

## **Linux als Fileserver für Windows nutzen**

---

### **1. Einführung in Samba**

Samba basiert auf dem SMB (Server Message Block) Protokoll, welches von Microsoft verwendet wird. Anhand von SMB ist es in einem Microsoft Windows Netz möglich Dateien und Drucker mit anderen Nutzern gemeinsam zu nutzen. SMB baut auf dem NetBIOS Protokoll auf, um die Namensverwaltung innerhalb eines Netzes zu verwalten. NetBIOS bildet nicht wie TCP/IP Netzwerknamen auf die IP-Adresse ab, sondern anhand der Netzwerkkartenadresse (MAC-Adresse) wird dem Rechner ein Name zugewiesen. Damit man Samba einsetzen kann, müssen die verwendeten Betriebssysteme das SMB Protokoll beherrschen, was heute unter nahezu jedem Betriebssystem der Fall sein sollte.

Ein Samba Server besteht prinzipiell aus 2 Komponenten, dem `smbd` und dem `nmbd` Daemon.

#### **1.1 Der `smbd` Daemon**

`smbd` ist der Server Daemon welcher Dateien und Druckdienste für Windows Clients bereitstellt die das SMB Protokoll benutzen. Dieses Protokoll ist kompatibel mit dem LanManager Protokoll und kann auch LanManager Clients bedienen. Das beinhaltet MSCLIENT 3.0 für DOS, Windows for Workgroups, Windows 95/98/ME, Windows NT, Windows 2000, OS/2, DAVE for Macintosh, und `smbfs` für Linux.

#### **Der `nmbd` Daemon**

`nmbd` ist ein Server der NetBIOS Anfragen versteht und auch beantworten kann, welche z. B. von SMB Clients wie Windows 95/98/ME, Windows NT, Windows 2000, und LanManager Clients. `nmbd` ist auch beteiligt an den "Browsing Protokollen" (LAN browsing) welche die Computer des Netzwerkes in der Netzwerkumgebung sichtbar machen.

Zum Zeitpunkt des bootens wollen SMB/CIFS Clients mit einem SMB Server Kontakt aufnehmen, um herauszufinden welche Rechner welche IP-Adresse nutzt. Inmitten anderer Dienste, horcht `nmbd` nach solchen Anfragen und wenn der eigene NetBIOS Name in der Konfigurationsdatei angegeben ist, antwortet er mit der IP-Adresse. Sein "eigener NetBIOS Name" ist standardmäßig der primäre DNS Name der Maschine auf der `nmbd` läuft. Damit antwortet `nmbd` mit seinem eigenen Namen auf die Broadcast Adresse des Netzwerks.

`nmbd` kann auch als WINS (Windows Internet Name Server) Server genutzt werden. Das bedeutet, `nmbd` arbeitet als WINS Datenbank Server. Diese Datenbank befüllt er mit Registrierungsanfragen die er erhält und antwortet auf Fragen nach Namen von Clients mit diesen.

### **2. Was kann Samba alles - und was nicht**

Die wohl wichtigste Eigenschaft von Samba besteht darin, dass erst durch Samba Linux in der Netzwerkumgebung von Windows sichtbar wird und somit auf die dort verfügbaren Ressourcen zugegriffen werden kann.

Samba kann also für Windows Rechner im Netz als File und Printserver dienen. Seit der Samba Version 2.2.0 kann Samba auch als PDC (Primary Domain Controller) für Windows NT und Windows 2000 konfiguriert werden, Samba kann also in einem Windows Netzwerk auch als Anmeldeserver fungieren.

Samba kann in der aktuellen Version 3.0 in eine ADS Domäne integriert werden. Einen ADS Server kann es aber nicht ersetzen!!

### **3. Konfiguration des Servers**

#### 3.1 Die global Sektion

Die Konfiguration von Samba gestaltet sich recht einfach, hat man erst einmal alle möglichen Konfigurationsparameter der zentralen Samba Konfigurationsdatei `smb.conf` auswendig gelernt ;-).

Die Konfigurationsdatei von Samba liegt bei unserem Linux unter `/etc/samba/smb.conf`.

Die Konfigurationsdatei ist im Wesentlichen in zwei Abschnitte gegliedert: eine GlobalSection und in die jeweils angebotenen Dienste oder Verzeichnisse. Die jeweiligen Parameter sind größtenteils selbsterklärend.

#### Zugriffsrechte und Sicherheitsaspekte

Samba kann unterschiedlich konfiguriert werden was den Zugriff angeht. Es gibt drei verschiedene Arten von Sicherheitsstufen:

- Share-Level-Sicherheit
- User-Level-Sicherheit
- Domain-Level-Sicherheit

Die *Share-Level-Sicherheit* ist die einfachste Form der Zugriffssteuerung auf Samba. Jedes Verzeichnis oder Drucker bekommt ein eigenes Passwort. Die einzelnen Freigaben können auch ganz freigegeben werden, so dass kein Passwort mehr benötigt wird. Die Folge, es gibt keinerlei Zugriffsbeschränkung.

Der Nachteil dieser Strategie liegt in der Anzahl der Passwörter oder aber in der praktisch nicht vorhandenen Zugriffskontrolle. Angenommen wir haben auf unserem Sambaserver 10 Verzeichnisse freigegeben, so hat jedes Share ein eigenes Passwort. Unsere User werden Ihre Monitore mit gelben Postit Klebezetteln zu kleistern, da sie sich diese vielen Passwörter aller Wahrscheinlichkeit nicht merken werden können, oder wollen.

Die *User-Level-Sicherheit* setzt voraus, dass sich die User, welche auf den Sambaserver zugreifen wollen, an dem Linux Rechner mit Benutzername und Passwort authentifiziert haben. Greift der User jetzt auf den Sambaserver zu, gelten diese Daten als Zugangsberechtigung. Weiterhin müssen sich Client und Server in derselben Arbeitsgruppe befinden.

Einem Verzeichnis auf dem Sambaserver ist also ein Benutzer mit dazugehörigem Passwort

zugeordnet. Es können auch User in einer Gruppe zusammengefasst werden, um beispielsweise einer ganzen Abteilung Zugriff auf ein Verzeichnis zu geben. Natürlich müssen die Anmeldedaten auf dem Samba-Server in einer Datenbank verwaltet werden. Der Nachteil dieser Lösung besteht darin, dass wenn der User sein Passwort ändert, dieses auch auf dem Samba-Server geändert werden muss.

Bei der *Domain-Level-Sicherheit* gibt es einen zentralen Anmeldeserver, welcher mit Windows NT 4.0 in Form des Domänenkonzeptes eingeführt wurde. Der Vorteil liegt darin, dass der User seinen Benutzernamen mit Passwort an eine zentrale Instanz übermittelt, die bei korrekter Eingabe, die Anmeldung erlaubt. Genauso sieht es bei Zugriff auf den Samba-Server aus, der bei einem Zugriff die Userdaten bei dem zentralen Anmeldeserver verifiziert und den Zugriff erlaubt. Es ist keine doppelte Benutzerverwaltung mehr nötig.

Doch nun zurück zur Konfiguration von Samba. Oben sagten wir dass sich die Konfigurationsdatei aus zwei Abschnitten zusammensetzt, die GlobalSection und die Einträge für die einzelnen Shares. Eine GlobalSection könnte z.B. so aussehen:

```
[global]
workgroup = fh-trier
server string = %h, running not windows %v
security = user
encrypt passwords = Yes
netbios name = samba-srv
guest account = nobody
```

Was sagt uns nun dies alles? Alles gar nicht so schlimm. Wie oben beschrieben bei den Sicherheitsaspekten, verwendet diese Konfiguration die User-Level-Sicherheit, zu erkennen an dem Parameter `security`. Da bei der User-Level-Sicherheit alle PC's in derselben Arbeitsgruppe sein müssen um auf den Samba-Server zugreifen zu können, geben wir eine Arbeitsgruppe mit dem Parameter `workgroup` an. `server string` dient eigentlich nur der Kosmetik. Damit erzeugt Samba in der Windows Netzwerkumgebung einem Eintrag im Feld 'Kommentar', `encrypt passwords` wird unbedingt benötigt, wenn Windows Clients ab Windows 95 SR2 auf die Samba Freigaben zugreifen können sollen, da diese Betriebssysteme ihre Passwörter verschlüsselt übertragen. Anhand von `netbios name` können Windows Clients den Samba Server per WINS auflösen.

Man kann trotz User-Level-Sicherheit Verzeichnisse global freigeben. Dies geschieht mit dem Eintrag `guest account` in der global Sektion. Soll ein Verzeichnis global für alle Freigegeben werden muss allerdings in der Konfiguration der Shares der Eintrag `guest ok = yes` eingetragen sein.

Für einen einfachen Samba Server reicht diese global Sektion vollkommen aus. Nun geht es daran Verzeichnisse frei zu geben.

### 3.2 Die Share Sektion

Eine einfache Freigabe könnte so aussehen:

```
[Docs]
comment = Diverse Dokumentationen
path = /daten/dokumentation
valid users = @samba
```

```
read only = No
```

Den Inhalt der eckigen Klammer zeigt sich im Windows Explorer als Bezeichnung des Shares. Das oben freigegebene Verzeichnis erscheint also unter dem Namen Docs. Der Parameter `comment = Wert` befüllt wieder das Kommentarfeld im Windows Explorer. `path = absoluter Pfad` gibt das Verzeichnis an, welches freigegeben werden soll. Oben habe ich ebenfalls davon gesprochen, Benutzer in einer Gruppe zusammen zu fassen, dies geschieht mit `valid users = Wert`. Wichtig ist das @!!. Als letztes gibt `read only = No` an, dass die User, welche auf das Share Zugriff haben (@samba), auch schreibenden Zugriff haben.

Auf vielen Samba Servern werden die Homeverzeichnisse der User vorgehalten. Dieses Share nimmt aber in gewisser Weise eine Sonderstellung ein, welche in den Berechtigungen begründet ist. So ist es ja nicht erwünscht, dass alle User die entsprechenden Homeverzeichnisse der anderen sehen sollen, bzw. sogar einen Blick in dieses Verzeichnis werfen. Die Sharedefinition für die Homeverzeichnisse der User sieht so aus:

```
[homes]
comment = Heimatverzeichnis
read only = No
browseable = No
```

Die ersten drei Zeilen kennen wir ja bereits, sie füllen das Kommentarfeld und erlauben den schreibenden Zugriff auf das entsprechende Homeverzeichnis. Mit dem Parameter `browseable = No` verhindern wir, dass das Share in der Windows Netzwerkumgebung angezeigt wird. Für normale Shares muss dieser Parameter nicht auf = Yes gesetzt werden, dass ist per Standarddefinition so eingerichtet.

Die bisher vorgestellten Parameter für die Sharedefinitionen zeigen natürlich nur einen kleinen Teil, um schnell zu einem Ergebnis zu kommen. Hier möchte ich ein paar weitere nützliche Parameter für die Sharedefinitionen zeigen. Ein weiteres Share könnte z.B. so aussehen:

```
[MP3]
comment = Meine MP3 Sammlung
path = /daten/mp3
valid users = tux
directory mask = 0770
create mask = 0770
read only = No
```

In dieser Sharedefinition sind zwei weitere Parameter verbaut worden, `directory mask = Wert` und `create mask = Wert`. Damit lassen sich die Verzeichnis- (directory mask) bzw. die Dateirechte bei der Erstellung beeinflussen. In obigem Beispiel werden alle neu erstellten Verzeichnisse und Dateien im Share MP3 mit Schreib- Lese- und Ausführungsrechten ausgestattet. Der Parameter `valid users = tux` in dieser Sharedefinition zeigt auch, wie einfach es sein kann, User aus bestimmten Verzeichnissen auszusperrern. So darf nur tux, auf die unter MP3 abgelegte Musik zugreifen.

Haben wir alle entsprechenden Shares angelegt, bietet uns Samba das Tool `testparm`, mit welchem Sie die Konfigurationsdatei `smb.conf` auf Syntaxfehler auf Fehler überprüfen kann. Die Ausgabe von `testparm` könnte z.B. so aussehen:

```

root@[tuxhome:~]# testparm
Load smb config files from /etc/samba/smb.conf
Processing section "[tmp]"
Processing section "[Docs]"
Processing section "[Daten]"
Processing section "[MP3]"
Processing section "[Programme]"
Processing section "[Filme]"
Processing section "[Voll]"
Processing section "[homes]"
Loaded services file OK.
root@[tuxhome:~]#

```

`testparm` listet uns als erstes alle gefundenen Sharedefinitionen auf und, so denn die Syntax stimmt, als letztes ein OK. Danach werden wir aufgefordert mit Enter zu bestätigen. Danach zeigt uns `testparm` die Syntax der konfigurierten Shares auf, so wie sie in der Konfigurationsdatei auch zu sehen ist.

#### **4. SMB Userverwaltung**

Ein großes Kapitel bei der erfolgreichen Einrichtung von Samba ist die Userkonfiguration. Damit Windowsnutzer auf die von Ihnen bereitgestellten Verzeichnisse, und sei es nur das entsprechende Homeverzeichnis des Users, zugreifen können müssen zwei Grundlegende Voraussetzungen gegeben sein:

- Alle Windowsnutzer die auf die bereitgestellten Verzeichnisse zugreifen können sollen müssen als Systemuser existieren, und
- es müssen diese User ebenfalls als "Sambauser" angelegt sein.

Für den User `tux` muss ein Systemaccount im Betriebssystem bestehen. Die entsprechenden User dürften, zumindest für kleine Installationen, bereits auf dem System bestehen. Das eigentlich wichtigere ist, diesen User Samba bekannt zu machen. Für diese Aufgabe gibt es den Befehl `smbpasswd`, der uns bei dieser Aufgabe behilflich ist. Die Syntax des Befehls ist recht simpel:

```

root@[tuxhome:~]# smbpasswd -a username
New SMB password:
Retype new SMB password:
Password changed for user username.
root@[tuxhome:~]#

```

Ab dem jetzigen Zeitpunkt kann der Windowsnutzer `username` auf die Shares zugreifen, welche Samba bereitstellt. Wie gesagt dieser Schritt ist für alle Windowsnutzer nötig, die auf die freigegebenen Verzeichnisse Zugriff erhalten sollen, ansonsten kommt es, Windowstypisch, zu den seltsamsten Fehlermeldungen. Samba merkt sich die angelegten Benutzer in der Datei `smbpasswd`. Darin speichert Samba, ähnlich wie die System `/etc/passwd`, das Passwort in verschlüsselter Form. Man kann mit `smbpasswd` auch Benutzer sperren/deaktivieren. Hierfür ruft man `smbpasswd` mit der Option `-d username` auf. Ab diesem Zeitpunkt scheitern alle Versuche des Benutzers sich zu authentifizieren. Das ganze kann mit `smbpasswd -e username` wieder rückgängig gemacht werden, der Benutzer wird also wieder aktiviert.

## 5. Verzeichnisberechtigungen

Ein ebenfalls nicht zu vernachlässigendes Thema sind die Rechte der freigegebenen Verzeichnisse auf dem Samba Server. Hier gilt es sich eine geeignete Strategie zu überlegen, wer worauf welchen Rechte hat. Also oberste Faustregel gilt es zu beachten, dass die freigegebenen Verzeichnisse die entsprechenden Rechte besitzen, so das die Benutzer auch in diesem Verzeichnis machen können was sie dürfen. Betrachten wir nochmals das obige Beispiel:

```
[Docs]
comment = Diverse Dokumentationen und Online Bücher
path = /daten/dokumentation
valid users = @samba
read only = No
```

Als erstes muss festgelegt werden, welche Rechte die Benutzer in diesem Verzeichnis erhalten sollen. Sollen die Benutzer nur den Inhalt des Verzeichnisses einsehen und die darin enthaltene Dokumentation auch lesen können, so muss das Verzeichnis /daten/dokumentation mindestens folgende Rechte haben:

```
drwxr-x---  9 root  root  248 Jan 18  2004 /daten/dokumentation/
```

Mit diesen Rechten ausgestattet können die Benutzer die im Verzeichnis enthaltenen Dateien lesen aber keine neuen Dateien oder Verzeichnisse erstellen, und das obwohl in der Sharedefinition der Parameter `read only = No` gesetzt ist. Da sich die Benutzer aber nicht in der Gruppe `root` befinden wird ihnen der Schreibzugriff verwehrt. Wollen Sie aber ihren Nutzern auch schreibenden Zugriff auf das Verzeichnis geben, müssen die Verzeichnisberechtigungen so aussehen:

```
drwxrwx---  9 root  samba  248 Jan 18  2004 /daten/dokumentation/
```

Damit haben die Benutzer, welche sich in der Gruppe `samba` befinden, Schreib- und Leserechte.

In meinen Augen macht es Sinn, für Samba eigene Usergruppen zu erstellen, da sich so die Zugriffsrechte feiner gestalten lassen, da man beim Parameter `valid users =` dadurch mit den Gruppen arbeiten kann, anstatt alle Benutzer im Parameter `valid users =` aufzuführen. So sieht das doch wesentlich übersichtlicher aus

```
valid users = @verwaltung @vorstand @buchhaltung
```

als so

```
valid users = martin claudia vorstand1 vorstand2 anja martina christian
```

Einen weiteren Vorteil hat das ganze nebenbei auch noch, so sehen wir auf Anhieb, welche Gruppen auf welche Verzeichnisse Zugriff haben, oder eben nicht.

Einen letzten Tipp möchte ich noch in Bezug auf die Datei- und Verzeichnisberechtigungen geben. Oft ist es gewünscht, dass Dateien und Verzeichnisse bei Erstellung bestimmte Rechte

erhalten, bzw. einer bestimmten Gruppe oder einem bestimmten Benutzer zugeordnet werden. Für diesen Fall gibt es vier Parameter für die Konfigurationsdatei *smb.conf*:

```
directory mask = 0770
create mask = 0770
force group = bspgrp
force user = bspuser
```

Mit diesen Parametern, welche in der Sharedefinition zum Einsatz kommen, lassen sich die beschriebenen Aufgaben meistern.

## **6. Variablen**

Samba bietet für die Konfiguration eine ganze Reihe von nützlichen Variablen, welche die Konfiguration von Samba komfortabler gestalten. Mit diesen Variablen lassen sich z.B. Logdateien erstellen, welche den Hostnamen der Clientmaschine enthält, oder im einfachsten Fall die Version von Samba ausspuckt.

```
server string = %h, running Samba %v
```

Mit dieser Zeile haben wir das Kommentarfeld in der Netzwerkkonfiguration mit Inhalt befüllt. Dort sehen Sie aber nicht *%h, running Samba %v*, sondern beispielsweise *tuxhome, running Samba 2.2.5*. Dies soll nur ein kleines Beispiel für den Einsatzzweck für die Variablen von Samba sein, die Verwendung bei Ihnen hängt von Ihren Bedürfnissen ab. Eine Liste der verfügbaren Variablen und deren Bedeutung möchte ich Ihnen dann doch mitgeben:

- %S Name des aktuellen Dienstes/Verzeichnis
- %P Das Rootverzeichnis des aktuellen Dienstes/Verzeichnis
- %u Benutzername
- %g Primäre Gruppe des Benutzers (%u)
- %H Das Homeverzeichnis des in %u angegebenen Benutzers
- %v Die Samba Version
- %h Internet Hostname (DNS) des Samba Servers
- %m Der NetBIOS Name einer Clientmaschine
- %L Der NetBIOS Name des Samba Servers, ermöglicht mehrere Namen
- %M Internet Hostname (DNS) eines Clients
- %d PID des aktuellen Samba Servers
- %I IP-Adresse der Clientmaschine
- %T Aktuelle Zeit und Datum

Es gibt noch ein paar weitere Variablen, → [manpage zu smb.conf](#)

## 7. Samba als PDC in einem NT/w2k/XP Netz

### 7.2 Die smb.conf

Im wesentlichen unterscheidet sich die smb.conf eines normalen Samba Servers und eines Samba PDC's nur in 4 Dingen. Er muß Domain Master sein (`domain master = yes`), Domänen Anmeldungen zulassen (`domain logon = yes`), der bevorzugte Master sein (`preferred master = yes`) und ebenfalls der lokale Master sein (`local master = yes`). Alles andere in der nun folgenden smb.conf ist Optional, bietet sich jedoch an, da es die Funktionsfähigkeit um einiges erweitert.

Hier nun die smb.conf, so wie ich sie verwende (nur die global section):

```
[global]
; Basic server settings
netbios name = HOSTNAME
workgroup = DOMAINNAME

; we should act as the domain and local master browser
os level = 64
preferred master = yes
domain master = yes
local master = yes

; security settings (must user security = user)
security = user
admin users = root
domain admin group = root

; encrypted passwords are a requirement for a PDC
encrypt passwords = yes

; support domain logons
domain logons = yes

; where to store user profiles?
logon path = \%Nprofiles%u

; where is a user's home directory and where should it
; be mounted at?
logon drive = H:
logon home = \%L%u
passwd chat = New*password* %nn New*password*(again)* % nn *changed*
; Automatic generation of machine accounts
add user script = /usr/sbin/useradd -d /dev/null -g 500 -s /bin/false -
M %u

; specify a generic logon script for all users
; this is a relative **DOS** path to the [netlogon] share
logon script = %U.cmd

; necessary share for domain controller
[netlogon]
comment = netlogon
path = /daten/netlogon
valid users = @samba
read only = yes
```

```

write list = christian administrator

; share for storing user profiles
[profiles]
    comment = profiles
    path = /daten/profiles/
    valid users = @samba
    read only = no
    create mask = 0600
    directory mask = 0700

```

Damit Windows Maschinen sich in die Domäne aufnehmen lassen können muss ein sogenannter Maschinen Account auf dem Samba Server angelegt werden. Dies kann auf zwei Arten geschehen. Entweder man erstellt manuell diesen Maschinen Account mit den Befehlen `useradd -m rechner` oder man lässt dies automatisch geschehen. Damit das automatisch funktionieren kann, muss der Parameter `add user script = Wert` in der global Section gesetzt sein. In der obigen `smb.conf` ist dies der Fall, wir brauchen uns um die Maschinen Accounts also nicht weiter kümmern. Da die Maschinen Accounts automatisch erzeugt werden sollen ist es nötig das ein Nutzer mit Rootrechten Samba bekannt ist. Man kann beispielsweise `root Samba` bekannt machen, kann aber ein anderes Passwort verwenden.

### 7.3 Windows 2000/NT 4 Rechner in die Domäne aufnehmen

Nun kommt der entscheidende Schritt zu einer funktionierenden Domäne, die Windowsrechner an der Domäne anmelden.

Um eine Windows NT Maschine in diese aufzunehmen sind folgende Schritte nötig:

- Eigenschaften der Netzwerkumgebung,
- auf der Reiterkarte Identifikation auf Ändern,
- in der Zeile Domäne den Domänen Namen eintragen,
- vor Computerkonto in der Domäne erstellen den Haken setzen,
- als Benutzernamen `root` eingeben mit dem entsprechenden Passwort darunter,
- auf OK klicken.

Nach kurzer Zeit sollte eine Willkommensmeldung erscheinen, welche den Erfolg bescheinigt.




Um Windows 2000 Rechner in die Domäne aufzunehmen ist eine etwas andere Vorgehensweise nötig, da Microsoft mal wieder den jeweiligen Konfigurationsdialog komplett neu im System verteilt hat:

- Eigenschaften von Arbeitsplatz,
- auf der Reiterkarte Netzwerkidentifikation auf Eigenschaften,
- in der Zeile Domäne den Domänen Namen eintragen,
- in der darauf folgenden Dialogbox wieder `root` samt Passwort eingeben.

Nun sollte ebenfalls nach kurzer Zeit die Willkommensmeldung erscheinen.

Beispiel → Arbeitsplatz:

#### **Nur diesem Computer gespeicherte Dateien**

-  Gemeinsame Dokumente
-  Dateien von Albert Zweistein
-  Dateien von Christian Wilkin

Dateiordner  
Dateiordner  
Dateiordner

#### **Festplatten**





-  System (C:)
-  Daten (D:)
-  SYSDAT (E:)
-  leer (F:)

Lokaler Datenträger  
Lokaler Datenträger  
Lokaler Datenträger  
Lokaler Datenträger

38,3 GB  
57,2 GB  
27,9 GB  
37,2 GB

28,6 GB  
5,41 GB  
6,69 GB  
27,2 GB







#### **Geräte mit Wechselmedien**

-  3 1/2-Diskette (A:)
-  Wechseldatenträger (H:)
-  CD-Laufwerk (N:)
-  DVD/CD-RW-Laufwerk (O:)

3 1/2-Diskette  
Wechseldatenträger  
CD-Laufwerk  
CD-Laufwerk

692 MB  
0 Byte

#### **Netzlaufwerke**

-  download auf "Samba Server (tuxhome)" (R:)
-  sys auf "Samba Server (tuxhome)" (S:)
-  tmp auf "Samba Server (tuxhome)" (T:)
-  daten auf "Samba Server (tuxhome)" (U:)
-  prog auf "Samba Server (tuxhome)" (V:)
-  mp3 auf "Samba Server (tuxhome)" (W:)

Netzlaufwerk  
Netzlaufwerk  
Netzlaufwerk  
Netzlaufwerk  
Netzlaufwerk  
Netzlaufwerk

75,4 GB  
9,84 GB  
9,84 GB  
18,3 GB  
75,4 GB  
146 GB

16,9 GB  
2,21 GB  
2,21 GB  
9,98 GB  
16,9 GB  
76,1 GB