Überblick Samba - Administration

22. Juli 2005

Inhaltsverzeichnis

1 Ziel des Dokumentes

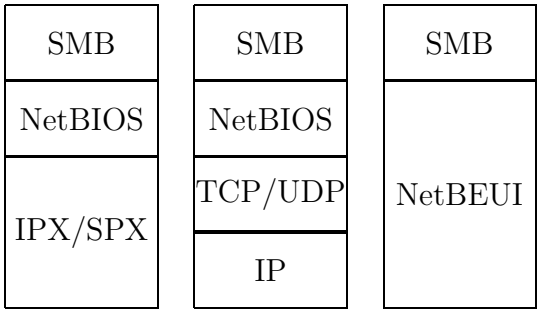
Es gibt sehr viel gute Literatur zu diesem Thema in Büchern, Zeitschriften und natürlich im Internet. Dieses Dokument soll deshalb nur eine Schnelleinstiegshilfe sein, um sichere Grundkenntnisse zu einem komplexen Thema zu erhalten und im Wege stehende Fettnäpfchen, die auch mit der neuen Version 3.0 verbunden sind, möglichst links liegen zu lassen zu können.

2 Grundlagen

2.1 Protokolle

Der Samba-Server ist ein Windows-NT/2000 - File- und Print Serverersatz auf Open Source Basis. Das zugrundeliegende Protokoll heißt SMB (Server Message Block - Protocol). Es stammt vom LAN Manager, der gemeinsamen Entwicklung der Firmen Microsoft und IBM ab.

SMB setzt bei Microsoftsystemen in der Regel auf NetBEUI auf, was unter Linux nicht verwendet werden kann. Hier kommt standardmäßig NetBIOS over TCP/IP zum Einsatz. Die folgende Darstellung soll die Hierarchie der einzelnen Protokollschichten verdeutlichen (aus [linServer]):



2.2 daemon - Komponenten

Entsprechend diesem Protokollaufbau sind die wichtigsten Funktionalitäten in zwei Programmen realisiert (aus [manSamba]):

2.2.1 a) nmbd

„... Der ***nmbd*** ist ein Server, der Anfragen zu NetBIOS-über-IP Namensdiensten verstehen und beantworten kann, wie sie von SMB-/CIFS-Clients produziert werden, z.B. Windows 95/98/ME, Windows NT, Windows 2000,

Windows XP und LanManager-Clients. Er nimmt auch an den Browsing-Protokollen teil, die in Windows die Ansicht "Netzwerkumgebungäusmachen.
Neben weiteren Diensten wird nmbd auf solche Anfragen warten, und wenn sein eigener NetBIOS-Name angegeben wird, wird er mit der IP-Adresse des Hosts antworten, auf dem er läuft. Sein ëigener NetBIOS-Nameïst per Voreinstellung der primäre DNS-Name des Hosts, auf dem er läuft, was aber durch netbios name in smb.conf überschrieben werden kann. Somit beantwortet nmbd Broadcastanfragen nach seinem (bzw. seinen) eigenen Namen. Weitere Namen, auf die nmbd antworten soll, können mit Hilfe von Parametern in der Konfigurationsdatei smb.conf(5) eingestellt werden.
nmbd kann auch als WINS-Server (Windows Internet Name Server) benutzt werden. Das heißt prinzipiell, dass er als WINS-Datenbankserver fungiert und aus den empfangenen Anfragen zu Namensregistrierungen eine Datenbank erstellt, mit der er auf Clientabfragen nach diesen Namen antwortet.

2.2.2 b) smbd

Der ***smbd*** ist der Server-Daemon, der Dateifreigaben und Druckdienste für Windows-Clients bietet. Der Server bietet Raum für Dateien und Druckerdienste für Clients mit Hilfe des Protokolls SMB (oder CIFS). Dieses ist kompatibel mit dem LanManager-Protokoll und kann LanManager-Clients bedienen. Dazu gehören MSCLIENT 3.0 für DOS, Windows for Workgroups, Windows 95/98/ME, Windows NT, Windows 2000, OS/2, DAVE für den Macintosh und smbfs für Linux.
Eine umfangreiche Beschreibung der Dienste, die der Server anbietet, ist in der Manpage zur Konfigurationsdatei enthalten, in der die Attribute jener Dienste eingestellt werden (siehe smb.conf(5)). Diese Manpage beschreibt nicht die Dienste, sondern konzentriert sich auf die administrativen Aspekte des Serverbetriebs.
Beachten Sie bitte, dass es beim Betrieb dieses Servers erhebliche Folgen für die Sicherheit gibt, und die Manpage zu smb.conf(5) sollte unbedingt gelesen werden, bevor mit der Installation begonnen wird.
Eine Sitzung wird immer dann erzeugt, wenn ein Client eine verlangt. Jeder Client erhält eine Kopie des Servers für jede Sitzung. Diese Kopie bedient dann während der Sitzung alle Verbindungen, die der Client herstellt. Wenn alle Verbindungen ihres Clients geschlossen sind, terminiert die Kopie des Servers für diesen Client.
Die Konfigurationsdatei sowie alle Dateien, die sie lädt, werden automatisch einmal pro Minute geladen, falls sie sich verändern. Sie können ein erneutes Laden erzwingen, indem Sie ein SIGHUP an den Server senden. Das erneute Laden der Konfigurationsdatei hat keinen Einfluss auf Verbindungen zu Diensten, die bereits hergestellt sind. Der Benutzer muss sich entweder von

dem Dienst trennen oder smbd muss terminiert und neu gestartet werden. ...“

3 Installation

3.1 Binäre Pakete

Jede Distribution bringt eigene, fertige Pakete mit, die ausreichend aktuell sein sollten. Man kann aber auch passende Versionen finden unter:
$\Longrightarrow$ `http://us2.samba.org/samba/ftp/Binary_Packages/`

3.2 Quelltext

Die Installation aus dem Quelltext ist etwas aufwändiger und manchmal leider nicht zu umgehen, z.B. wenn unter Debian eine Anbindung von Samba an einen LDAP-Verzeichnisdienst geplant ist.
Die Downloadadresse:
$\Longrightarrow$ `http://us2.samba.org/samba/ftp/samba-latest.tar.gz`

4 Konfiguration

Neben der Kommandozeile stehen auch grafische Tools zur Verfügung, wie das zur Samba-Suite gehörende ***swat*** oder der Allrounder ***webmin***, auf die ich an dieser Stelle nur verweisen möchte:
$\Longrightarrow$ `http://gertranssmb3.berlios.de/output/SWAT.html`
$\Longrightarrow$ `http://www.webmin.com`

4.1 Überblick

Im Verzeichnis ***/etc/samba*** liegen die Konfigurationsdateien des Servers. In der Hauptsache findet die Einrichtung in der Datei **smb.conf** statt. Diese Datei ist wie eine ini-Datei von Microsoft aufgebaut und weicht inhaltlich je nach Linuxdistribution unterschiedlich stark vom Originalbeispiel der Samba-Entwickler ab, sollte jedoch immer eine minimale Funktionalität bieten:

- sinnvolle globale Einstellungen im Abschnitt ***[global]***
- Freigabe aller vorhandenen Unix-Heimatverzeichnisse durch ***[homes]***
- Freigabe aller am Rechner eingerichteten Drucker durch ***[printers]***

4.2 Sicherheitsmodis (security level)

Der Server kann in Bezug auf die Authentifizierungsmethode in verschiedenen Modis arbeiten, die dann von der Arbeitsweise her, den gesamten Server betreffen. Dieser Modus wird im Abschnitt ***[global]*** mit dem Parameter «***security = ...*** » angegeben. Prinzipiell gibt es nur zwei Modis: den Share-Level und den User-Level. Die unten beschriebenen Modis Server-, Domain- und ADS-Level sind nur Erweiterungen des User-Levels. Mögliche Einstellungen sind hier (Siehe auch unter [SMBlevel]):

security = share Diese unsichere, alte Funktionsweise gleicht dem Zugreifen auf Shares, wie zu Zeiten von „Windows for Workgroups“. Sie sollte nicht mehr verwendet werden.

security = user Dieser „user level“ ist standardmäßig voreingestellt. Samba verhält sich dadurch wie ein Windows NT - Anmeldeserver. Ein Benutzer muss sich also mit Benutzerkennung und Passwort überhaupt erst einmal anmelden, bevor er irgendein Share oder einen Drucker zu Gesicht bekommt. Dieses restriktive Verhalten lässt sich aufweichen, wenn man im globalen Bereich «***map to guest = bad user*** » setzt, dann kann jeder die sichtbaren Freigaben wenigstens sehen. (Er arbeitet dann als Gastbenutzer „nobody“, bedingt durch die Einstellung «***bad user = nobody***»

security = server Der Modus arbeitet wie der „user level“, mit dem Unterschied, dass der Samba-Server die Authentifikation einem speziellen Passwortserver überlässt. Seitdem Samba als Domänen-Mitglied betrieben werden kann, ist dieser Modus nicht mehr von Wichtigkeit, auch aus Sicherheitsgründen ist davon abzuraten.

security = domain Wenn Samba im diesem Modus betrieben wird, hat der Server einen Trust Account (Maschinen-Account) und reicht alle Authentifizierungsanfragen an die Domänencontroller weiter. Mit anderen Worten: Diese Konfiguration macht den Samba-Server zu einem Domänen-Mitglied.

security = ads Wenn Sie ADS benutzen und mit Samba 3 starten, können Sie der ADS als normales Active-Directory-Mitglied beitreten. Warum Sie das tun sollten? Ihre Sicherheitsrichtlinien könnten die NT-kompatiblen Authentifizierungsprotokolle schlichtweg verweigern. Wenn alle Server in Ihrem Netzwerk Windows 2000 und höher nutzen, würde Samba als NT4-artiges Domänen-Mitglied NT-kompatible Authentifizierungsdaten benötigen. Samba im AD-Mitgliedsmodus allerdings kann auch Kerberos-Tickets auswerten.

4.3 Ein einfaches Beispiel

Das folgende Listing zeigt eine einfache, kommentierte Konfiguration, in der schon ein paar Sicherheitsoptionen gesetzt sind. Sie stellt angeschlossene Drucker und die Unix-Heimatverzeichnisse zur Verfügung. Kommentarzeilen können mit «#» (Raute) oder mit «;» (Semikolon) eingeleitet werden.

```
[global]
   # Assoziation mit einer Arbeitsgruppe/Domain:
   workgroup = SMBNET

   # Die sichtbare Server-Beschreibung mit den
   # Zeichenkettensubstitutionen %h (Hostname) und
   # %v (Version), siehe dazu 'man smb.conf':
   server string =  Testserver auf %h (Samba %v)

   # Kein Arbeiten mit root-Rechten erlauben:
   invalid users = root

   # Verbindungen nur lokal und via Ethernet-
   # Karten erlauben:
   bind interfaces only = yes
   interfaces = lo, eth*

   # Oft bringt diese Option mehr Performance:
   socket options = TCP_NODELAY

   # Drucksystem auswaehlen:
   printing = CUPS

   # Deutsche Umlaute korrekt darstellen (bei
   # SuSE nicht erforderlich, da voreingestellt):
   dos charset = iso8859-15
   unix charset = iso8859-15
   display charset = iso8859-15

[homes]
   # Verwendete Bezeichnung:
   comment = Home Directories

   # Dateirechte so auf die home-Verzeichnisse legen,
   # dass die Befehle cd und ls abgelehnt werden:
   valid users = %S

   # Schreibrecht erteilen:
```

```
   read only = No

   # Nicht sichtbar im Netz:
   browsable = no

[printers]
   # Verwendete Bezeichnung:
   comment = All Printers

   # Die Warteschlange; das Verz. muss Schreibrecht
   # fuer 'others' und das Sticky-Bit haben:
   path = /var/spool/samba

   # Voraussetzung fuer Druckbetrieb:
   printable = yes

   # Nicht sichtbar im Netz:
   browseable = no

   # Gastzugriff erlauben:
   guest ok = yes
```

Diese Konfigurationsdatei sollte man stets mit dem Kommando ***testparm*** auf richtige Syntax überprüfen lassen. Aufruf:

```
testparm /etc/samba/smb.conf
```

Oder einfach (wenn der Standard-Pfad stimmt):

```
testparm
```

In der Serverkonfiguration sind viele nützliche Standardeinstellungen gesetzt worden, die man allerdings in der Datei nicht sieht und auch nicht mit einfachem ***testparm***-Aufruf ausgegeben bekommt. Dazu benötigt man die Option «***-v***» (verbose - ausführlich). Mit dieser Option kann man dann auch prima nach bestimmten Parametern suchen, z.B. nach dem verwendeten Security-Level:

```
root@srv01:~$ testparm -v | grep 'security ='
Load smb config files from /etc/samba/smb.conf
Processing section "[homes]"
Processing section "[printers]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions

        security = USER
        paranoid server security = Yes
```

Oder, um auf eine andere wichtige Voreinstellung zu kommen, kann man nach dem Schlüsselwort ***encrypt*** in der Ausgabe des Testkommandos suchen:

```
root@srv01:~$ testparm -v | grep encrypt
Load smb config files from /etc/samba/smb.conf
Processing section "[homes]"
Processing section "[printers]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions

        encrypt passwords = Yes
        update encrypted = No
```

Die gefundene Zeile «***encrypt***» schaltet für die gesamte Kommunikation Verschlüsselung ein. Das bedeutet für die verwendeten Clients, dass sie dies beherrschen müssen. Hier sieht man z.B. mit Windows 95 und Windows NT ohne Service Pack 3 alt aus, da diese Betriebsysteme Benutzernamen und Passworte unverschlüsselt im Netz senden!

4.4 (Re-)Starten der Dienste

Wie oben erwähnt, besteht Samba aus zwei Programmen, die beide gestartet werden müssen. Bei ***Debian*** erledigt das ein Skript:

```
/etc/init.d/samba start
```

Falls der Server bereits läuft sowie nach grundlegenden Änderungen der ***smb.conf*** ist ein Neustart durchzuführen mit:

```
/etc/init.d/samba restart
```

Unter ***SuSE*** sind zwei Skripte zu starten:

```
/etc/init.d/nmbd start
/etc/init.d/smbd start
```

Kleine Änderungen (Hinzufügen eines Shares) bedürfen seit der Version 3.0 keinen Serverneustart mehr. Der Daemon liest die Konfigurationsdatei automatisch ein.
Ob die Angelegenheit von Erfolg gekrönt war, kann man zu allererst den Logfiles ***/var/log/samba/log.nmbd*** und ***/var/log/samba/log.smbd*** entnehmen.

4.5 Testen des Servers

Der erste Verbindungstest sollte lokal von der Linux-Maschine aus erfolgen:

```
smbclient -L localhost -U%
```

Diese Zeile startet den Linuxclient, befragt den Server ***localhost*** und schaltet die User-Authentifizierung mit «***-U%***» ab. Die komplette Browseliste wird ausgegeben.

Ein weiterer Test von der Kommandozeile aus wäre jetzt mit dem Servernamen und z.B. mit dem Unix-Benutzer ***tux*** möglich. Dabei wird bei der Anmeldung nach dem Passwort gefragt:

```
root@srv01:~$ smbclient //srv01/homes -U tux
Password:
Domain=[LNX2] OS=[Unix] Server=[Samba 3.0.14a-Debian]
smb: \> ls
  .                   D        0  Tue Jul  5 12:41:13 2005
  ..                  D        0  Tue May 10 07:39:15 2005
    .bash_history     H       32  Tue Jul  5 12:41:13 2005

          blocks of size 262144. 1983 blocks available
smb: \>
```

Dieser Client arbeitet ähnlich wie ein einfacher FTP-Client. Versuchen Sie es einfach mal mit «***help***». Verlassen können Sie das Programm mit «***exit***». Interessant ist nun natürlich. ob Windows-Clients den Server benutzen können.

Am schnellsten ausprobieren lässt sich dies, wenn man dort unter «START ⇒ AUSFÜHREN ⇒»in das Textfeld schreibt:

```
\\192.168.0.1\tux
```

und mit «ENTER» bestätigt.

4.6 Erweiterung des Beispiels

Um wenigstens ein Public-Share zu haben, das allen einen Zugang bietet, reicht es nicht wie für Linux-Clients aus, dass ein Share ans Ende der smb.conf wie folgt gesetzt wird,

```
[pub]
   # Verzeichnis fuer schnellen Austausch:
   path = /tmp

   # Gastzugriff erlauben:
   guest ok = yes

   # Schreibrecht erteilen:
   read only = no
```

für Windows-Clients muss ein Mapping des Gastbenutzers auf ***nobody*** vorgenommen werden. Der Abschnitt «***[global]***» der Datei `smb.conf` ist deshalb noch um die Zeile

```
map to guest = bad user
```

zu erweitern.
Zusammen mit der Default-Einstellung «***guest account = nobody***», die Sie zu sehen bekommen, wenn Sie mit

```
testparm -v |grep nobody
```

die Konfiguration untersuchen, geschieht ein „Öffnen“ des Servers für anonymous. Browseable Shares werden sichtbar, auf Shares mit der Eigenschaft «***public = yes***» wird Zugriff unter Verwendung der Rechte des mit «***guest account = ...*** » angegebenen Benutzers gewährt. Der Standardbenutzer (= Standardgast) ist hierbei nobody. Dies ist wie oben gezeigt voreingestellt.
HINWEIS: Es kann an dieser Stelle zu Problemen kommen, wenn die Datei ***/etc/samba/smbpasswd***, die die Samba-Benutzer führt, mit Hilfe des Konvertierungstools ***mksmbpasswd*** erzeugt wurde.
Jetzt kann man einfach auf den Server zugreifen mit:

```
\\192.168.0.1
```

und bekommt alle „browseable“ Shares zu sehen. Auf Shares mit «***public = yes***» bzw. mit dem Synonym «***guest ok = yes***» erhält man Zugriff.
Testen kann man das auch in neueren KDE-Umgebungen, wenn im Browser ***konqueror*** als URL «***smb://192.168.0.1***» angegeben wird.

4.7 Samba-Benutzer

Die eben beschriebene Möglichkeit ist für anonyme Gastzugänge ausreichend. Sollen sich aber Benutzer mit Kennung und Kennwort auf verschlüsseltem Wege am Server anmelden, muss Samba der Umgang mit dem Authentifizierungssystem von Windows beigebracht werden.
Windows- und Linuxbenutzerverwaltung sind **nicht** kompatibel.
Das Problem lässt sich grundsätzlich auf zwei Arten lösen:

Lokale, doppelte Buchführung: Beide Systeme werden parallel auf dem Linuxsystem geführt. In der Praxis sind das zwei Datenbanken, die Dateien «***/etc/passwd***» und »***/etc/samba/smbpasswd***« mit jeweils gleichen Benutzern. Diese Methode ist einfach zu realisieren und bietet einige Sicherheit.

Zentrale Verwaltung: Benutzer werden in ein zentrales Verzeichnis eingetragen, das lokal oder entfernt sein kann. So kann zum Beispiel der Open Source Verzeichnisdienst LDAP mit beiden Authentifizierungssystemen sehr gut umgehen. Die Einrichtung eines solches Servers ist jedoch mit bedeutend höherem Planungs- und Administrationsaufwand verbunden.

Es gibt dafür natürlich noch weitere Server wie Novells eDirectory oder Microsofts Active Directory.

Ich möchte hier die erste Lösung etwas näher beschreiben:

Lokale, doppelte Buchführung:

Im einfachsten Falle werden die Benutzer auf dem Samba-Server lokal mit dem Passwort-Backend ***smbpasswd*** verwaltet. Wichtig ist dabei die Passwortdatei
/etc/samba/smbpasswd. Bei Verbindungsanforderungen durch einen Client wird diese Datei konsultiert und anhand ihrer Einträge (pro Benutzer eine Zeile) entschieden, ob die Anfrage berechtigt ist.

Es ist also, selbst wenn kein Unix-Login erlaubt sein soll, bei Verwendung von verschlüsselter Übertragung in jedem Falle ein Eintrag in der Datei
/etc/passwd erforderlich. Schon das Anlegen eines Sambabenutzers mit dem Programm «***/usr/bin/smbpasswd***» gelingt nicht, wenn es den betreffenden Benutzer dort nicht gibt.

Sinnvoll ist es, sich am Anfang die Frage zu beantworten, ob der Benutzer zusätzlich zum Samba-Account noch einen Unix-Account benötigt. Ist dies nicht der Fall, genügt es, einen „dummy"-Linuxbenutzer einzurichten. Das geschieht mit folg. Zeile:

```
useradd -s /bin/false -d /dev/null max
```

Dabei bedeutet die Option «***−d /dev/null***», dass kein Heimatverzeichnis für den Benutzer ***max*** zur Verfügung gestellt wird und die Option
«***−s /bin/false***», dass er auch keine Loginshell bekommt.

HINWEIS: Bei SuSE funktioniert die Angabe «***−d /dev/null***» nicht; diesen Teil der Kommandozeile weglassen und per Hand (am besten mit dem Tool ***vipw***) als Heimatverzeichnis «***/dev/null***» in die Datei ***/etc/passwd*** eintragen!

Jetzt ist der eigentliche, authorisierte (Samba-)Benutzer anzulegen:

```
root@srv01:~$ smbpasswd -a max
New SMB password:
Retype new SMB password:
Added user max.
```

Damit gibt es nun für ***max*** einen Login am Server. Ein Share der smb.conf, speziell für bestimmte Benutzer und Mitglieder der Unix-Gruppe ***smbadmin***, könnte so aussehen:

```
[share1]
   # Verzeichnis:
   path = /home/dokumente

   # Zugriff erlauben fuer:
   valid users = max susi +smbadmin

   # Schreibrecht erteilen:
   read only = no
```

Einen Benutzer ***fred*** kann man damit über die Gruppe ***smbadmin*** Schreibzugriff geben:

```
# Gruppe und Verzeichnis einrichten:
groupadd smbadmin
mkdir -m 775 /home/dokumente
chgrp smbadmin /home/dokumente

# Den Benutzer 'fred' einer weiteren
# Gruppe namens 'smbadmin' zuordnen:
usermod -G smbadmin fred

# Dem Benutzer 'fred' ein Samba-Konto geben:
smbpasswd -a fred
```

Ist die Frage von oben anders zu beantworten, sollen also die Standard-Unix-Accounts über SMB benutzt werden können (das stellt die Section ***[homes]*** zur Verfügung), muss lediglich dem z.B. bereits vorhandenen Unix-Benutzer namens ***tux*** mit folg. Zeile ein Konto und zugleich ein Passwort eingerichtet werden:

```
root@srv01:~$ smbpasswd -a tux
New SMB password:
Retype new SMB password:
Added user tux.
```

Soviel ersteinmal zur Benutzerverwaltung mit Samba. Im Kapitel „Samba als Primärer Domänencontroller“, findet sich eine spezielle Art von Konto: der Maschinenaccount, der für die Vertrauensstellung einer Workstation in einer Domäne erforderlich ist.
Auch zum Thema User-/Group-Mapping zwischen Windows und Linux mit Hilfe des Samba-Daemons ***winbind*** wäre noch viel zu sagen, ich darf hier auf [LinMag] verweisen, wo diese Thematik behandelt wird.

4.8 Namensauflösung

Die Namensauflösung muss entsprechend der verwendete Protokolle von zwei Seiten her betrachtet werden. Einmal muss das IP-basiert (/etc/hosts, DNS-Server) geschehen und zum anderen für die Windows NetBIOS-Namen (lmhosts, WINS).
Interessant ist, was folgender Test offenbart:

```
testparm -v |grep resolve
...
name resolve order = lmhosts wins host bcast
```

Mit nur einer Zeile im Abschnitt «***[global]***» wird Samba zum WINS-Server, der NetBIOS Namensanfragen in IP-Adressen auflöst:

```
wins support = yes
```

Es darf aber in einem Netz nur einen WINS Server geben! Diesbezüglich noch ein Hinweis: Samba kann **nicht** wie gezeigt als WINS Server auftreten **und zugleich** auf einen anderen (oder womöglich sich selbst!) mit der Angabe

```
wins server = 192.168.0.101
```

verweisen. Eins schließt das Andere aus!

$\Longrightarrow$ http://hjotten.de/networking/namen.aspx
$\Longrightarrow$ http://lug.krems.cc/docu/samba/ch07_03.html
$\Longrightarrow$ http://www.linux-praxis.de/lpic2/lpi201/2.209.1.html

4.9 Netzwerk-Browsing

Was ist Browsing?
Es ist sicher eine tolle Angelegenheit, die Netzressourcen in der Netzwerkumgebung unter Windows auf grafischem Wege (sog. „durchklicken“) zu finden. Es stellt die Alternative zum automatischen Zuweisen von Netzlaufwerken mit Hilfe einer Batchdatei dar. Leider funktioniert das Browsing aber nicht immer reibungslos. Was Browsing ist und wie es arbeitet, wird auf der Website von [NetBrowse] sehr gut erklärt. Hier ein Auszug:
„... Für die meisten bedeutet Browsing, dass sie die MS Windows- und Samba-Server in der Netzwerkumgebung sehen können und dass, wenn man auf das Icon eines bestimmten Servers klickt, ein Fenster geöffnet wird, in dem man die verfügbaren Freigaben und Drucker des Servers sehen kann.

Was so einfach klingt, ist in Wirklichkeit eine komplexe Interaktion verschiedener Technologien. Die dabei involvierten Technologien (oder Methoden) beinhalten Folgendes:

- MS Windows-Maschinen melden ihre Präsenz im Netzwerk an.

- Maschinen kündigen sich anderen Maschinen im Netzwerk an.
- Eine oder mehrere Maschinen fassen diese Ankündigungen lokal zusammen.
- Der Client findet die Maschine, die diese Liste von Maschinen gesammelt hat.
- Der Client ist in der Lage, den Namen der Maschine in eine IP-Adresse aufzulösen.
- Der Client ist in der Lage, sich mit einer anderen Maschine zu verbinden.

Die Samba-Anwendung, die die Verwaltung des Browsings und die Namensauflösung kontrolliert, heißt ***nmbd***. ...“

Zum Funktionsprinzip und Einstellungen siehe auch:
$\Longrightarrow$ http://wytech.de/base/samba.php3
$\Longrightarrow$ http://archiv.tu-chemnitz.de/pub/2001/0027/data/slice01.html

4.10 Printserver

Ähnlich wie mit der Abschnitt ***[homes]*** arbeitet der Abschnitt ***[printers]***. Dort werden alle am Server arbeitenden Drucker (Sie müssen am Server einen passenden Druckertreiber haben, ebenso benötigen die Windows-Clients den passenden Treiber auf ihrer Maschine dazu!) freigegeben. Voraussetzung ist, dass das Drucksystem im Abschnitt «***[global]***» bekanntgegeben (hier: CUPS) und geladen wird. Das Laden geschieht automatisch durch den Standardwert «***load printers = yes***».

```
# Drucksystem auswaehlen u. aktivieren:
printing = CUPS
```

Wurde der Druckername im Cups-Server mit «***epson-c84***» gewählt, heißt dann auch das Druckershare so und kann angesprochen werden mit:

```
\\192.168.0.1\epson-c84
```

Die Drucker lassen sich aber natürlich auch über die Netzwerkumgebung oder Systemsteuerung einbinden.

4.11 Transparentes Samba

Im Zusammenhang mit dem Mountmechanismus mittels der Dateien ***/etc/fstab*** und ***/etc/mtab*** lässt sich eine Samba-Freigabe sehr komfortabel im lokalen Dateisystem nutzen. Der Benutzer ***root*** kann das im einfachsten Fall manuell zu Testzwecken durchführen:

```
mount -t smbfs //192.168.0.1/pub /mnt
```

Die Angabe des Dateisystemtypes «***-t smbfs***« kann sogar noch entfallen, da moderne Kernel ihn automatisch erkennen.
Obwohl es sich um ein „public share“ handelt, wird ein Prompt zur Eingabe eines Passwortes gezeigt, den der ***root*** zwar mit «ENTER«(leeres Passwort) einfach nur quittiert, sich aber beim automatischen Mounten als störend erweist. Abschilfe schafft hier die Option ***guest***:

```
mount  //192.168.0.1/pub /mnt -o guest
```

Aber damit wird man in der Praxis immer noch nicht so glücklich sein. Besser der Admin schreibt es in die ***/etc/fstab*** hinein (= eine Zeile!):

```
//192.168.0.1/pub  /net/pub  smbfs
     noauto,umask=0000,guest 0 0
```

Mit der Angabe «***umask=0000***» wird den Benutzern volle Rechte an dieser Ressource gegeben.

Das nächste Beispiel zeigt einen Eintrag der Datei ***/etc/fstab***, mit dem sich der Benutzer ***max*** das private Share ***share1*** in sein Dateisystem einhängen kann:

```
//192.168.0.1/share1  /net/sh1  smbfs
        noauto,umask=0002,
              username=max,password=geheim 0 0
```

So kann ein Benutzer selbst sein Share mit «***mount /net/sh1***» mounten.
Nur gibt es dabei ein kleines Problem: Die Datei ***/etc/fstab*** muss systemweit lesbar sein. Und die Passwörter stehen dort im Klartext drin!
Die Lösung ist ein Verweis in Form einer Mountoption («***credentials=***») auf eine Datei, in der die Zugangsdaten stehen und die dann nur für ***root*** lesbar ist (Alles in einer Zeile!):

```
//192.168.0.1/share1  /net/sh1  smbfs
    noauto,umask=0002,
         credentials=/etc/samba/credentials 0 0
```

Die Datei ***/etc/samba/credentials*** sieht dann so aus:

```
username = max
password = geheim
```

Ein einfacher Benutzer kann also mit diesen Einträgen die Ressourcen einhängen. Wenn das automatisch beim Booten geschehen soll, ist die Option «***noauto***» wegzulassen. Das funktioniert aber nur, wenn der smbfs-Treiber entweder fest im Kernel oder in der initialen RAM-Disk eingebunden ist.
ACHTUNG: Der Mountpunkt muss dem jeweiligen Benutzer gehören und

er muss Schreibrecht daran haben, sonst schlägt auch das manuelle Mounten mit «***mount /net/pub***» fehl! Eine weitere Voraussetzung ist, dass das suid-Bit für die Programme ***smbmnt*** und ***smbumount*** gesetzt ist.

4.12 Primary-Domänen-Controller

Die Funktionalität eines PDC wird von Anwendern hoch geschätzt, weil sich damit einige Vorteile ergeben, so z.B.

- Single Sign On (Mit einem Login stehen alle Dienste zur Verfügung)
- Verfügbarkeit der persönlichen Arbeitsumgebung von unterschiedlichen Maschinen aus (die Maschinen müssen Mitglied der Domäne sein)
- Mehr Möglichkeiten über zentrale Policyverwaltung, bessere Kontrolle der Zugriffsrechte

In aller Kürze hier ein Konfigurationsbeispiel, das dies unter der Maßgabe, zusätzlich zwei öffentliche Shares (***pub*** mit Schreib/Leserechten und ***doku*** mit Nur-Leserechten) zur Verfügung zu stellen, realisiert:

```
# Beispiel-Config-Datei smb.conf:
[global]
        # Name der Domaene, darf nicht mit dem
        # Hostnamen des Servers identisch sein!
        workgroup = PDC-1

        # Damit ein Public-Share gleich sichtbar ist:
        map to guest = bad user
        guest account = nobody

        # Fuer PDC-Funktionalitaet:
        domain master = yes
        domain logons = yes
        preferred master = yes
        local master = yes
        os level = 64

        # Wirkt sich positiv aus, aber bitte nur einen
        # WINS-Server im Netzwerk laufen lassen:
         wins support = yes

        logon script = logon.bat
        logon path = \\%N\profiles\%u

        # Folgender Eintrag kann erst auskommentiert
```

```
        # werden, NACHDEM die Machinen zur Domaene
        # hinzugefuegt wurden:
        # invalid users = root

[profiles]
        path = /srv/winprofiles
        writeable = yes
        create mask = 600
        directory mask = 700
        browseable = no

[netlogon]
        path = /home/netlogon
        browseable = no
        read only = yes

[share1]
        comment = Freigabe 1
        path = /src/share1
        browseable = no

[share2]
        comment = Freigabe 2
        path = /src/share2
        browseable = no

        # Zugriff erlauben fuer:
        valid users = max susi +smbadmin

[pub]
        comment = Vorsicht, jeder darf alles!
        path = /srv/pub
        read only = no
        guest ok = yes

[doku]
        comment = Dokumentation
        path = /srv/doku
        read only = yes
        guest ok = yes

[printers]
        comment = All Printers
        path = /var/tmp
```

```
        printable = yes
        create mask = 0600
        browseable = no
```

Jetzt müssen die Maschinenaccounts (wieder doppelt, für Unix und SMB) angelegt werden: Dabei ist darauf zu achten, dass es sich beim letzten Argument um den NetBIOS-Namen der betreffenden Windows-Workstation (hier: pc01) handelt. Das abschließende Dollarzeichen wird nur beim Unix-Account benötigt. Zum Problem bei SuSE mit dem Teil «***−d /dev/null***» siehe obigen Abschnitt „Lokale, doppelte Buchführung".

```
useradd -g users -s /bin/false -d /dev/null pc01$
smbpasswd -a -m pc01
```

Zum Integrieren der Workstations muss nun auch der Benutzer ***root*** (evl. mit anderem Passwort) in die SMB-Benutzerdatenbank gebracht werden:

```
smbpasswd -a root
```

Das kann nun geschehen (im Beispiel mit MS-Windows 2000):
Rechter Mausklick auf «***Arbeitsplatz***» ⇒ «***Eigenschaften***»,
⇒ Registerkarte «***Computername***» ⇒ Schalter: «***Ändern***»,
⇒ bei «***Mitglied von:***» die Option «***Domäne***» aktivieren und in die Textbox den Domänennamen, hier «***PDC-1***» eintragen.
(Evl. unter «***Weitere***» das Häckchen bei «***Primäres DNS-Suffix...***» entfernen.)
Dieses Einbinden der Workstation in die Domäne darf nur ein authorisierter Benutzer vornehmen, hier ist es «***root***» mit dem eben vergebenen Samba-Kennwort.
Nach dem Rechnerneustart kann sich ein einfacher Benutzer mit Unix- und Samba-Konto, z.B. ***tux*** an der Domäne anmelden.
Aber erst beim ersten Abmeldevorgang wird das Verzeichnis ***tux*** mit seiner Windows-Profiles Struktur auf den Server unter «***/srv/winprofiles***» kopiert und dort für spätere Logins bereitgehalten.
Das Share «***[netlogon]***» ist dafür gedacht, beim Login eines Benutzers bestimmte Skripte auszuführen. In Beispiel ist ein solches Skript die ***logon.bat***, die im Pfad «***/home/netlogon***» liegen muss und zum Einbinden zweier Shares so aussehen könnte:

```
net use x: \\srv01\pub
net use y: \\srv01\share1
```

Damit das funktioniert, ist die Datei allerdings mit einem Windows/DOS-Editor zu erstellen (CR/LF-Problem) oder unter Linux mit «***vi***», dann aber am Zeilenende stets Escape-Sequenzen: «***STRG-v***», danach «***STRG-m***» anfügen! Eine weitere Möglichkeit ist, die Batchdatei mit einem beliebigen Editor unter Linux zu erstellen und sie danach mit dem Tool ***unix2dos*** zu konvertieren.

Da der Server «***serv01***» auch den WINS Dienst anbietet, sollte dieser natürlich auch in den Einstellungen zur Netzwerkkonfiguration auf der Workstation eingetragen werden.
Weiterführene Links:
http://www.bildungsservice.at/technik/netzwerk/samba-domain-mit-NT40.htm
http://www.linux-magazin.de/Artikel/ausgabe/2003/02/domaenen/dom.html
http://www.tutorials.de/tutorials7525.html

4.13 Fehlersuche

Zur Eingrenzung von Fehlern bieten sich einige Möglichkeiten an:

- Das Kommando ***smbstatus***: Selbstest mit Ausgabe der Browseliste: «***smbclient -L localhost -U%***» und Verbindungsversuch mit «***smbclient //hostname/sharename -U user***»

- Logfiles auswerten:
  Standardmäßig gibt es im Verzeichnis ***/var/log/samba*** zwei Dateien: ***log.nmbd*** und ***log.smbd***
  Mit folgenden Einträgen im globalen Bereich der ***smb.conf*** lassen sich Protokolldateien mit der maximalen Größe von 500 Bytes für die einzelnen Clients erzeugen:

  ```
  log file = /var/log/samba/log.%m
  max log size = 500
  ```

- Das Kommando ***smbstatus***: Es ermöglicht es, aktuell aktive Verbindungen und geöffnete Dateien zu untersuchen. Damit lassen sich Sperrprobleme verfolgen.

- Das Kommando ***nmblookup***: Mit «***nmblookup '*'***» kann man nach allen Clients im Netzwerk fragen, siehe dazu auch [SMBfail].
  In gemischten Netzen tritt das Problem auf, dass NetBIOS- und Hostnamen auseinandergehalten werden müssen. Mit der Zeile:

  ```
  nmblookup -A 192.168.0.3
  ```

  werden die Namen sowie Arbeitsgruppe/Domäne des betreffenden Rechners ausgegeben.

4.14 Ausblick

Samba ist nicht nur ein einfacher Windowserver-Ersatz, die Software kann jeden SMB/CIFS-aktiven Client unterstützen. Das bedeutet, auch zwischen Linuxplattformen kann sie gut verwendet werden.

Seit Version 2.2 werden ACL's unterstützt. Um dieses Feature zu benutzen, muss Samba meist aus den Quellen kompiliert werden.
Damit sich der Server in heterogenen Netzen immer besser einsetzen lässt, soll mit der 2005 kommenden Version 4.0 eine eigene LDAP-Implementation und bessere Zusammenarbeit mit Kerberos enthalten sein. Damit kann Samba als Server für Microsofts Directory Services (ADS) auftreten.

Literatur

[linServer] *Linux-Server für Intranet und Internet*,
Jörg Holzmann, Jürgen Plate, HANSER Verlag, 2002

[manSamba] *Deutsche Samba Dokumentation im Netz*
http://gertranssmb3.berlios.de/output/nmbd.8.html

[SMBlevel] *Deutsche Samba Dokumentation im Netz*
http://gertranssmb3.berlios.de/output/ServerType.html

[WillSam] *Willemers Informatik-Ecke, Samba*
http://www.willemer.de/informatik/unix/lisamba.htm

[LinMag] *Linux-Magazin, 03/2005*
http://www.linux-magazin.de

[NetBrowse] *Deutsche Samba Dokumentation im Netz*
http://gertranssmb3.berlios.de/output/NetworkBrowsing.html

[SMBfail] *Samba Fehlersuche*
http://www.64-bit.de/dokumentationen/netzwerk/b/001/
25741-14.htm

Weitere Links:

$\Longrightarrow$ http://samba.sernet.de
Offizielle Samba Webseiten

$\Longrightarrow$ http://gertranssmb3.berlios.de/Samba-HOWTO-Sammlung.pdf
Download der deutschen HOWTO Collection

$\Longrightarrow$ http://www.gentoo.de/doc/de/quick-samba-howto.xml
Grundlagen, Tipps

$\Longrightarrow$ http://www.linux-praxis.de/linux3/samba.html
Sehr grundlegende Arbeit, viele praktische Hinweise