. Doppelklicken Sie auf `first-share` im Fenster *Netzwerkumgebung*, und Sie sehen Ihre Dateien, wie in Abbildung 7.5 dargestellt.

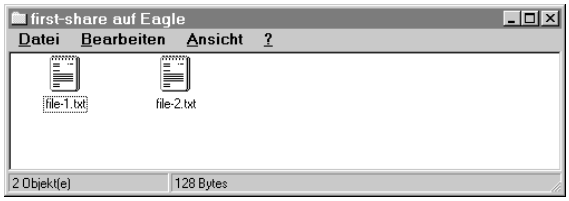


Abb. 7.5: Die ersten zwei Dateien in Ihrer Dateifreigabe

Sehen Sie sich nun die Dateien an. Wenn Sie auf `file-1-txt` doppelklicken, sollten Sie das in Abbildung 7.6 dargestellte Fenster sehen.

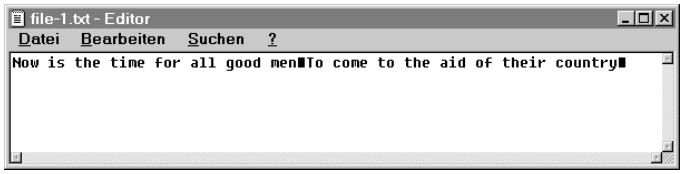


Abb. 7.6: file-1.txt enthält seltsame Zeichen

Schließen Sie dieses Fenster und doppelklicken Sie auf `file-2.txt`. Sie sollten genau das sehen, was in Abbildung 7.7 dargestellt ist.

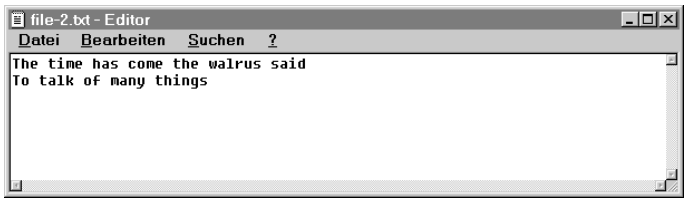


Abb. 7.7: file-2.txt sieht normaler aus

Warum werden diese zwei Dateien so unterschiedlich im Editor dargestellt? Das Problem liegt in der unterschiedlichen Art und Weise begründet, in der Windows und Unix das Ende einer Zeile in Textdateien speichern. Unter Unix wird das Zeilenende durch das Zeilenvorschubzeichen (NL) gekennzeichnet (oktal 012, hex 0x0A), während unter Windows (und DOS) das Zeilenende durch ein Wagenrücklaufzeichen oder CR (oktal 015, hex 0x0C) gefolgt von einem Zeilenvorschubzeichen gekennzeichnet ist. Als Sie `file-2.txt` erstellten, haben Sie die Datei explizit in eine Textdatei im DOS-Format konvertiert (über `todos`), während `file-2.txt` als Textdatei im Unix-Format belassen wurde.

Obwohl der Zweck jeder Zeile, die Sie in der `smb.conf` eingefügt haben, offensichtlich sein mag, möchte ich dennoch jede einzeln darstellen und erklären.

7.2.1 [first-share]

Diese Zeile führt einen neuen Abschnitt für die Freigabe `first-share` ein. Jede Freigabe wird als neuer Abschnitt eingeführt, mit ihrem Namen in eckigen Klammern.

7.2.2 comment

Diese Zeile bietet einen beschreibenden Kommentar für die Freigabe. Sie dient der Dokumentation der Freigabe in `smb.conf` und wird unter dem Begriff Kommentar in der Auflistung in der Netzwerkumgebung angezeigt.

7.2.3 path

Die Zeile teilt Samba mit, welchen Teil des Dateisystems es für Clients freigeben soll. Sie sollten immer ein vorhandenes Verzeichnis angeben oder eine Datei, die als symbolischer Link zu einem Verzeichnis dient (siehe nachfolgend).

Existiert das Verzeichnis nicht, können Windows-Clients seltsame Fehlermeldungen ausgeben, wenn sie versuchen, auf die Freigabe zuzugreifen. Ein Beispiel:

```
Zugriff auf \\EAGLE\first-share nicht möglich: Der
Netzwerkname kann entweder auf dem laufenden Netzwerk nicht
gefunden werden oder ist fehlerhaft.
```

7.2.4 browsable

Streng genommen brauchen Sie diesen Eintrag nicht zu definieren, da die Standardeinstellung `yes` ist. Sie müssen die Einstellung nur dann ändern, wenn Sie nicht wollen, dass Clients bestimmte Freigaben sehen können.

Um die Auswirkung der Einstellung für `browsable` zu sehen, ändern Sie sie in Ihrer `smb.conf` auf

```
browsable = no
```

und starten Samba neu. Gehen Sie dann zurück zur `FOWLPLAY`-Netzwerkumgebung und doppelklicken Sie auf den Server `EAGLE`. Können Sie nun die von Ihnen erzeugten Freigaben sehen, wenn sich das Fenster öffnet?

7.3 Zugriffsrechte

Sie haben nun eine Dateifreigabe eingerichtet und müssen Sie nutzbar machen. Browsen Sie z.B. die Dateifreigabe (nachdem Sie die Einstellung `browsable` in Ihrer `smb.conf` wieder aktiviert, Samba beendet und neu gestartet haben). Versuchen Sie dann, eine Datei in die Freigabe zu kopieren oder eine neue Datei bzw. ein neues Verzeichnis in der Freigabe zu erstellen. Wie Sie feststellen werden, ist dies nicht möglich. Sie werden nun untersuchen, warum das so ist und wie Sie dies ändern können.

Die Dateifreigabe eignet sich hervorragend für die gemeinsame Benutzung von Dateien, die von Benutzern mit ausreichenden Berechtigungen für das freigegebene Verzeichnis gestellt wurden, aber Client-Workstations können keine Dateien in der Freigabe erzeugen. Auch wenn Sie sich als `root` einloggen (der Superuser unter Unix), können Sie an diesem Punkt keine Dateien in der Freigabe erzeugen. Wie Sie bald sehen werden, ist die Freigabe standardmäßig nur mit Leseberechtigungen versehen.

Damit Clients auf die Dateifreigabe zugreifen können, müssen Sie Samba einige Dinge über die Dateifreigabe mitteilen. Zunächst sollten Sie jedoch einen Blick darauf werfen, wie Samba bestimmt, ob ein Client Zugriff auf Dateifreigaben erhält.

In Kapitel 2, »Windows-Netzwerke«, haben Sie sich die Art und Weise angesehen, in der ein CIFS/SMB-Client auf eine Ressource oder eine Dateifreigabe zugreift. Um Ihr Gedächtnis aufzufrischen, sind hier noch einmal die erforderlichen Schritte, obwohl einige davon ausgelassen werden können, aufgelistet:

1. Der Client verhandelt über den Protokolldialekt, um festzulegen, welche Variante des CIFS/SMB-Protokolls er unterstützt.
2. Der Client führt dann möglicherweise ein Login im Netzwerk oder auf dem Server durch, abhängig vom benutzten Protokolldialekt und der Präsenz von Logon-Servern. An diesem Punkt übermittelt der Client einen Benutzernamen und ein Passwort, aber dieser Schritt kann übergangen werden, insbesondere bei älteren Clients.
3. Der Client verlangt eine Verbindung zu einem Verzeichnisbaum oder einer Freigabe.

Um die Verbindungsanfrage des Clients zu einer Freigabe weiterzuverarbeiten, bestimmt Samba zunächst, ob die verlangte Freigabe existiert. Der folgende einfache Ansatz überprüft, ob dies der Fall ist:

1. Die Datei `smb.conf` wird nach einem Abschnitt durchsucht, der mit dem verlangten Freigabenamen übereinstimmt. Wird einer gefunden, wird dieser benutzt.
2. Wird die Freigabe nicht gefunden, überprüft Samba, ob sie einen `[homes]`-Abschnitt in der Datei `smb.conf` hat. Falls ja, wird die Datei `passwd` durchsucht, um festzustellen, ob der Freigabename einem Benutzernamen entspricht. Ist das so, wird ein Klon der `[homes]`-Freigabe erzeugt (wie später in diesem Kapitel ausführlich dargestellt) und die neue Freigabe verwendet.
3. Wird die Freigabe immer noch nicht gefunden, überprüft Samba, ob die `smb.conf` einen `[printer]`-Abschnitt enthält. Falls ja, wird festgestellt, ob die verlangte Freigabe einem Drucker in der Datei `printcap` entspricht. Ist das der Fall, wird ein Klon der `[printers]`-Freigabe erzeugt und diese geklonte Freigabe benutzt. Der Abschnitt `[printers]` wird ausführlich in Kapitel 8, »Drucker«, dargestellt.
4. Wird die Freigabe auch hier nicht gefunden, überprüft Samba, ob eine Standardfreigabe existiert, ändert in diesem Fall den Namen der Standardfreigabe in den entsprechenden verlangten Freigabenamen und benutzt diese Freigabe.
5. Wird keine Freigabe gefunden, gibt Samba eine Fehlermeldung über einen ungültigen Netzwerknamen an den Client zurück.

Danach durchläuft Samba die folgende Prozedur, um festzustellen, ob ein Client Zugriff auf die gefundene Freigabe erhält und als welcher Benutzer dieser Client zugreifen kann. Die Schritte werden nacheinander ausgeführt, und der Prozess wird mit dem ersten erfolgreichen Schritt beendet. Ist keiner der Schritte erfolgreich, wird der Zugriff zur Freigabe verweigert.

1. Wenn der Client einen Benutzernamen und ein Passwort überträgt, die authentifiziert werden können, wird der Zugriff zur Freigabe als authentifizierter Benutzer gewährt. Einige ältere Clients können ihre Benutzernamen über die Syntax `\\Server\Service%Benutzername` übertragen.
2. Hat der Client bereits einen gültigen Benutzernamen übermittelt und sendet nun ein korrektes Passwort (für die Freigabeanfrage), wird der Zugriff auf die Freigabe gewährt.
3. Der NetBIOS-Name des Clients und alle vorher verwendeten Benutzernamen werden über die Standardmechanismen des Betriebssystems (oder die Datei `smbpasswd`) mit dem übertragenen Passwort authentifiziert. Kann ein Benutzername erfolgreich authentifiziert werden, wird der Zugriff auf die Freigabe als authentifizierter Benutzer gewährt.
4. Wurden bereits vorher ein Benutzername und ein Passwort durch den Server (über ein `SessionSetupandX`) authentifiziert und hat der Client den Authentifizierungs-Token in der Freigabeanfrage übertragen, wird der Zugriff auf die Freigabe als Benutzername gewährt, der in dem Token definiert ist. Dieser Schritt wird übergangen, wenn für die Freigabe Revalidation spezifiziert ist (`revalidate = yes`).
5. Wenn für die Freigabe ein Feld `user =` definiert ist, der Client ein Passwort übertragen hat und die Kombination aus Benutzername, der für die Freigabe definiert wurde, und Passwort authentifiziert wurde, wird der Zugriff auf die Freigabe als definierter Benutzer gewährt.

Handelt es sich bei der Freigabe jedoch um eine `Guest-only`-Freigabe, wird der Zugang zur Freigabe als im Gast-Account definierter Benutzername gewährt, ohne dass einer der vorstehenden Schritte durchlaufen wird. Jedes übertragene Passwort wird ignoriert.

Wenn der als zugreifender Benutzer gewählte Benutzer sich in einer ungültigen Benutzerliste befindet (die später in diesem Kapitel beschrieben wird), wird die Verbindungsanfrage an diesem Punkt abgewiesen.

Über diese Prozedur kann Samba feststellen, unter welchem Account der Zugriff auf Dateien in der Freigabe gewährt wird. Der Zugriff auf Freigaben und die erlaubten Zugriffsberechtigungen auf Dateien werden jedoch noch von mehreren Parametern in der Datei `smb.conf` kontrolliert.

7.3.1 Parameter für den Zugriff auf Freigaben

Die folgenden Parameter sind auf die eine oder andere Weise relevant für den Zugriff auf Freigaben durch Clients. Die meisten Samba-Administratoren benutzen nicht viele dieser Parameter. Wie immer finden Sie die aktuelle Liste aller Parameter und die endgültige Darstellung ihrer Funktionen in den Manpages zu `smb.conf` für die aktuelle Version von Samba. Dort können Sie sich über den Befehl `man smb.conf` die Parameter genauer ansehen.

admin users

Dieser Freigabe-Parameter richtet die Benutzer ein, denen administrative Privilegien für die Freigabe gegeben werden. Wenn sie auf die Freigabe zugreifen, führen sie alle Dateioperationen als `root` aus.

Namen, die mit einem `@` beginnen, werden zunächst als NIS-Netzgruppe interpretiert und dann, wenn sie nicht in NIS gefunden werden, als Unix-Gruppen. Namen, die mit einem + beginnen, werden als Unix-Gruppen und Namen, die mit `&` beginnen, als NIS-Gruppen interpretiert.

Dieser Parameter kann sehr gefährlich sein, da jeder Benutzer in der `admin-users`-Liste alles tun kann, was er will, z.B. auch alle Dateien in der Freigabe löschen.

Standardmäßig gibt es keine administrativen Benutzer für eine Freigabe. Ein Beispiel:

```
admin users = root, fred
```

Diese Einstellung definiert `root` und `fred` als administrative Benutzer.

default service

Über diesen globalen Parameter wird der Name der Standardfreigabe festgelegt. Diese Standardfreigabe wird benutzt, wenn die von einem Client verlangte Freigabe nicht gefunden werden kann, und ihr Name wird in diesem Fall in den entsprechenden verlangten Namen umgeändert.

Normalerweise werden für die Standardfreigabe die Parameter `guest ok = yes` und `read only` eingerichtet.

Dieser Parameter hat keinen Standardwert. Ein Beispiel:

```
default service = lastchance
```

Mit dieser Einstellung wird `lastchance` zur Standardfreigabe.

guest account

Über diesen globalen Parameter wird der Name des Gast-Accounts definiert. Oft wird er auf `pcguest` eingestellt, und er muss in der auf dem Server benutzten Account-Datenbank eingetragen sein (z.B. der Datei `/etc/passwd`, NIS usw.). Normalerweise hat dieser Account kein gültiges Passwort, so dass sich niemand dort einloggen kann, weder von Unix noch von einem Client. Der Account kann nur benutzt werden, um den Zugriff auf Dateien zu kontrollieren.

Dieser Parameter kann im globalen Abschnitt und in einzelnen Freigabeabschnitten eingerichtet werden. Gast-Accounts, die in einem Freigabeabschnitt definiert sind, setzen globale Gast-Accounts außer Kraft.

Der Standardwert für diesen Parameter wird während der Kompilierung auf `nobody` gestellt. Ein Beispiel:

```
guest account = pcguest
```

Damit wird der Account mit dem Namen `pcguest` als Gast-Account definiert.

guest ok

Dieser Freigabe-Parameter bestimmt, ob der Zugriff zu einer Freigabe auch ohne Übertragung eines Benutzernamens und eines Passworts gewährt wird.

Wenn Clients einen Gastzugriff erhalten, greifen Sie auf die Dateien in der Freigabe als Gast-Account zu.

Ein Synonym für `guest ok` ist `public`.

Der Standardwert für diesen Parameter ist `no`. Ein Beispiel:

```
guest ok = yes
```

Damit kann über den Guest-Account auf die Freigabe zugegriffen werden.

guest only

Dieser Freigabe-Parameter bestimmt, dass nur Gastverbindungen zu einer bestimmten Freigabe erlaubt sind. Er muss in Verbindung mit `guest ok` oder `public` benutzt werden.

Der Standardwert für diesen Parameter ist `no`. Ein Beispiel:

```
guest only = yes
```

Damit wird festgelegt, dass auf diese Freigabe nur über den Guest-Account zugegriffen werden kann.

hosts allow

Dieser Parameter richtet die Liste der Hosts ein, die auf Freigaben zugreifen können. Wird er im Abschnitt `[global]` eingerichtet, bezieht er sich auf alle Freigaben, unabhängig von den Einstellungen für einzelne Freigaben. Wird der Parameter nicht im globalen Abschnitt verwendet, kann er für einzelne Freigaben definiert werden.

Hosts können über ihren Namen oder ihre IP-Adresse spezifiziert werden. Die vollständige Syntax für diesen Parameter ist die gleiche wie für die TCP-Wrappers-Datei `hosts_allow`. Weitere Details finden Sie in der Manpage (`man hosts_allow`).

Es gibt keinen Standardwert für diesen Parameter. Ein Beispiel:

```
hosts allow = 192.1.1. graham.goodies.com
```

Mit dieser Einstellung kann jeder Host im Subnetz `192.1.1.0/24` und das System mit dem Namen `graham.goodies.com` auf die Freigabe zugreifen.

hosts deny

Dieser Parameter ist die Umkehrung von `hosts allow`. Die für diesen Parameter aufgelisteten Hosts erhalten keinen Zugriff auf Freigaben, es sei denn, eine bestimmte Freigabe setzt die Liste mit einer eigenen Liste zugelassener Hosts außer Kraft. Gibt es Unstimmigkeiten zwischen den Parametern `hosts deny` und `hosts allow`, erhält `hosts allow` die Priorität.

Es gibt keinen Standardwert für diesen Parameter. Ein Beispiel:

```
hosts deny = 192.1.1. badhost.bad-company.com
```

Diese Einstellung verweigert jedem Host im Subnetz `192.1.1.0/24` und dem Host `badhost.bad-company.com` den Zugriff auf die Freigabe.

invalid users

Dieser Freigabe-Parameter spezifiziert eine Liste von Benutzern, die keinen Zugriff auf die Freigabe erhalten sollten. Der Parameter benutzt die gleiche Syntax wie der oben stehende Parameter `admin users`.

Es gibt keinen Standardwert für diesen Parameter. Ein Beispiel:

```
invalid users = root fred @bin
```

Mit dieser Einstellung wird den Benutzern `root` und `fred` sowie jedem Benutzer in der Gruppe `bin` der Zugriff auf die Freigabe verweigert.

max connections

Mit diesem Freigabe-Parameter wird die maximale Anzahl der Clients festgelegt, die sich mit der Freigabe verbinden können. Ist der hier spezifizierte Wert größer als 0, werden nach Erreichen der angegebenen Anzahl von Benutzern, die sich mit der Freigabe verbunden haben, keine Clients mehr zugelassen. Fällt die Anzahl der verbundenen Benutzer unter die angegebene Zahl, können sich wieder neue Benutzer bis zum definierten Maximum verbinden. Wird der Wert des Parameters auf `0` gesetzt, gibt es keine Einschränkung für die Anzahl der zugelassenen Verbindungen.

Dieser Parameter kann die Arbeitslast auf einem Samba-Server beschränken und bietet außerdem eine Methode für die Durchsetzung der Lizenzierungsbestimmungen, wenn Sie lizenzierte Software freigeben. Beachten Sie, dass Sie die Anzahl der verbundenen Benutzer einschränken, nicht die Anzahl der aktiven Benutzer. Ein Benutzer, der sich zu Beginn des Tages mit der Freigabe verbindet und den ganzen Tag verbunden bleibt, ohne jemals etwas damit zu tun, belegt auch eine dieser maximalen Verbindungen.

Der Standardwert für diesen Parameter ist `0`, der, wie vorher schon erwähnt, uneingeschränkte Verbindungen zur Freigabe ermöglicht. Ein Beispiel:

```
max connections = 100
```

Mit dieser Einstellung werden nicht mehr als 100 Verbindungen zur Freigabe zugelassen.

read list

Dieser Freigabe-Parameter definiert eine Liste von Benutzern, denen Nur-Lese-Zugriff auf die Freigabe gewährt wird. Das heißt, diese Benutzer erhalten keinen Schreibzugriff auf die Freigabe, auch wenn die Freigabe beschrieben werden kann.

Es gibt keinen Standardwert für diesen Parameter. Ein Beispiel:

```
read list = fred @guests
```

Diese Einstellung definiert, dass `fred` und alle Benutzer in der Gruppe `guests` Nur-Lese-Zugriff auf die Freigabe erhalten. Das heißt, sie können die Freigabe nicht beschreiben.

read only

Dieser Freigabe-Parameter ist `das` Gegenteil des Parameters `writable`. Wird er aktiviert, können Clients keine Freigabe beschreiben.

Der Standardwert für diesen Parameter ist `yes`, d.h., die Freigabe ist im Nur-Lese-Modus. Wenn Sie für eine Freigabe hier keinen Wert definieren, kann sie nur gelesen werden. Ein Beispiel:

```
read only = no
```

Mit dieser Einstellung kann die Freigabe beschrieben werden. Den gleichen Effekt können Sie auch über den Parameter `writable = yes` erzielen.

valid users

Dieser Freigabe-Parameter listet die Benutzer auf, denen der Zugriff auf die Dateifreigabe gewährt wird. Er benutzt die gleiche Syntax wie der Parameter `admin users`.

Standardmäßig ist dieser Parameter leer, was bedeutet, dass jeder Benutzer auf die Freigabe zugreifen kann. Ein Beispiel:

```
valid users = fred @accounts
```

Mit dieser Einstellung sind der Benutzer `fred` und alle Benutzer in der Gruppe `accounts` die einzigen, die auf die Freigabe zugreifen können.

writable

Dieser Freigabe-Parameter (und sein Synonym `writeable`, für die, die Probleme mit der Rechtschreibung haben) zeigt an, ob Clients die Freigabe beschreiben können. Siehe auch `read only`.

Der Standardwert für diesen Parameter ist `no`, d.h. standardmäßig ist eine Freigabe also nur lesbar. Beispiele für die Benutzung des Parameters sind:

```
writable = yes
writeable = yes
read only = no
```

Alle diese Einstellungen führen dazu, dass jeder die Freigabe beschreiben kann, der die Berechtigungen hat, Dateien und Verzeichnisse in der Freigabe zu beschreiben.

write list

Dieser Freigabe-Parameter definiert eine Liste von Benutzern, die Schreib-/Lese-Zugriff auf eine Freigabe haben, unabhängig vom Wert des Parameters `read only`.

Ist ein Benutzer sowohl in der `read-list` als auch in der `write-list`, wird ihm Schreibzugriff gewährt.

Standardmäßig hat dieser Parameter keinen Wert, was bedeutet, dass für alle Benutzer der Wert des Parameters `read only` gilt. Ein Beispiel:

```
write list = root @admin
```

Diese Einstellung definiert, dass zumindest der Benutzer `root` und alle Benutzer in der Gruppe `admin` Schreibzugriff auf die Freigabe haben.

7.3.2 Zugriffsrechte für Ihre erste Freigabe first-share einrichten

Nachdem Sie sich die für den Zugriff auf Freigaben relevanten Parameter angesehen haben, können Sie damit beginnen, die Probleme zu korrigieren, auf die Sie vorher in puncto Zugriff gestoßen sind, als Sie nicht in Ihre Freigabe `first-share` schreiben konnten.

Die Probleme liegen darin begründet, dass für eine Freigabe standardmäßig folgender Parameter eingerichtet ist:

```
writable = no
```

Um sicherzustellen, dass weder Datei- noch Verzeichnisberechtigungen zu den Problemen führten, schauen Sie sich die Unix-Berechtigungen für das freigegebene Verzeichnis an und ändern Sie sie für jedermann zugänglich in `RWX` (0777 der Einfachheit halber) um. Wenn Sie Ihre `umask` nicht geändert haben, sieht `/home/first-share` auf Ihrem Samba-Server wie folgt aus:

```
ls -al /home/first-share
total 4
drwxr-xr-x    2  root  root    1024 Jan 5 14:23
drwxr-xr-x    17 root  root    1024 Jan 5 14:23
-rw-r--r--    1  root  root      69 Jan 5 14:22 file-1.txt
-rw-r--r--    1  root  root      59 Jan 5 14:23 file-2.txt
```

Ändern Sie die Berechtigungen für das Verzeichnis nun folgendermaßen auf 0777 um:

```
chmod 0777 /home/first-share
```

Damit sollte jeder Unix-Benutzer in das Verzeichnis schreiben können, da hiermit Benutzer-, Gruppen- und allgemeines Lese-, Schreib- und Ausführungsrecht eingerichtet wurde.

Wenn Sie aber nun versuchen, von Ihrem Client eine Datei in der Freigabe zu erstellen, erhalten Sie das, was Sie in Abbildung 7.8 sehen.

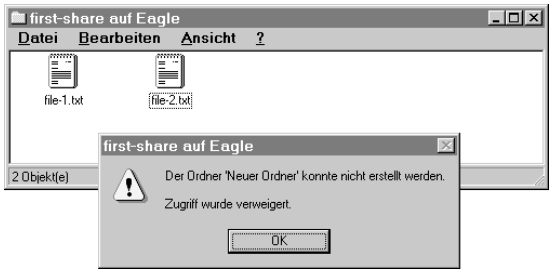


Abb. 7.8: Der Zugriff wird verweigert, obwohl das Verzeichnis schreib- und lesbar ist

Fügen Sie Ihrer Freigabe `first-share` Folgendes hinzu (und starten Sie Samba neu), damit Clients in die Freigabe schreiben können:

```
writable = yes
```

Versuchen Sie es! Sie können jetzt Verzeichnisse und Dateien in der Freigabe erstellen, wie Sie in Abbildung 7.9 sehen.

Abb. 7.9: Kann die Freigabe beschrieben werden, können Sie darin Ordner erstellen

Auf dem Samba-Server sieht Ihr freigegebenes Verzeichnis nun so aus:

```
ls -al /home/first-share
total 4
drwxrwxrwx    2  root  root    1024 Jan 5 14:23
drwxr-xr-x    17 root  root    1024 Jan 5 14:23
drwxr-xr-x    2  boss  boss    1024 Jan 6 01:09 Neuer Ordner
-rw-r--r--    1  root  root      69 Jan 5 14:22 file-1.txt
-rw-r--r--    1  root  root      59 Jan 5 14:23 file-2.txt
```

Der von Ihnen erstellte Ordner `Neuer Ordner` wurde als Verzeichnis erzeugt, und zwar mit Ihnen als Besitzer und Ihrer Gruppe als Gruppenbesitzer. Aber dies ist nur möglich, weil Sie das Verzeichnis `/home/first-share` auf den Modus `0777` eingestellt haben, was relativ gefährlich ist.

Sie könnten viele der anderen Parameter einrichten, die für den Zugriff auf `first-share` relevant sind, darunter

- `write list` – Die Liste der Benutzer, die Schreibzugriff auf die Freigabe haben.
- `valid users` – Die Liste der Benutzer, die auf die Freigabe zugreifen können.

Ihre Freigabe kann z.B. so aussehen:

```
[first-share]
   comment = Meine erste Freigabe
   path = /home/first-share
   browsable = yes
   writable = yes
   valid users = boss joe +users
   write list = root boss
```

`first-share` wurde dahingehend geändert, dass `boss`, `joe` und jeder Benutzer in der Unix-Gruppe `users` auf die Freigabe zugreifen, aber nur `boss` und `joe` in die Freigabe schreiben können.

Sie sollten sich jetzt ansehen, wie Dateiberechtigungen von Samba gehandhabt werden, damit Sie verstehen können, wie Unix-Dateiberechtigungen mit Anfragen für das Lesen oder Beschreiben von Dateien zusammenspielen.

7.4 Berechtigungen

Wie Sie im vorherigen Abschnitt gesehen haben, führt Samba zuerst grobe Überprüfungen durch, z.B. zum Lese-, Schreib- oder benutzerbasierten Zugriff auf Freigaben. Wenn eine verlangte Operation, wie z.B. das Öffnen einer Datei zum Lesen oder Bearbeiten, diese Zugriffsüberprüfungen durchlaufen hat, muss sie immer noch den normalen Beschränkungen des Betriebssystems für Dateien und Verzeichnisse entsprechen. Diese basieren auf dem Benutzer, den Samba als den zugreifenden Benutzer auf die Dateifreigabe definiert hat.

Das heißt, dass normale Unix-Berechtigungen für alle Dateioperationen gelten, nachdem Samba die Zugriffsberechtigung erteilt hat.

Erinnern Sie sich daran, dass Sie vorher die Dateien `file-1.txt` und `file-2.txt` im Editor öffnen konnten. Dies liegt daran, dass sie allgemein lesbar sind und daher jeder auf sie zugreifen kann. Hier ist noch einmal eine detaillierte Auflistung des Freigabeverzeichnisses:

```
ls -al /home/first-share
total 4
drwxr-xr-x    2  root  root    1024 Jan 5 14:23
drwxr-xr-x    17 root  root    1024 Jan 5 14:23
drwxr-xr-x    2  boss  boss    1024 Jan 6 01:09 Neuer Ordner
-rw-r--r--    1  root  root      69 Jan 5 14:22 file-1.txt
-rw-r--r--    1  root  root      59 Jan 5 14:23 file-2.txt
```

Wie Sie sehen können, haben die Dateien `file-1.txt` und `file-2.txt` den Status `0644`, das heißt, sie sind nicht gruppen- oder allgemein beschreibbar. Wenn Sie versuchen, diese Dateien von einem Client zu ändern, werden Sie keinen Erfolg haben. Öffnen Sie eine der Dateien im Editor und versuchen Sie, sie zu ändern. Wann erhalten Sie eine Fehlermeldung? Sie sollten dann eine erhalten, wenn Sie versuchen, Ihre Änderungen zu speichern. Irgendwann gibt der Editor die Fehlermeldung aus, die Sie in Abbildung 7.10 sehen.

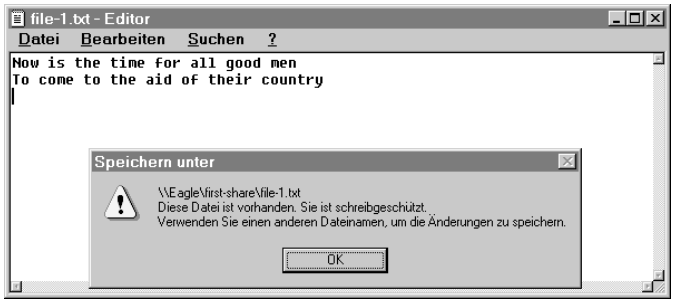


Abb. 7.10: Der Editor kann file-1.txt nicht speichern

Diese Fehlermeldung liegt in der Art und Weise begründet, wie der Editor versucht, die Datei zu speichern. Er versucht, die Datei `file-1.txt` in der Dateifreigabe zu erzeugen, aber da diese Datei bereits existiert und Sie keine Berechtigung haben, die Datei zu bearbeiten, wird die Operation nicht ausgeführt.

Würden Sie versuchen, die Datei unter einem Namen zu speichern, der noch nicht in dem Verzeichnis (oder der Freigabe, die Begriffe sind aus Sicht desjenigen, der Samba verwaltet, austauschbar) existiert, wäre der Editor bereit und würde eine neue Datei erstellen.

Was Sie hier sehen, ist Folgendes: Nach allen Zugriffskontrollen, die Samba durchführt, überprüft es außerdem die normalen Berechtigungen für das Dateisystem, die der Benutzer (den Samba als mit der Dateifreigabe verbunden ansieht) für die Dateien in der Freigabe hat!

Wenn Sie keine Leseberechtigungen für die Verzeichnisse haben, werden Sie auch ihre Inhalte nicht betrachten können, obwohl Sie die Verzeichnisse selbst sehen können. Normalerweise erhalten Sie Meldungen mit dem Inhalt »Zugriff verweigert«, wenn Sie versuchen, durch solche Verzeichnisse zu browsen.

Nun, wenn Dateien und Verzeichnisse in einer Samba-Freigabe erzeugt werden, wer besitzt sie und welche Berechtigungen werden ihnen vererbt? Schauen Sie sich noch einmal das lange Listing für das Verzeichnis `first-`

`share` an und werfen Sie einen Blick auf den Eintrag für den Ordner `Neuer Ordner`.

Hier noch einmal der Eintrag:

```
drwxr-xr-x    2  boss  boss    1024 Jan 6 01:09 Neuer Ordner
```

Sie sind der Besitzer des Verzeichnisses, das sich in Ihrer primären Gruppe befindet und das Dateirecht 0755 hat.

Lassen Sie uns nun einen kurzen Blick auf Unix-Dateirechte werfen, da Samba Nummern wie 0755 benutzen möchte, wie Sie vorher gesehen haben. Nachfolgend zeige ich Ihnen, was sie bedeuten und wie Sie sie konstruieren können.

Jede Datei in einem Unix-Dateisystem hat einen Besitzer, eine Gruppenzugehörigkeit und Dateirechte (manchmal auch Berechtigungen genannt), die aus vier Teilen bestehen:

- ✘ Die SETUID-, SETGID- und T-Bits.
- ✘ Die Benutzer- oder Besitzerberechtigungen, die eine beliebige Kombination aus Lese-, Schreib- und Ausführungsrecht sein können. Sie bestimmen, welche Zugriffsrechte der Besitzer auf die Datei hat.
- ✘ Die Gruppenberechtigungen, die eine beliebige Kombination aus Lese-, Schreib- und Ausführungsrecht sein können. Sie bestimmen, welche Zugriffsrechte jeder der Benutzer der Gruppe, in der sich die Datei befindet, auf die Datei hat.
- ✘ Die anderen oder allgemeinen Berechtigungen, die eine beliebige Kombination aus Lese-, Schreib- und Ausführungsrecht sein können. Sie bestimmen, welche Zugriffsrechte jeder auf die Datei hat, der nicht der Besitzer ist und sich nicht in der Gruppe des Besitzers befindet.

Ein Dateirecht wird normalerweise durch die drei Buchstaben `RWX` und den Bindestrich (-) dargestellt. `RWX` heißt Lesen, Schreiben und Ausführen, während `R-X` Lesen und Ausführen bedeutet. Um ein Dateirecht zu ändern, drücken Sie es als eine Serie von 12 Bits oktal aus. Daher wird das Recht einer Datei als vier oktale Ziffern dargestellt, wobei die erste Ziffer auf folgende Art codiert ist:

4=SETUID, 2=SETGID, 1=T oder STICKY Bit

Die verbleibenden drei Ziffern (die eigentlichen Berechtigungsbits) werden auf folgende Weise codiert:

4 = Lesen, 2 = Schreiben, 1 = Ausführen

Wenn Sie ein bestimmtes Recht für eine Datei konstruieren wollen, addieren Sie einfach die Codierungen für die von Ihnen gewünschten Berechtigungen. `RWX` summiert sich also oktal zu 7, `RW-` zu 6, `R-X` zu 5 usw.

Ein Dateirecht von 1755 heißt also:

- ✘ Für die Datei ist das T-Bit eingestellt, oder für das Verzeichnis wurde das Sticky Bit eingerichtet.
- ✘ Der Besitzer hat Lese-, Schreib- und Ausführungzugriff.
- ✘ Mitglieder der Besitzergruppe haben Lese- und Ausführungszugriff.
- ✘ Jeder andere hat Lese- und Ausführungszugriff.

Abschließend möchte ich noch erwähnen, dass eine Dateiberechtigung von 0755 von `ls -al` als `RWXR-XR-X` ausgegeben wird.

Samba handhabt die Erstellungsberechtigungen für Dateien und Verzeichnisse separat. Sie können eine ganze Reihe von Parametern einrichten, um sowohl die Eigentumsverhältnisse als auch die Berechtigungen für erstellte Dateien und Verzeichnisse zu kontrollieren.

7.4.1 Parameter für die Erstellung von Dateien und Verzeichnissen

Die folgenden Abschnitte listen viele der Freigabe-Parameter auf, die für die Berechtigungen und Eigentumsverhältnisse der von Samba erstellten Dateien und Verzeichnisse relevant sind. Wie immer finden Sie eine komplette Auflistung der Parameter und ihre aktuellen Funktionen in den Manpages zur `smb.conf` der aktuellen Samba-Version.

create mask, create mode

Diese Freigabe-Parameter sind synonym und kontrollieren die Berechtigungen, die bei der Erstellung von Dateien vergeben werden. Der gegebene Wert ist eine Bitmaske, die mit der aus dem verlangten DOS-Attrribut kalkulierten Unix-Maske verglichen wird.

Jedes nicht eingestellte Bit der Maske wird bei Erstellung der Datei aus den Berechtigungen für die Datei entfernt.

Standardmäßig hat die Erstellungsmaske einen Wert von `0744`, der festlegt, dass der Besitzer für neue Dateien `RWX`-Berechtigungen erhält, während Mitglieder der Besitzergruppe und alle anderen Benutzer nur die Berechtigungen `R--` erhalten.

Ein Beispiel:

```
create mask = 0755
```

Diese Einstellung definiert, dass der Besitzer RWX-Berechtigungen hat, Mitglieder der Besitzergruppe R-X-Berechtigungen und alle anderen Benutzer R-X-Berechtigungen erhalten.

directory mask, directory mode

Diese Freigabe-Parameter sind synonym und kontrollieren die Berechtigungen, die während der Erstellung von Verzeichnissen vergeben werden. Der gegebene Wert ist eine Bitmaske, die mit der Unix-Maske verglichen und aus dem verlangten DOS-Atrribut kalkuliert wird.

Jedes nicht eingestellte Bit in der Maske wird aus den Berechtigungen für die Datei entfernt, wenn es erstellt wird.

Standardmäßig hat die Verzeichnismaske einen Wert von 0755. Ein Beispiel:

```
directory mask = 0744
```

Diese Einstellung definiert, dass der Besitzer RWX-Berechtigungen hat, Mitglieder der Besitzergruppe und alle anderen Benutzer nur die Berechtigungen R-- erhalten.

Das Ausführungsbit hat eine spezielle Bedeutung für Verzeichnisse. Es ermöglicht einem Benutzer, in dieses Verzeichnis zu wechseln. Wenn also Benutzer keinen X-Zugriff auf ein Verzeichnis haben, können sie Dateien in dem Verzeichnis öffnen, wenn sie Lesezugriff auf die Dateien haben, aber sie können nicht in das Verzeichnis wechseln. Aufgrund der Art und Weise jedoch, wie Samba und Windows das Browsing von Ordnern handhaben, spielt das X-Bit für den Zugriff von Windows aus keine Rolle.

force create mode

Über diesen Freigabe-Parameter können Sie bestimmte Berechtigungsbits erzwingen, wenn Dateien in einer Freigabe erstellt werden. Dies können Sie tun, indem Sie ein bitweises or der hier spezifizierten Bits mit den Bits durchführen, die von create mask berechnet werden. Beachten Sie, dass damit der Paramter force create mode den Parameter create mask außer Kraft setzt.

Der Standardwert für diesen Paramter ist 0000, was heißt, dass keine zusätzlichen Berechtigungsbits in den Parameter create mask/mode gezwungen werden. Ein Beispiel:

```
force create mode = 0755
```

Mit dieser Einstellung haben die erstellten Dateien eine Berechtigung von mindestens 0755 (oder `RWXR-XR-X`).

force directory mode

Über diesen Freigabe-Parameter können Sie bestimmte Berechtigungsbits erzwingen, wenn Verzeichnisse in einer Freigabe erstellt werden. Dies können Sie `tun`, indem Sie ein bitweises `or` der hier spezifizierten Bits mit den Bits durchführen, die von `create mask` berechnet werden. Beachten Sie, dass damit der Paramter `force directory mode` den Parameter `directory mask` außer Kraft setzt.

Der Standardwert für diesen Paramter ist `0000`, was heißt, dass keine zusätzlichen Berechtigungsbits in den Parameter `directory mask/mode` gezwungen werden. Ein Beispiel:

```
force directory mode = 0755
```

Mit dieser Einstellung haben die erstellten Verzeichnisse eine Berechtigung von mindestens 0755 (oder `RWXR-XR-X`).

force group

Dieser Freigabe-Parameter spezifiziert einen Unix-Gruppennamen, der als Standard-Primärgruppe für alle Benutzer verwendet wird, die auf die Freigabe zugreifen.

Standardmäßig hat dieser Parameter keinen Wert, was bedeutet, dass alle neuen Dateien und Verzeichnisse über die Anwendung der normalen Unix-Regeln einen Gruppenbesitzer erhalten (ist das SETGID-Bit auf das Vaterverzeichnis eingestellt, wird die Gruppe dieses Verzeichnisses benutzt, sonst die primäre Gruppe des Erstellers).

Ein Beispiel:

```
force group = users
```

Mit dieser Einstellung werden alle neuen Dateien in der Freigabe mit einem Gruppenbesitzer von `users` erstellt.

force user

Dieser Freigabe-Parameter spezifiziert einen Unix-Benutzernamen, der als Standardbenutzer für alle Benutzer verwendet wird, die auf die Freigabe zugreifen.

Standardmäßig hat dieser Parameter keinen Wert, was bedeutet, dass alle neuen Dateien in der Freigabe dem Unix-Benutzer gehören, der als mit der

Freigabe verbunden angesehen wird (siehe den Abschnitt »Zugriffsrechte« vorher in diesem Kapitel). Ein Beispiel:

```
force user = boss
```

Diese Einstellung bedeutet, dass alle neue Dateien in der Freigabe Eigentum von `boss` sind.

7.4.2 Einige Beispiele

Sie haben nun viele der Parameter kennengelernt, die für die Erstellung von Dateien und den Zugriff auf Dateien in Samba relevant sind. Wie können Sie diese anwenden? Hier sind einige Beispiele.

Wenn Sie wollen, dass alle Dateien, die in einem bestimmten Verzeichnis erstellt werden, einer bestimmten Gruppe gehören, benutzen Sie den Parameter `force group`. Wollen Sie z.B., dass alle Dateien und Verzeichnisse, die in einer bestimmten Freigabe erzeugt werden, den Gruppen-Accounts gehören, verwenden Sie für die Freigabe den folgenden Parameter:

```
force group = accounts
```

Um zu verhindern, dass alle Dateien und Verzeichnisse, die in einer bestimmten Freigabe erstellt werden, allgemein offene Berechtigungen haben (um vielleicht Unix-Benutzer daran zu hindern, auf die Dateien in der Freigabe zuzugreifen), benutzen Sie die folgenden Parameter für die Freigabe:

```
create mask = 0750
directory mask = 0750
```

Sie müssen beide Parameter definieren, da Samba die Erstellung von Dateien und Verzeichnissen separat handhabt.

Modifizieren Sie Ihre Freigabe `first-share`, um einige dieser Änderungen einzufügen:

```
[first-share]
comment = Meine erste Freigabe
   path = /home/first-share
   browsable = yes
   writable = yes
   create mask = 0750
   create directory = 0750
   force group = users
```

Nachdem Sie Ihre `smb.conf` modifiziert haben, um `first-share` mit den vorher beschriebenen Parametern zu ändern, starten Sie Samba neu und

erstellen von einem Client die Datei `new-file.txt` und das Verzeichnis `Neuer Ordner (2)`.

Wenn Sie sich das Freigabeverzeichnis komplett auflisten lassen, sollten Sie nun Folgendes sehen:

```
ls -al /home/first-share
total 4
drwxrwxrwx   4  root  root    1024 Jan 6 16:58
drwxr-xr-x  17  root  root    1024 Jan 5 14:23
drwxr-x---   2  boss  boss    1024 Jan 6 14:53 Neuer Ordner
drwxr-x---   2  boss  users   1024 Jan 6 16:58 Neuer Ordner (2)
-rw-r--r--   1  root  root      69 Jan 5 14:22 file-1.txt
-rw-r--r--   1  root  root      59 Jan 5 14:23 file-2.txt
-rwxr-----   1  boss  users      0 Jan 6 16:58 new-file.txt
```

Beachten Sie, dass die Datei `new-file.txt` und das Verzeichnis `Neuer Ordner (2)` beide den Gruppenbesitzer `users` und keine allgemeinen Berechtigungen haben.

Sie haben für das Verzeichnis `/home/first-share` die Berechtigungen 0777 eingerichtet, was sehr gefährlich ist. Eine bessere Methode, über die Sie Dateien in der Freigabe von Clients erstellen können, besteht darin, den Gruppenbesitzer des Verzeichnisses in eine Gruppe zu ändern, der Sie angehören.

Um dies zu tun, müssen Sie herausfinden, zu welchen Gruppen auf dem Unix-Rechner Sie gehören. Wenn Sie sich also als `boss` in Ihren Client eingeloggt haben, müssen Sie feststellen, welchen Gruppen `boss` angehört:

```
group boss
boss: boss wheel users
```

Ändern Sie dann den Gruppenbesitzer des Verzeichnisses `/home/first-share` in eine dieser Gruppen. `users` ist eine gute Wahl, besonders wenn Sie planen, dass andere Leute auf die Dateien in der Freigabe zugreifen und sie gemeinsam benutzen sollen. Sie müssen außerdem das Gruppen-Schreibbit für das Verzeichnis einrichten.

Um diese Änderungen durchzuführen, benutzen Sie die folgenden Befehle:

```
chgrp users /home/first-share
chmod 0775 /home/first-share
```

Wenn Sie sicherstellen wollen, dass diese Änderungen für alle Dateien und Verzeichnisse in der Freigabe gelten, fügen Sie den Befehlen `chgrp` und `chmod` den Parameter `-R` hinzu.

Befehle:

```
chgrp -R users /home/first-share
chmod -R 0775 /home/first-share
```

All dies verdeutlicht einen sehr nützlichen Aspekt von Samba: Sie können die Dateien in Ihren Dateifreigaben von Unix aus verwalten. Das heißt, Sie können auf alle Standard-Unix-Funktionen zugreifen, darunter Skripting (auch Perl) und `cron`-Dateien.

In einem Studentenlabor z.B., in dem von den Studenten verlangt wird, dass sie ihre Laborarbeiten zu einem bestimmten Datum und einer bestimmten Zeit übermitteln (indem sie es in die Freigabe `\\eagle\labwork` kopieren), bewirkt das folgende Shell-Skript, dass

- ✘ die Dateien Eigentum des Professors werden, damit Studenten ihre Beiträge nach dem Abgabetermin nicht mehr ändern können,
- ✘ der Professor eine Mail erhält, die ihm mitteilt, welche Studenten ihre Arbeit noch nicht übermittelt haben.

```
#!/bin/sh
# Finde heraus, wer seine Laborarbeit noch nicht übertragen
# hat, und ändere dann den Besitzer, damit Studenten nicht
# nach dem Abgabetermin abgeben können.
# Sende Mail an Professor über die, die ihre Arbeit nicht
# abgegeben haben. Offensichtlich gibt es ein kleines
# Risiko, dass jemand erneut übertragen kann. Dies könnte
# dadurch behoben werden, den Samba-Server zu beenden.
# Zunächst sollten alle Studenten daran gehindert werden,
# Dateien von PCs in das Verzeichnis einzufügen.
chmod 1700 /home/labwork
# Finde dann heraus, wem die übertragene Arbeit gehört
ls -ld /shares/labwork | tr -s " " " " | cut -f3 -d" " > ↵
   /tmp/submitted.$$
# Ändere nun den Besitzer dieser Dateien
chown -R professor /home/labwork/*
# Finde nun heraus, wer noch nicht übertragen hat,
# basierend auf einer Studentenliste
diff -y /tmp/submitted.$$ /home/professor/students | grep \< ↵
   | \ mail -s "Studenten, die Ihre Arbeit nicht rechtzeitig ↵
   übertragen haben" professor
rm -f /tmp/submitted.$$
```

Dieses Shell-Skript ist nur ein Beispiel. In der Realität, in der Professoren oft mehrere Klassen haben usw., wären einige Änderungen nötig.

7.5 Spezielle Dateifreigaben

Jetzt, da Sie sich angesehen haben, wie Dateifreigaben eingerichtet werden, wie Sie den Zugriff auf diese Freigaben kontrollieren und wie Sie Zugriffsberechtigungen für Freigaben handhaben, ist es an der Zeit, einige spezielle Freigaben anzusehen, die Samba bietet.

Clients greifen gern auf ihre Home-Verzeichnisse zu, und wenn Samba das Home-Verzeichnis eines jeden Benutzers als separate Dateifreigabe darstellen müsste, hätten Administratoren ein schweres Leben. Stellen Sie sich vor, Sie müssten für jeden neuen Benutzer, den Sie dem System hinzufügen, einen Abschnitt in `smb.conf` einfügen und dann jedes Mal Samba neu starten.

Um das Leben all der überarbeiteten Administratoren etwas leichter zu gestalten, bietet Samba zwei spezielle Freigabeabschnitte: `[homes]` und `[printers]`. Ich werde den Abschnitt `[printers]` im nächsten Kapitel ausführlich darstellen. Die Details zum Abschnitt `[homes]` finden Sie nachfolgend.

Wenn ein Client eine Verbindung zu einer Dateifreigabe anfordert, werden die existierenden Dateifreigaben überprüft. Wird eine Entsprechung gefunden, wird diese Freigabe zur Verfügung gestellt. Wenn Samba aber keine Entsprechung findet, wird die angeforderte Freigabe als Benutzername behandelt und in der `passwd` gesucht. Existiert der Name und kann er authentifiziert werden, wird eine Freigabe durch Klonen der `[homes]`-Freigabe erstellt. Das heißt, dass die neue Freigabe die meisten der Parameter in der `[homes]`-Freigabe übernimmt.

Wenn die neue Freigabe erstellt wird, wird ihr Name in den Benutzernamen geändert und der Pfad der Freigabe auf das Home-Verzeichnis des Benutzers eingestellt, wenn kein solches im Abschnitt `[homes]` definiert ist.

Nachfolgend finden Sie ein Beispiel für eine `[homes]`-Freigabe in der `smb.conf`:

```
[homes]
   comment = Home-Verzeichnisse
   browsable = no
   writable = yes
```

Das ist alles, was Sie brauchen, damit Clients auf das angeforderte Home-Verzeichnis zugreifen können. Fügen Sie die oben stehenden Einträge in Ihre `smb.conf` ein und starten Sie Samba neu. Sie sollten dann unter Windows die DOS-Eingabeaufforderung aufrufen und folgenden Befehl ausführen können:

```
net use h: \\EAGLE\homes
```

Danach können Sie auf alle Dateien in Ihrem Home-Verzeichnis von Ihrem PC zugreifen.

7.6 Handhabung und Umsetzung von Dateinamen

Unix-Dateinamen und Windows-Dateinamen folgen verschiedenen Regeln.

Unix erlaubt fast jedes Zeichen in einem Dateinamen, außer dem Verzeichnistrennzeichen (/) und Escape, und unterscheidet in Namen zwischen großen und kleinen Buchstaben. Außerdem können Unix-Dateinamen sehr lang sein (bis zu 255 Zeichen). Auch Pfadnamen sind unter Unix oft bis zu 1.024 Zeichen lang.

DOS (6.22 und früher) dagegen hat die Einschränkung auf die 8.3-Namen, die nicht länger als acht Zeichen sein dürfen, mit einer Dateierweiterung, die aus nicht mehr als drei Zeichen bestehen darf. Außerdem benutzt DOS für Datei- und Verzeichnisnamen Großbuchstaben und beschränkt die Länge der Pfadnamen, die erheblich kürzer sind als die, die in Unix-Systemen erlaubt sind. Windows für Workgroups folgt den DOS-Beschränkungen.

Für Windows 95 (mit DOS 95) und Windows NT wurden viele dieser Einschränkungen aufgehoben, so dass Dateinamen länger als 11 Zeichen sein können und sowohl große als auch kleine Buchstaben in Dateinamen erlaubt sind. Aber auch Windows 95 und Windows NT haben Beschränkungen für die Länge von Datei- und Pfadnamen, die wesentlich kleiner sind als in Unix-Systemen. Windows 95 kürzt Dateinamen nach 127 Zeichen ab. Beträgt der komplette Pfadname mehr als 255 Zeichen (inklusive dem Server- und den Freigabenamen), weigert sich Windows 95, weitere Ordner oder Dateien zu erstellen. Windows NT unterliegt den gleichen Beschränkungen. Aber sowohl Windows 95 als auch Windows NT können mit längeren Datei- und Pfadnamen umgehen, wenn diese bereits in der Samba-Freigabe existieren (und vielleicht unter Unix erzeugt wurden).

Um für Kompatibilität mit älteren Clients (DOS, Windows für Workgroups, PATHWORKS usw.) und Anwendungen, die von 8.3-Dateinamen abhängen, zu sorgen, bietet Samba viele Freigabe-Parameter, die kontrollieren, wie Unix-Dateinamen und -Pfadnamen an Clients ausgegeben werden. Außerdem gibt es Parameter, die kontrollieren, wie die Groß-/Kleinschreibung beim Erzeugen neuer Dateien verwendet wird.

Samba bezeichnet diese Umsetzung der Dateinamen als *Name Mangling* und verwendet den folgenden allgemeinen Ansatz:

- Die ersten fünf alphanumerischen Zeichen vor dem ersten Punkt werden in Großbuchstaben umgewandelt und als erster Teil des umgesetzten Namens verwendet.
- An diesen ersten Teil des umgesetzten Namens wird eine Tilde (~) angehängt, gefolgt von einem aus zwei Zeichen bestehenden Hash-Wert des ursprünglichen `root`-Namens (d.h. ohne die ursprüngliche Erweiterung). Die Erweiterung wird aber in die Berechnung für den Hash-Wert eingefügt, wenn sie groß geschriebene Zeichen enthält oder länger als drei Zeichen ist. Statt der Tilde kann auch ein anderes Zeichen verwendet werden (über den Parameter `mangling char`), wenn Benutzer widersprechen oder Anwendungen Probleme mit Tilden haben.
- Die ersten drei Zeichen der ursprünglichen Erweiterung werden in Großbuchstaben umgewandelt und als Erweiterung für den umgesetzten Namen verwendet. Wenn der Dateiname keinen Punkt enthält, hat der umgesetzte Name keine Erweiterung.
- Dateien, die Unix-Namen haben, die mit einem Punkt beginnen, werden als versteckte DOS-Dateien behandelt. Ihre umgesetzten Namen ähneln den vorher dargestellten, allerdings mit der Ausnahme, dass der erste Punkt entfernt und eine Erweiterung von ___ (d.h. drei Underscores) hinzugefügt wird, unabhängig von der ursprünglichen Erweiterung.

Nachfolgend finden Sie ein Beispiel für die Umsetzung von Namen, die von einem Windows-95-DOS-Rechner durchgeführt wurde, der Ihre Freigabe `first-share` auflistet, nachdem einige neue Dateien und Ordner hinzugefügt wurden. Die umgesetzten Namen sehen Sie auf der linken Seite und die vollständigen Namen auf der rechten.

```
E:\>dir
Volume in drive E ist FIRST-SHARE
Directory of E:\
file-1   txt       69  01-05-99  2:22p  file-1.txt
file-2   txt       59  01-05-99  2:23p  file-2.txt
NEWFO~YX     <DIR>     01-06-99  2:53p  New Folder
File-1   txt       69  01-06-99  5:33p  File-1.txt
new-file txt        0  01-06-99  4:58p  new-file.txt
ANOTH~9Y     <DIR>     01-06-99  4:58p  another-new-folder
A-FIL~BH TXT       24  01-07-99 12:45a  a-file-with-a-long-name.txt
         5 file(s)           221 bytes
         2 dir(s)     33,488,896 bytes free
E:\>
```

Standardmäßig arbeitet Samba 2.0 wie ein Windows-NT-Server: D.h., es unterscheidet nicht zwischen Groß- und Kleinschreibung, behält aber die jeweilige Groß-/Kleinschreibung bei. Wenn Samba also Dateien öffnet, stellt es Dateinamen entsprechend in einer nicht groß-/kleinsensitiven Art und Weise dar, wenn es aber neue Dateien erstellt, behält Samba die Schreibweise bei, die der Client verwendet.

In den meisten Fällen sind die Standardeinstellungen von Samba genau richtig, aber Sie müssen möglicherweise einige dieser Einstellungen für spezielle Clients oder Anwendungen ändern.

Die folgenden Parameter sind für die Handhabung von Dateinamen relevant. Wie immer finden Sie eine vollständige Liste der Parameter und das letzte Wort zu ihren Funktionen in den Manpages zu `smb.conf` für die aktuelle Samba-Version.

mangled names

Dieser Freigabe-Parameter kontrolliert, ob Unix-Namen, die nicht mit DOS-Namen kompatibel sind, in DOS-kompatible Namen umgesetzt werden sollen. Standardmäßig setzt Samba Namen für Clients um, die Nicht-DOS-Namen nicht handhaben können.

Der Standardwert für diesen Parameter ist `yes`. Standardmäßig werden also Nicht-DOS-Namen in ihre DOS-kompatiblen Entsprechungen umgesetzt.

Wenn Sie den Wert für diesen Parameter auf `no` setzen, werden die Namen nicht umgesetzt. Dann sehen DOS-Clients und DOS-Befehlsprompts den Dateinamen einfach abgeschnitten, wie es den normalen DOS-Regeln entspricht.

mangle case

Dieser Freigabe-Parameter kontrolliert, ob Dateinamen umgesetzt werden, die nicht der Standardschreibweise (definiert über `default case`) entsprechen. Ist *dieser* Parameter aktiviert, werden Namen wie z.B. *Mail* in die Standardschreibweise umgesetzt.

Der Standardwert für diesen Parameter ist `no`. Damit wird definiert, dass Namen in gemischter Schreibweise nicht umgesetzt werden.

mangling char

Dieser Freigabe-Parameter definiert das Zeichen, das Samba als Mangling-Zeichen verwendet, wenn es Namen umsetzt. Standardmäßig ist dies die Tilde (~).

Ein Beispiel:

```
mangle char = ^
```

Mit dieser Einstellung benutzt Samba das (^) statt der Tilde.

case sensitive

Dieser Freigabe-Parameter bestimmt, ob Samba bei Dateinamen auf die Groß-/Kleinschreibung achten soll. Ist dieser Parameter auf `no` eingestellt, muss Samba eine nicht groß-/kleinsensitive Suche für alle Dateinamen durchführen, die von Clients übertragen werden.

Der Standardwert für diesen Parameter ist `no`.

default case

Dieser Freigabe-Parameter kontrolliert die Standardschreibweise für neue Dateien und sollte in Kombination mit dem Parameter `preserve case` verwendet werden.

Die Standardschreibweise ist Kleinschreibung.

preserve case

Dieser Freigabe-Parameter kontrolliert das Verhalten von Samba beim Erstellen neuer Dateien. Ist der Wert für diesen Parameter auf `yes` gesetzt, wird die Schreibweise benutzt, die der Client verwendet (auch gemischte Schreibweise), sonst wird die Schreibweise verwendet, die durch den Parameter `default case` definiert ist.

Standardmäßig wird die Schreibweise beibehalten.

short preserve case

Dieser Freigabe-Parameter kontrolliert das Verhalten von Samba beim Erstellen von Dateien mit DOS-kompatiblen Namen (d.h. 8.3-Namen in Großbuchstaben). Ist der Wert für diesen Parameter auf `yes` gesetzt, werden solche Dateien mit groß geschriebenen Namen erstellt, sonst wird die Schreibweise verwendet, die durch den Parameter `default case` definiert ist.

Dieser Parameter kann mit `preserve case = yes` verwendet werden, damit lange Dateinamen ihre Schreibweise beibehalten können, während kurze Namen klein geschrieben werden.

Der Standardwert für diesen Parameter ist `yes`.

7.7 Datei-Locking

Standardmäßig unterstützt Samba zwei verschiedene Arten von Datei-Locking, *share modes* und opportunistisches Locking, kurz *oplocks*.

share modes unterstützen die Standard-DOS/Windows-Zugriffsanfragen von `DENY_DOS`, `DENY_ALL`, `DENY_READ`, `DENY_WRITE`, `DENY_NONE` und `DENY_FCB`.

Unter den Unix-Versionen, die *Shared Memory* (gemeinsamen Speicher) unterstützen (die meisten Unix-Versionen), wird die Unterstützung für `share modes` unter Benutzung von Shared Memory implementiert, was sehr schnell ist. Wenn Ihre Unix-Version Shared Memory nicht unterstützt, wird die Unterstützung für `share modes` unter Benutzung von Lock-Dateien implementiert, was sehr langsam sein kann.

Wahrscheinlich müssen Sie `share modes` nicht deaktivieren. Sollte dies doch einmal der Fall sein, können Sie sie über den folgenden Befehl für jede einzelne Freigabe deaktivieren:

```
share modes = no
```

oplocks sind eine Leistungserweiterung, die zusammen mit dem Windows-NT-Server eingeführt wurden. Sie ermöglichen einem Client, viele Dateioperationen zwischenzuspeichern, solange der Client der einzige ist, der auf eine bestimmte Datei zugreift. Öffnet ein anderer Client die gleiche Datei, muss der Server dem Client mit dem oplock einen *oplock break* schicken, sodass dieser Client das lokale Zwischenspeichern beendet.

Wenn Clients oplocks erhalten können, sind Leistungssteigerungen von 30 Prozent und mehr möglich, da Clients aggressive Zwischenspeicherungen von Dateioperationen durchführen können (inklusive Öffnen und Schließen und möglicherweise erneutes Ausführen einiger Operationen im Zwischenspeicher).

oplocks sind standardmäßig in Samba aktiviert. In einigen Fällen ist es möglich, dass Client-Programme bei aktivierten oplocks nicht richtig funktionieren, also wollen Sie sie möglicherweise deaktivieren. Dies können Sie über den `oplocks`-Befehl für einzelne Freigaben erreichen:

```
oplocks = false
```

Sie können die oplocks für einzelne Dateien auch über den Parameter `veto oplock files` deaktivieren:

```
veto oplock files = /*.mbx/
```

Der Vollständigkeit halber möchte ich noch erwähnen, dass SGIs Irix 6.5.2f jetzt oplock-Unterstützung auf Kernel-Ebene bietet, und Linux und BSD dies

ebenfalls bald bieten werden. Samba kann Kernel-oplocks erkennen und sie benutzen, wenn sie verfügbar sind. So können oplocks unterbrochen werden, wann immer ein lokaler Unix-Prozess oder eine NFS-Operation auf eine Datei zugreift, die der `smbd` geschützt hat. Dies bietet größere Datenkonsistenz zwischen SMB, NFS und lokalen Dateizugriffen.

Obwohl Sie normalerweise den Parameter `kernel` nicht brauchen, können Sie oplocks auch folgendermaßen deaktivieren:

```
kernel oplocks = off
```

7.8 Symbolische Links

Standardmäßig folgt Samba symbolischen Links im Unix-Dateisystem, die auf Dateien innerhalb des freigegebenen Verzeichnisses verweisen, nicht aber solchen zu Dateien/Verzeichnissen außerhalb dieses Verzeichnisses.

Zwei Freigabe-Parameter kontrollieren dieses Verhalten: `follow symlinks` und `wide links`.

Standardmäßig sind diese Parameter auf `yes` bzw. `no` gesetzt.

Wenn Sie `follow symlinks` auf `no` setzen werden keine symbolischen Links mehr verfolgt, was zu einer geringen Leistungsabnahme führt.

Die Einstellung `wide links = yes` führt dazu, dass auch Links außerhalb des freigegebenen Verzeichnisses verfolgt werden.

7.9 Handhabung von CD-ROMs

Ein Problem mit der Freigabe von CD-ROMs liegt darin, dass sie in das Dateisystem auf dem Samba-Server gemountet werden müssen. Wenn ein Benutzer die CD-ROM in einem CD-ROM-Laufwerk auswechselt und auf die neue CD zugreifen möchte, muss jemand oder etwas auf Ihrem Samba-Server eingreifen und die CD-ROM mounten. Wäre es nicht großartig, wenn die CD-ROM gemountet werden könnte, sobald der Client auf die CD-ROM-Freigabe zugreift?

Nun, Samba bietet eine solche Funktion mit den Befehlen `preexec/postexec` und `root preexec/root postexec`. Diese Parameter ermöglichen die Ausführung bestimmter Unix-Befehle, wenn ein Client sich mit einer Dateifreigabe verbindet bzw. wenn er die Verbindung zu einer Dateifreigabe beendet. Die Root-Version führt den Befehl einfach als `root` aus.

Eine CD-ROM-Freigabe könnte so aussehen:

```
[cdrom]
   comment = CD-ROM, bei Verbindung automatisch gemountet
   browsable = yes
   read only = yes
   path = /mnt/cdrom
   root preexec = /bin/mount /dev/hdd /mnt/cdrom
   root postexec = /bin/umount /mnt/cdrom
```

Natürlich ist das tatsächlich benutzte Gerät (`/dev/hdd`) abhängig von Ihrem System (dieses gilt für ein Linux-System mit der CD-ROM am zweiten IDE-Controller als Slave-Gerät).

Mit einer solchen Freigabe können Benutzer die CD-ROM im CD-ROM-Laufwerk auf dem Server auswechseln und sie dann erneut mappen (z.B. mit `net use /d v:` und dann `net use v: \\EAGLE\cdrom`).

7.10 Andere Parameter

Die folgenden Parameter sind alle auf irgendeine Art und Weise relevant für die Freigabe von Dateien, passen aber nicht richtig in einen der vorher beschriebenen Absätze.

maxopenfiles

Dieser globale Parameter kontrolliert die maximale Anzahl offener Dateien, die ein `smbd`-Dateifreigabeprozess für einen Client geöffnet haben kann. Seit Samba 2.0.0 ist der Standardwert für diesen Parameter `10.000` Dateien, obwohl `smbd` dies auf einen sinnvolleren Wert setzt, wenn das Betriebssystem so viele offene Dateien nicht unterstützt. Unter Linux wird `maxopenfiles` also standardmäßig auf etwa `246` eingestellt.

In früheren Samba-Versionen war `maxopenfiles` ein Parameter, dessen Wert während der Kompilierung festgesetzt wurde.

nis homedir und homedir map

Diese globalen Parameter weisen Samba an, den Standort von Home-Verzeichnissen über NIS zu holen. Sie werden in Situationen benutzt, in denen sich das Home-Verzeichnis eines Benutzers auf einem entfernten Rechner befindet und Samba über NFS auf dieses zugreifen würde.

Solange wie auf den tatsächlichen Home-Verzeichnis-Servern ebenfalls Samba läuft, kann ein Logon-Server die Heimatfreigabe so zurückgeben, dass sie sich auf einem anderen Server befindet. Dafür konsultiert er die

NIS-Map, die über den Parameter `homedir map` definiert ist. Dies funktioniert nur, wenn es einen funktionierenden NIS-Server gibt und Samba als Logon-Server läuft.

Die Standardwerte für `nis homedir` und `homedir map` sind `false` bzw. `auto.home`.

ole locking compatibility

Dieser globale Parameter ermöglicht es Ihnen, die OLE-Kompatibilität für Bereichslocks zu deaktivieren, die Samba bietet. Einige Unix-Lockmanager können abstürzen oder andere Probleme haben, wenn die OLE-Funktion von Samba aktiviert ist, daher wollen sie sie vielleicht deaktivieren.

Der Standardwert für diesen Parameter ist `yes`.

strip dot

Dieser globale Parameter definiert, ob Samba abschließende Punkte (dots) von Unix-Dateinamen abschneidet. Einige CD-ROMs haben Dateinamen, die mit einem einzelnen Punkt enden.

Der Standardwert für diesen Parameter ist `no`.

7.11 Zusammenfassung

In diesem Kapitel haben Sie die Freigabe von Dateien und viele der Parameter betrachtet, die kontrollieren, wie Dateifreigaben für Clients verfügbar gemacht werden. Sie haben ebenfalls gelernt, wie Dateien in diesen Dateifreigaben erstellt werden und wie auf sie zugegriffen werden kann. Außerdem haben Sie einen Blick auf fortschrittlichere Funktionen im Zusammenhang mit Dateifreigaben geworfen.

Dabei haben Sie sich detailliert die einzelnen Schritte angesehen, die Samba durchläuft, um festzulegen, ob eine Freigabe existiert, ob ein bestimmter Client auf eine verlangte Freigabe zugreifen und ob dieser Client Dateien in der verlangten Freigabe lesen oder beschreiben kann. Mit diesen Informationen können Sie jetzt viele Probleme in der Konfiguration von Samba meistern.

Im nächsten Kapitel werden Sie sich ansehen, wie Sie Druckerfreigaben einrichten, wie Sie Samba installieren, damit es die automatische Installation von Druckertreibern unter Windows 9x unterstützt, und wie Sie von Unix-Systemen aus zu Windows-Clients drucken, die mit Druckern verbunden sind.

7.12 Frage & Antwort

Meine smb.conf-Datei enthält eine Dateifreigabe namens [docs]. In dieser Freigabe sollen die Autoren von Dokumenten ihre Dokumente speichern können. Der Gruppeneigentümer des Verzeichnisses ist docs, und alle Autoren sind Mitglieder dieser Gruppe. Die Berechtigungen für das Verzeichnis sind 0770, aber niemand kann in das Verzeichnis schreiben. Was habe ich falsch gemacht?

Haben Sie einen der folgenden Parameter in der Freigabe definiert?

```
writable = yes
writeable = yes
read only = no
```

Denken Sie daran, dass eine Freigabe standardmäßig mit Nur-Lese-Rechten besetzt ist und Sie das Schreibrecht erst aktivieren müssen, bevor jemand in sie schreiben kann, unabhängig von den Verzeichnis- oder Dateiberechtigungen in der Freigabe.

Ich habe eine neue Freigabe namens kits definiert, aber niemand kann sich mit ihr verbinden. Einige Benutzer erhalten die Fehlermeldung: »Das angegebene Freigabeverzeichnis kann nicht gefunden werden.« Andere bekommen ein Dialogfeld, das besagt: »Kann nicht auf \\server\kits zugreifen ...« Was könnte das Problem sein?

Überprüfen Sie die Pfadangabe in Ihrem Freigabeabschnitt für `[kits]`. Wenn der Pfad nicht existiert oder die Benutzer nicht darauf zugreifen dürfen, erhalten sie diese Art von Fehlermeldungen.

Wie klein kann ein [homes]-Abschnitt sein? Wenn die Freigabe nicht browsable ist, braucht sie doch eigentlich keinen Kommentar.

Sie müssen die Freigabe zum Beschreiben freigeben, damit Benutzer zumindest in ihre Home-Verzeichnisse schreiben können. Der kleinste `[homes]`-Abschnitt hat also zwei Zeilen, z.B.:

```
[homes]
writable = yes
```

Wie würden Sie verhindern, dass Dateien, die in der [homes]-Freigabe erstellt werden, allgemein lesbar sind?

Hier müssen Sie die Parameter `create mode` und `directory mode` verwenden, um die korrekte Handhabung von Dateien und Verzeichnissen zu garantieren. Fügen Sie also einfach den folgenden Eintrag in Ihre `[homes]`-Freigabe ein:

```
create mode = 0750
directory mode = 0740
```

Dies verhindert alle allgemeinen Berechtigungen.

Wie würden Sie sicherstellen, dass nur die Rechner A, B und C auf die Dateifreigabe [docs] zugreifen können?

Um sicherzustellen, dass nur die Rechner A, B und C auf eine Dateifreigabe zugreifen können, fügen Sie der Freigabe einfach eine `hosts-allow`-Aussage hinzu. Denken Sie daran, dass Sie Namen oder IP-Adressen, Netzgruppen usw. benutzen können. In Ihrem Fall fügen Sie also folgendes in die Freigabe `[docs]` ein:

```
hosts allow = A B C
```

Woche 2

Es geht weiter ...

Tag 8

Drucker

von Richard Sharpe

In Kapitel 7, »Dateifreigaben«, haben Sie sich angesehen, wie Dateifreigaben konfiguriert werden. In diesem Kapitel werden Sie die Details zum Drucken über Samba kennenlernen. Sambas Philosophie zum Thema Drucken lautet: Wenn Unix es drucken kann, kann Samba es auch.

Das Drucken mit Samba umfasst folgende Bereiche:

- ✘ Drucksysteme
- ✘ Konfiguration der Druckerfreigabe
- ✘ Automatische Installation des Druckertreibers
- ✘ Drucken von Unix an Windows-Systemen

Abbildung 8.1 zeigt, wie vielseitig Samba sein kann, da es das Drucken von Windows- (und DOS-)Clients an Drucker, die an Unix-Systeme angeschlossen sind, unterstützt und Unix-Systemen ermöglicht, an Drucker zu drucken, die mit Windows-Systemen verbunden sind.

Ich werde jeden dieser Bereiche darstellen und das Drucken auf Ihrem Samba-Server konfigurieren.

Wenn Sie bis jetzt Kapitel 7 noch nicht gelesen haben, sollten Sie noch einmal zurückgehen und es nachholen, da ich hier auf vielen der Konzepte aufbaue, die dort dargestellt sind.

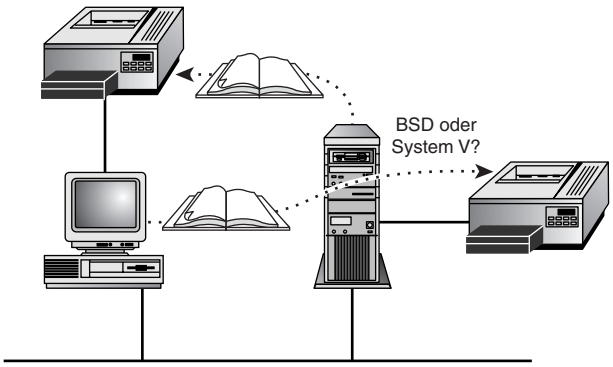


Abb. 8.1: Drucken mit Samba in einer vielseitigen Umgebung

8.1 Samba und Drucken

Aus dem Blickwinkel eines CIFS/SMB-Clients umfasst das Drucken das Öffnen einer Datei in einer speziellen Dateifreigabe, das Schreiben in diese Datei und das Schließen der Datei. Was nach Schließen der Datei mit ihr passiert, interessiert den Client nicht weiter, aber Benutzer wollen in der Regel, dass diese Datei gedruckt wird. Wenn Sie eine Druckerfreigabe unter Samba konfigurieren, kümmert Samba sich um das Drucken der Dateien. Abbildung 8.2 bietet einen detaillierten Blick auf das Drucken von einem CIFS/SMB-Client.

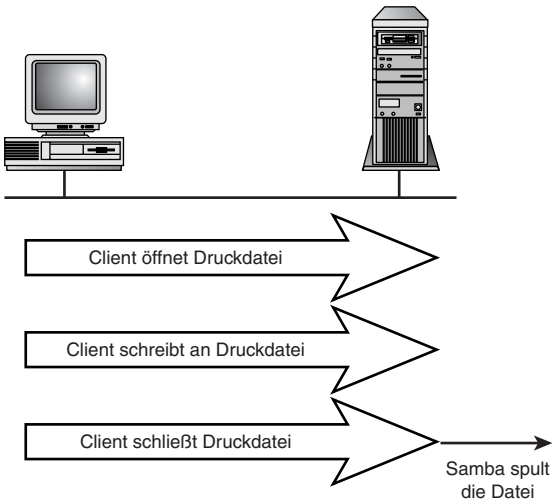


Abb. 8.2: Ein detaillierterer Blick auf das Drucken von einem Client

Wie Sie in Abbildung 8.2 sehen, durchläuft ein Client beim Drucken einer Datei folgende Schritte:

1. Der Client öffnet eine Datei in der Druckerfreigabe zum Beschreiben. Daher muss der Server Platz im Dateisystem zur Verfügung stellen, in dem die Datei gespeichert werden kann.
2. Der Client schreibt die Druckdatei. Er kann jegliche CIFS/SMB-Operationen benutzen. Ein bösartiger Client kann viele Daten schreiben, dann an den Anfang der Datei zurückgehen und einige Daten überschreiben.
3. Der Client schließt die Datei, und ab diesem Punkt sendet der Server den Druckauftrag an das Drucksystem (spoolt den Auftrag).

Einige Clients, wie z.B. Windows 95, öffnen Druckdateien in Druckerfreigaben mit leeren Dateinamen (d.h. " "). Lassen Sie sich hiervon nicht verwirren, wenn Sie eine `smbd`-Logdatei durchsehen.

Abgesehen vom Drucken der Dateien wollen Clients oft auch den Status von Druckerwarteschlangen einsehen können. Samba unterstützt dies, indem es Informationen zur Druckerwarteschlange ausgibt, wenn ein Client diese Informationen verlangt.

Da also eine Druckerfreigabe im Wesentlichen eine Dateifreigabe mit einigen zusätzlichen Attributen ist, wissen Sie bereits das meiste, das Sie für die Erstellung einer Druckerfreigabe wissen müssen. Fast das Einzige, das Sie einer Dateifreigabe hinzufügen müssen, ist der Parameter `printable`. Hier ist ein erster Versuch für eine Druckerfreigabe:

```
[first-printer]
   comment = Mein erster Drucker
   path = /var/spool/samba
   printable = yes
```

Warum braucht eine Druckerfreigabe eine Pfadangabe? Nun, die Dateien, die der Client druckt, müssen an irgendeinen Platz im Dateisystem gespeichert werden, während sie geschrieben werden. Normalerweise wäre dieses Verzeichnis allgemein beschreibbar und hätte das Sticky Bit (t) gesetzt, damit niemand Dateien löschen kann, die ihm nicht gehören.

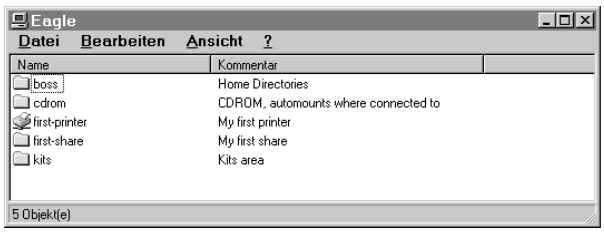


Abb. 8.3: Ihre erste Druckerfreigabe first-printer in der Netzwerkumgebung

Wenn Sie Samba nun neu starten, was sehen Sie? Abbildung 8.3 zeigt, was Sie in Windows 9x oder Windows NT in der Netzwerkumgebung sehen können, wenn Sie auf `EAGLE` doppelklicken.

Wie Sie sehen, wird Ihr Drucker angezeigt. Wenn Sie auf Ihrem Client einen Drucker konfigurieren und etwas an diese Freigabe übertragen, was passiert dann? Sie treffen auf eine Menge Probleme. Das erste ist, dass Samba annimmt, dass der mit dieser Freigabe verbundene Drucker `first-printer` heißt. Da der Drucker an diesem Punkt wahrscheinlich nicht existiert, endet die Datei, die Sie drucken wollen, in dem über den Pfadparameter definierten Verzeichnis und sitzt dort. Nachfolgend finden Sie ein detailliertes Listing des Verzeichnisses `/var/spool/samba`, nachdem eine Datei an `first-printer` übertragen wurde:

```
ls -al /var/log/samba
total 14
drwxrwxrwt   2 root    root    1024 Jan  8 12:09 .
drwxr-xr-x  14 root    root    1024 Dec 30 15:45 ..
-rwxr--r--   1 rsharpe sharpe 12240 Jan  8 11:59 rjspc1.a00652
```

Um dieses Problem zu beheben, können Sie den Parameter `printer` einfügen, um Samba mitzuteilen, welchen Drucker es benutzen soll. Um z.B. Druckaufträge an lp weiterzuleiten, benutzen Sie in der Druckerfreigabe folgenden Parameter:

```
printer = lp
```

Wenn Sie Ihrer Freigabe `first-printer` einen solchen Parameter hinzufügen und der Druckertyp der gleiche ist wie der auf Ihrem Client definierte (d.h. der Client hat den korrekten Treiber), sollten Sie nach einem Neustart von Samba den Drucker sehen können, wenn Sie an diese Warteschlange drucken.

Leider wird der erste Auftrag, den Sie an den Drucker übertragen haben, nie in die Warteschlange gestellt, da Samba einen Druckauftrag nur weiterleitet, wenn er geschlossen wird. Der erste Auftrag sitzt einfach in dem durch den Pfadparameter definierten Verzeichnis, bis Sie ihn löschen.

Wenn Sie überprüfen wollen, ob das Drucken tatsächlich funktioniert, stoppen Sie die Warteschlange auf Ihrem Samba-Server, bei einem BSD-basierten Drucksystem z.B. über den Befehl:

```
lpc stop lp
```

Überprüfen Sie dann den Drucker von Ihrem Client aus. Unter Windows 9x oder Windows NT sollten Sie etwas Ähnliches sehen wie in Abbildung 8.4.

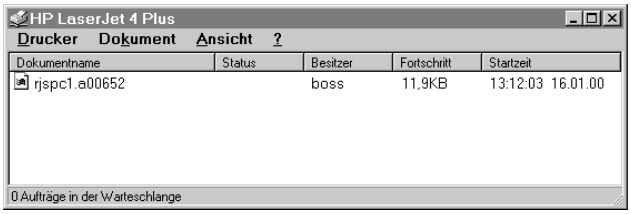


Abb. 8.4: Warteschlangeninformationen für Ihre Druckerfreigabe

Denken Sie daran, die `lp`-Warteschlange neu zu starten, damit zukünftige Druckaufträge gedruckt werden.

Das Drucken kann so einfach, aber auch eine sehr komplexe Aufgabe sein. Möglicherweise müssen Sie sich Gedanken machen über das von Ihrem Samba-Server benutzte Drucksystem, über Druckertreiber, die PostScript mit der berüchtigten vorangestellten Zeichenkette Strg-D (^D) generieren, und eine Menge anderer Probleme. Viele dieser Probleme werden in den folgenden Abschnitten dargestellt.

8.2 Unterstützte Drucksysteme

Samba benutzt die Druckbefehle des Betriebssystems, um eine Datei zu drucken, die an eine Druckerfreigabe übertragen wurde. Es verwendet außerdem die anderen Befehle, die das Drucksystem bietet, um den Status von Druckerwarteschlangen und Druckaufträgen abzufragen, Druckerwarteschlangen anzuhalten, neu zu starten usw.

Es gibt jedoch viele verschiedene Drucksysteme unter Unix. Abgesehen von den ursprünglichen BSD und System V, die unterschiedliche Befehle zum Ausführen druckrelevanter Aufgaben verwenden, gibt es auch *PLP (Portable Line Printer)* und *LPRNG*, die beide auf dem BSD-Ansatz mit einigen Verbesserungen basieren. PLP und LPRNG wurden vom gleichen Autor, Patrick Powell, entwickelt.

Zusätzlich zu diesen Drucksystemen haben die beliebten Unix-Varianten AIX und HPUX ihre eigenen Drucksysteme. Außerdem unterstützt Samba das Drucken für QNX und SOFTQ.

Wenn Samba kompiliert wird, richtet es das Standarddrucksystem ein, indem es folgendermaßen nach Makros sucht (siehe `$SRCDIR/include/includes.h`):

1. Bei einem AIX-System wird das Drucksystem auf AIX eingestellt.
2. Bei einem HPUX-System wird das Drucksystem auf HPUX eingestellt.

3. Bei einem QNX-System wird das Drucksystem auf QNX eingestellt.
4. Bei einem System-V-System wird das Drucksystem auf SYSV eingestellt.
5. Sonst wird das Drucksystem auf BSD eingestellt.

Dies funktioniert für die meisten Systeme, muss aber geändert werden, wenn Ihr System nicht AIX, HPUX, QNX oder System V ist, aber das System-V-Drucksystem benutzt. Es muss auch geändert werden, wenn Sie PLP, LPRNG oder SOFTQ verwenden.

Wenn Sie das Standarddrucksystem tatsächlich ändern müssen, fügen Sie einfach folgenden Eintrag in den globalen Abschnitt ein:

```
printing = <Ihre Drucksystem-Auswahl>
```

Ist Ihr System wirklich außergewöhnlich und entspricht es keinem der Drucksysteme, müssen Sie möglicherweise individuelle Druckbefehle einrichten, damit das Drucken korrekt funktioniert.

8.3 Die [printers]-Freigabe

Wenn Sie jedesmal eine neue Druckerfreigabe einrichten müssten, wenn Sie Ihrem Samba-Server einen neuen Drucker hinzufügen wollen, könnte die Administration eines Samba-Servers kompliziert werden. Um Ihr Leben zu vereinfachen, bietet Samba ein Schema für Drucker, das der in Kapitel 7 beschriebenen `[homes]`-Freigabe ähnlich ist. Kurz: Wenn ein Client eine Verbindung zu einer Freigabe verlangt, folgt Samba diesem Ansatz:

1. Die Freigabe wird in der `smb.conf` gesucht und, falls vorhanden, zurückgegeben.
2. Wird die Freigabe nicht gefunden, gibt es aber einen `[homes]`-Abschnitt, sucht Samba in der `passwd`-Datei einen Benutzer mit dem gleichen Namen wie dem der verlangten Freigabe. Wird dieser gefunden, wird er als Freigabe zurückgegeben.
3. Ist auch dies nicht erfolgreich, existiert aber ein `[printers]`-Abschnitt, sucht Samba nach einem Drucker mit dem gleichen Namen wie die verlangte Freigabe und gibt dies als Freigabe zurück.
4. Findet Samba auch hier die Freigabe nicht, sucht es nach einer Standardfreigabe und gibt diese, falls vorhanden, zurück.
5. Ist auch dies nicht erfolgreich, gibt Samba eine Fehlermeldung aus, die besagt, dass der Netzwerkname nicht verfügbar ist.

Die [printers]-Freigabe sieht aus wie jede andere Druckerfreigabe. Tatsächlich können Sie diese zu einer [printers]-Freigabe machen:

```
[printers]
   comment = Alle Drucker in dieser Freigabe, aus printcap
   path = /var/spool/samba
   printable = yes
```

Wenn Samba die [printers]-Freigabe benutzt, weil die verlangte Freigabe nicht als Abschnitt existiert und nicht als [homes]-Freigabe aufgelöst werden kann, wird ein Klon der [printers]-Freigabe erzeugt, dem der Name der verlangten Freigabe und des verlangten Druckers gegeben wird. Das heißt, dass alle Drucker, die durch den [printers]-Abschnitt definiert werden, ihre Parameter aus dem [printers]-Abschnitt holen.

Woher bekommt Samba die Liste der Drucker? Aus der printcap-Datei. Basiert Ihr Drucksystem auf BSD oder benutzt es PLP oder LPRNG, finden Sie die printcap-Datei unter /etc/printcap. Benutzt Ihr System dagegen das System-V-Drucksystem, können Sie eine printcap-Datei erstellen oder den Parameter printcap name verwenden, der später in diesem Kapitel dargestellt wird.

8.4 Druckrelevante Parameter

Die folgenden Parameter beeinflussen auf die eine oder andere Weise die Funktionsweise der Druckerfreigaben. Die meisten Samba-Administratoren verwenden nicht viele dieser Parameter. Wie immer finden Sie die komplette Auflistung der Parameter und das letzte Wort zu ihren Funktionen in den Manpages zur smb.conf für die aktuellste Samba-Version. Sie können den Befehl man smb.conf verwenden, um einen Blick auf die Parameter zu werfen.

Viele der Parameter, die in den nächsten Abschnitten aufgelistet sind, nehmen Variablen wie %p, %j usw. an, die bei Ausführung der Befehle durch die druckrelevanten Informationen ersetzt werden. Tabelle 8.1 zeigt die Bedeutung vieler dieser Variablen.

Tabelle 8.1: Ersetzungen für Druckervariablen

Variable	Beschreibung
%p	Ersetzen durch Druckernamen
%j	Ersetzen durch Auftragsnummer
%s	Ersetzen durch vollen Pfadnamen für Spool-Datei.
%f	Ersetzen durch Namen der Spool-Datei (ohne Pfad)

load printers

Dieser globale Parameter definiert, ob Samba alle Drucker zum Browsen in die `printcap`-Datei lädt.

Der Standardwert für diesen Parameter ist `yes`, d.h. standardmäßig sind alle Drucker in Ihrer `printcap`-Datei für das Browsing verfügbar. Wenn Sie dies nicht wollen, fügen Sie einfach folgenden Eintrag in den globalen Abschnitt Ihrer `smb.conf` ein:

```
load printers = no
```

lppause command

Dieser Parameter definiert den Befehl, den Samba ausführt, um das Drucken eines bestimmten Druckauftrags anzuhalten. Dies sollte ein Befehl oder ein Skript sein, der bzw. das einen Warteschlangennamen und eine Jobnummer annimmt und den Auftrag anhält.

Dieser Parameter hat nur bei den Drucksystemen SysV und SOFTQ einen Standardwert.

Detaillierte Informationen finden Sie in den Manpages zur `smb.conf`.

lpq cache time

Dieser globale Parameter kontrolliert, wie lange `lpq`-Informationen zwischengespeichert werden. Er verhindert, dass `lpq command` zu oft aufgerufen wird. Der Wert wird in Sekunden ausgedrückt.

Der Standardwert für diesen Parameter ist 10 Sekunden.

lpq command

Dieser Parameter definiert den Befehl, den Samba ausführt, um Statusinformationen zur Druckerwarteschlange für Clients zu erhalten. Dies sollte ein Programm oder ein Skript sein, das einen Warteschlangennamen annimmt und Statusinformationen über den Drucker ausgibt.

Der Standardwert für diesen Parameter hängt vom Wert des Parameters `printing` ab.

lpresume command

Dieser Parameter definiert den Befehl, den Samba ausführt, um einen Druckauftrag für Clients neu zu starten oder fortzusetzen. Dies sollte ein Programm oder ein Skript sein, das einen Druckernamen und eine wieder aufzunehmende Auftragsnummer annimmt. Der Parameter bewirkt das Gegenteil von `lppause command`.

Dieser Parameter hat nur für die Drucksysteme SysV oder SOFTQ einen Standardwert.

Detaillierte Informationen finden Sie in den Manpages zur `smb.conf`.

lprm command

Dieser Parameter definiert den Befehl, den Samba ausführt, um einen Druckauftrag für Clients zu löschen. Dies sollte ein Programm oder ein Skript sein, das einen Druckernamen und eine zu löschende Auftragsnummer annimmt.

Der Standardwert für diesen Parameter hängt vom Wert des Parameters `printing` ab.

Ein Beispiel:

```
lprm command = /usr/bin/lprm -P%p %j
```

Mit dieser Einstellung benutzt `lprm command /usr/bin/lprm` und erhält einen Warteschlangennamen und die Auftragsnummer.

min print space

Dieser Parameter definiert den mindestens auf der Festplatte zur Verfügung stehenden Speicherplatz, damit Clients den Druckauftrag durchführen können. Er wird in Kilobyte spezifiziert. Ein Wert von `0` (der Standardwert) heißt, dass Aufträge immer durchgeführt werden, unabhängig von freiem Platz auf der Festplatte.

postscript

Dieser Parameter legt fest, dass Samba Druckdateien als PostScript interpretieren soll. Samba fügt dann einen PostScript-Kommentar (`%!`) an den Anfang des Druckauftrags ein, womit Probleme mit PCs behoben werden, die darauf bestehen, die Zeichenkette Strg+D an den Anfang der Druckdaten zu setzen. Dies verwirrt PostScript-Drucker.

Der Standardwert für diesen Parameter ist `false` oder `no`.

print command

Dieser Parameter definiert den Befehl, den Samba ausführt, um einen Druckauftrag auszuführen. Dies sollte ein Programm oder ein Skript sein, das einen Druckernamen und einen Dateinamen annimmt und die Datei an den Drucker weiterleitet.

Der Befehl `print` muss mindestens eine der Variablen `%s` oder `%f` und kann die Variable `%p` enthalten.

Der Standardwert für diesen Parameter hängt vom Wert des Parameters `printing` ab.

Ein Beispiel:

```
print command = /usr/local/samba/bin/localprintscrip %p %s
```

Mit dieser Einstellung wird ein lokales Skript namens `localprintscript` aufgerufen, dem der Warteschlangenname und die Auftragsnummer übergeben wird.

printable

Dieser Parameter definiert, dass eine Freigabe eine Druckerfreigabe ist, über die ein Client Druckdaten an das in der Freigabe definierte Verzeichnis übertragen kann. Wenn eine Freigabe als `printable` bestimmt ist, ist `sie` standardmäßig auch `writable`. Jeglicher `read-only`-Parameter gilt nur für nicht druckenden Zugriff auf die Freigabe.

Der Standardwert für diesen Parameter ist `no`; standardmäßig sind Freigaben also nicht `printable`. Das heißt, Clients können sich auf diesem Weg nicht als Druckerfreigaben verbinden.

Um eine Freigabe für das Drucken einzurichten, fügen Sie einfach folgenden Eintrag in den Abschnitt für die Freigabe ein:

```
printable = yes
```

printcap name

Dieser Parameter (und sein Synonym `printcap)` wird benutzt, um Samba den Standort der `printcap-Datei` mitzuteilen, in der nach Druckern gesucht wird, wenn die `[printers]`-Freigabe verwendet wird.

In System-V-Systemen, die für eine Auflistung verfügbarer Drucker `lpstat` verwenden, können Sie den Parameter `printcap name` auf `lpstat` setzen, um automatisch eine Liste aller verfügbaren Drucker zu erhalten.

Der Standardwert für diesen Parameter ist `/etc/printcap`.

Ein Beispiel:

```
printcap name = /etc/myprintcap
```

Mit dieser Einstellung benutzt Samba die Datei `/etc/myprintcap`, wenn es nach Druckern sucht.

printer

Dieser Parameter teilt Samba den Namen des Druckers mit, an den Druckaufträge übertragen werden, wenn der Client die Druckdatei geschlossen hat.

Dieser Parameter hat keinen Standardwert. Einige Beispiele:

```
printer = lp
printer = hplj4
```

printer driver

Dieser Parameter definiert, welchen Treibernamen Samba an Clients übergibt, die nach dem mit einem Drucker verbundenen Treiber fragen. Er wird mit der automatischen Installation von Druckertreibern verwendet, die später in diesem Kapitel dargestellt wird.

Dieser Parameter hat keinen Standardwert. Ein Beispiel:

```
printer driver = HP LaserJet 4 Plus
```

printer driver files

Dieser Parameter teilt Samba den Standort der Druckertreiber-Datei mit, die benutzt wird, wenn Treiber an Windows-9x-Clients übertragen werden.

Die Datei wird aus einer Windows-9x-`msprint.def`-Datei erstellt, wie es im Abschnitt »Automatische Treiberinstallation« beschreiben wird.

Der Standardwert für diesen Parameter ist `SAMBA_INSTALL_DIRECTORY/lib/printers.def`.

printer driver location

Dieser Parameter teilt Samba mit, welche Freigabe es übergeben soll, wenn Clients bei der automatischen Installation von Druckertreibern auf Windows-9x-Rechnern nach dem Standort der Treiberdateien fragen. Dies wird ausführlicher im Abschnitt »Automatische Treiberinstallation« dargestellt.

Dieser Parameter hat keinen Standardwert. Ein Beispiel:

```
printer driver location = \\%h\printers$
```

printing

Dieser Parameter teilt Samba das Drucksystem mit, das auf Ihrem Server verwendet wird. In der Regel wird das Drucksystem während der Kompilierung bestimmt, aber wenn Ihr System PLP, LPRNG oder SOFTQ benutzt

oder sehr außergewöhnlich ist (z.B. auf SysV basierend, aber LPD benutzend), müssen Sie diesen Parameter manuell einstellen.

Derzeit werden acht Drucksysteme unterstützt:

- ✘ AIX
- ✘ BSD
- ✘ HPUX
- ✘ LPRNG
- ✘ PLP
- ✘ QNX
- ✘ SOFTQ
- ✘ SYSV

Der Standardwert für diesen Parameter wird, wie oben angegeben, während der Kompilierung festgelegt.

queuepause command

Dieser Parameter definiert den Befehl, den Samba ausführt, um eine Druckerwarteschlange anzuhalten. Dies sollte ein Programm oder ein Skript sein, das einen Druckernamen annimmt und die Warteschlange für diesen Drucker stoppt.

Der Standardwert für diesen Parameter hängt vom verwendeten Drucksystem ab.

queueresume command

Dieser Parameter teilt Samba mit, welchen Befehl es benutzen soll, um Druckerwarteschlangen wieder aufzunehmen. Dies sollte ein Programm oder ein Skript sein, das einen Druckernamen annimmt und die Warteschlange für diesen Drucker wieder aufnimmt.

Der Standardwert für diesen Parameter hängt vom verwendeten Drucksystem ab.

8.5 Automatische Treiberinstallation

Windows 95 und Windows 98 unterstützen die automatische Installation von Druckertreibern über *Point and Print*. Samba implementiert die erforderliche Funktionalität, um diese automatische Installation zu unterstützen.

Obwohl die benötigten Einstellungen für die Unterstützung der Point-and-Print-Installation ausführlich im Samba-Dokumentationsverzeichnis beschrieben ist (`PRINTER_DRIVER.txt`), werde ich sie hier detailliert darstellen.

Zunächst müssen Sie eine `[printers$]`-Freigabe einrichten, in der alle Treiberdateien abgelegt werden. Diese Dateifreigabe sieht wie folgt aus:

```
[printer$]
   path = /usr/local/samba/printer
   public = yes
   writable = no
   browsable = yes
```

Denken Sie daran, dieses Verzeichnis zu erstellen, damit die Dateifreigabe die Dateien dort ablegen kann.

Der nächste Schritt besteht darin, die Druckerdefinitionsdatei zu erstellen, damit Windows 9x weiß, wie es die Drucker installieren soll, die Sie für die automatische Installation verfügbar gemacht haben. Dafür müssen Sie sich die Windows INF-Dateien `msprint.inf` und `msprint2.inf` aus dem Verzeichnis `C:\WINDOWS\INF` von Ihrem Windows-Rechner holen. Manchmal befinden sich diese Dateien auch in einem anderen Verzeichnis. Wenn Sie nicht unterstützte oder aktualisierte Treiber verwenden, müssen Sie diese Treiber erst auf Ihrem Windows-9x-System installieren, dann die Datei `oemNN.inf` kopieren und `statt msprint.inf` diese Datei benutzen.

Die Datei heißt nicht genau `oemNN.inf`, sondern hat einen ähnlichen Namen. Sie können die von Ihnen benötigte Datei finden, indem Sie in jeder derartigen Datei nach dem entsprechenden Druckernamen suchen.

Wenn Sie die Dateien auf Ihren Samba-Server kopiert haben, müssen Sie das in Samba integrierte Programm `make_printerdef` benutzen, um Ihren Drucker in die Datei `printer.def` einzutragen. Suchen Sie den exakten Namen für den Drucker, den Sie definieren (d.h. den Namen, unter dem er Windows bekannt ist), indem Sie die entsprechende INF-Datei durchsuchen. Für Drucker, deren Namen mit den Buchstaben A bis K beginnen, suchen Sie in `msprint.inf`, für andere Namen in `msprint2.inf`. Für das folgende Beispiel verwenden Sie einen Drucker des Typs HP LaserJet 4 Plus und erstellen so einen neuen `printers.def`-Eintrag:

```
make_printerdef msprint.inf "HP LaserJet 4 Plus">>
printers.def
```

Stellen Sie sicher, dass der neue Eintrag an das Ende der `printers.def`-Datei eingtragen wird. In diesen Beispielen platzieren Sie die Datei unter `/usr/local/samba/lib`.

Wenn `make_printerdef` ausgeführt wird, gibt es auf `stderr` die Dateien aus, die für die Installation benötigt werden. All diese Dateien müssen in die `[printer$]`-Freigabe kopiert werden, die Sie vorher definiert haben. Die Dateien befinden sich in der Regel alle im Verzeichnis `C:\WINDOWS\SYSTEM`. Für den HP LaserJet 4 Plus werden folgende Dateien benötigt:

```
FINSTALL.DLL, FINSTALL.HLP, HPPCL5MS.DRV, ICONLIB.DLL,
PJLMON.DLL, UNIDRV.DLL, UNIDRV.HLP
```

Zum Abschluss müssen Sie Ihrer `smb.conf` noch einige zusätzliche Parameter hinzufügen. Einer geht in den globalen Abschnitt und spezifiziert den Standort der Druckerdefinitionsdatei:

```
[global]
   ...
   printer driver file = /usr/local/samba/lib/printers.def
   ...
```

Dies ist die Datei, in die Sie alle Druckerdefinitionseinträge platzieren, die mit dem Programm `make_printerdef` erstellt werden.

Die anderen Parameter, `printer driver` und `printer driver location`, müssen Sie für jede Druckerfreigabe definieren, für die Sie automatische Treiberinstallation ermöglichen wollen. Das folgende Beispiel zeigt Ihre Freigabe `first-printer` nach den Änderungen zur Unterstützung automatischer Installation:

```
[first-printer]
   comment = Mein erster Drucker
   path = /var/spool/samba
   printable = yes
   printer driver = HP LaserJet 4 Plus
   printer driver location = \\%h\PRINTER$
```

Machen Sie sich keine Gedanken über das `%h` in der letzten Zeile. Es ist eine der Variablen, die Samba in der `smb.conf`-Datei benutzen kann. Diese werden in Kapitel 10, »Automatisierung auf Server-Seite«, ausführlicher dargestellt.

Wenn Sie alle Änderungen an Ihrer `smb.conf` durchgeführt und Samba neu gestartet haben, können Sie die automatische Installation des Treibers für den definierten Drucker ausprobieren. Rufen Sie Ihren Server in der Netzwerkumgebung auf, wie in Abbildung 8.5 dargestellt. Doppelklicken Sie