

# LPI 201

## Thema 209: Datei- und Dienstfreigabe

Autor: Kai Riedmann

# Samba - Server

## Konfiguration eines Samba-Servers

### Ziele:

- Samba-Server für verschiedene Clients aufsetzen
- Erstellen von Login-Scripts
- Aufsetzen eines nmbd-WINS-Servers
- Wechsel der Arbeitsgruppe
- Definition von Freigaben (Verzeichnisse / Drucker)
- Mounten von SMB-Freigaben unter Linux

### Schlüsseldateien, Begriffe und Hilfsmittel:

- smbd, nmbd
- smbstatus, smbtestparm, smbpasswd, nmblookup
- smb.conf, lmhosts

## Funktionen eines SMB-Servers:

- einfache Freigaben von Verzeichnissen unter Windows
- PDC / BDC (Domain-Server mit Anmeldefunktion usw.)
- WINS-Server (Namensauflösung unter Windows)

## Steuerung des SMB-Servers:

- smbd → Datei- Druckerfreigabe
- nmbd → Verwaltung der NetBIOS-Namen und WINS-Funktionen

## zentrale Konfigurationsdatei:

**smb.conf**

## smb.conf:

### 1. Konfiguration des Servers:

[GLOBAL]

workgroup = Arbeitsgruppe und Festlegung Domain (NT-Sys.)

security = Zugriffssicherheit

user /

encrypt passwords = Passwort-Verschlüsselung

printing = Druckeransteuerung

Namensauflösung

WINS-Dienste

Domain-Einträge

## WINS-Server:

Namensauflösung durch:

- lmhosts (lm = LanManager)
- /etc/hosts (wenn NetBIOS-Namen und DNS gleich sind)
- WINS (Namenserver für NetBIOS-Namen)

smb.conf: [GLOBAL]

|                    |                            |
|--------------------|----------------------------|
| name resolve order | = lmhosts hosts wins bcast |
| wins-server        | = IP                       |
| max ttl            | = Sekunden                 |

oder:

|              |            |
|--------------|------------|
| wins support | = yes      |
| max wins ttl | = Sekunden |
| min wins ttl | = Sekunden |

## lmhosts:

im Samba-Konfigurationsverzeichnis (/etc/samba bei Debian)

Aufbau:

*IP-AdresseNetBIOS-Name#NetBIOS-Typ*

NetBIOS-Typ:

| <i>NetBIOS Ressource</i>  | <i>Hex-Wert</i> |
|---------------------------|-----------------|
| Standard-Workstation      | 00              |
| Win-PopUp Dienst          | 03              |
| RAS-Server                | 06              |
| Domain Master-Browser/PDC | 1B              |
| Master-Browser            | 1D              |
| NetDDE-Dienst             | 1F              |
| Datei- oder Druckerdienst | 20              |
| RAS-Client                | 21              |
| Network Monitor Agent     | BE              |
| Network Monitor Utility   | BF              |

## Verzeichnis-Freigabe:

[*Freigabe-Name*]

comment = Kommentar

path = Pfad zur Freigabe

guest ok = Benutzung auch mit Gast-Accounts?

oder: public = Yes / No

read only = Nur-Lese-Zugriff

Homeverzeichnis:

[homes]

comment = ~username

read only = Nur-Lese-Zugriff (normalerweise: No)

browseable = Sichtbarkeit der Freigabe

## Drucker-Freigabe:

### [*Druckername*]

|              |                                        |
|--------------|----------------------------------------|
| comment      | = Drucker                              |
| path         | = Pfad zu einem temporären Verzeichnis |
| guest ok     | = Benutzung auch mit Gast-Accounts?    |
| oder: public | = Yes / No                             |
| printable    | = Freigabe ist ein Drucker             |
| printer name | = Druckername im Drucksystem           |

### [*printers*]

|            |                                  |
|------------|----------------------------------|
| comment    | = Alle Drucker anzeigen          |
| path       | = Pfad zu temporärem Verzeichnis |
| printable  | = Freigabe ist ein Drucker       |
| browseable | = Sichtbarkeit der Freigabe      |
| guest ok   | = Gast-Accounts erlaubt?         |

## Hilfsprogramme I:

testparm:

- Prüfung der smb.conf auf Syntax-Fehler
- trotzdem kann ein Service nicht lauffähig sein!!!

smbstatus:

- Darstellung des aktuellen Samba-Status

smbpasswd:

- Anlegen eines SMB-Users (Account unter Unix muss vorhanden sein): *-a*
- Ändern eines Passworts
- Löschen eines SMB-Users: *-x*
- deaktivieren eines SMB-Users: *-d*
- Aktivieren eines deaktivierten Eintrags: *-e*

## Hilfsprogramme II:

nmblookup:

- NetBIOS-Client
- bedient sich smb.conf (*wins server / name resolve order*)

Aufruf:

*nslookup NetBIOS-Name*

*nslookup 'NetBIOS-Name#Typnummer'* (nur nach Typ suchen)

*nslookup -U WINS-Server -R 'NetBIOS-Name#Typnummer'*

smbmount:

Mounten von Windows-Freigaben unter Unix / Linux

*smbmount //Servername/Freigabename Mountverzeichnis -o*

*Optionen*

*Optionen:*

*NAME=WERT, Komma-getrennt*

### Logon-Scripts:

[global]

logon scripts = *Scriptname*

Das Skript muss ein DOS-Skript mit CR/LF sein, sonst funktioniert es nicht

Beispiel:

logon scripts = %U.bat

[netlogon]

Unter dieser Freigabe wird das logon-Skript gesucht

## Logon-Scripts:

Beispiel-Skript:

```
NET USE F: \\samba\cdrom1  
NET USE G: \\samba\home  
NET USE H: \\samba\public
```

# NFS - Server

## Konfiguration eines NFS-Servers

Ziele:

- Erstellen einer export-Datei
- Zugriffsbeschränkung der exportierten Dateisysteme
- Konfiguration des tcpwrappers

Schlüsseldateien, Begriffe und Hilfsmittel:

- /etc/exports
- exportfs
- showmount
- nfsstat

## /etc/exports:

Aufbau:

Freizugebendes\_Verzeichnis Client(Optionen) Client2(Optionen)...

Client: Rechnername, IP-Adresse, IP-Netz, NIS-Gruppen  
(@Gruppe)

Beispiel: bla.foo.bar, 192.168.0.1, 192.168.0.0/24

Optionen:

- Schreib-/Leseerlaubnis:      rw/ro
- UserID-Mapping:
  - no\_root\_squash (alle, auch Root-Rechte werden gemappt)
  - root\_squash (Root wird auf anonyme UserID/GruppenID abgebildet)
  - all\_squash (alle User-IDs auf anonym abgebildet)
  - anonuid/anongid (setzen der anonymen Rechte auf einen festen Benutzer)

## /etc/exports:

### Optionen:

- secure: Nur Ports unter 1024 zur Verbindung erlaubt
- sync: alle Schreibbefehle werden sofort ausgeführt
- no\_wdelay: nur mit *sync*, Schreibzugriffe werden ohne Wartezeit direkt ausgeführt
- nohide: Wechseln zweier Dateisysteme in einer Freigabe möglich, weiteres mounten wird nicht verlangt
- no\_subtree\_check: ausschalten des Subtree-Checking
- insecure\_locks/no\_auth\_nlm: keine Authentifizierung bei Dateisperranfragen

## /etc/exports:

Beispiele:

|           |                                        |
|-----------|----------------------------------------|
| /         | einstein.foo.bar(rw,no_root_squash)    |
| /ftp/pub  | (all_squash, anonuid=123, anongid=555) |
| /projects | master(rw), notrust(ro, root_squash)   |
| /usr      | *.local.domain(rw) @nis-domain(rw)     |

### Mounten von NFS-Freigaben:

`mount nfs-server:freigabe mount-point`

Mount-Optionen:

- `rsize=1024` Lesecache
- `wsize=1024` Schreibcache
- `timeo=7` Wert in 1/10-Sek. bis eine Wiederholung nach einem RPC-Timeout gesendet wird
- `retrans=3` Anzahl der Minor-Timeouts bevor ein Major-Timeout gesendet wird
- `acregmin=3` Zeit, in der Attribute zwischengespeichert werden vor nächster Anforderung der Attribute vom Server
- `acregmax=60`
- `acdirmin=30` wie `acregmin`, nur mit Verzeichnissen
- `acdirmax=60`

## Mounten von NFS-Freigaben:

### Mount-Optionen:

- actimeo=n Gleichsetzen aller ac\*-Parameter auf Wert n
- retry=10000 Anzahl der Versuche für NFS-Anmeldung
- namlen=255 maximale Länge eines Dateinamens (RPC2)
- port=0 Portnummer für Verbindung zum NFS-Server
- mountport=n Portnummer des mountd
- mounthost=name  
Name des mountd-Hosts
- mountprog=100005  
alternative RPC-Programmnummer für Kontakt  
zum Mount-Dämon (bei mehreren NFS-Servern)
- mountvers=1 setzt alternative RPC-Versionsnummer
- nfsprog=100003  
alternative RPC-Programmnummer für Kontakt  
zum NFS-Dämon (bei mehreren NFS-Servern)
- nfsvers=2 setzt alternative RPC-Versionsnummer

## Mounten von NFS-Freigaben:

### Mount-Optionen:

- nolock NFL Locking ausgeschaltet
- bg Durchführen des Mounts im Hintergrund, wenn der erste Mount-Versuch fehlschlägt
- fg wie *bg*, nur im Vordergrund
- soft Senden eines I/O-Fehlers, wenn ein Major-TO eintritt
- hard sendet an das Programm einen “*server not responding*”
- intr wenn Major-TO auftritt können Signale zur Unterbrechung gesendet werden
- posix mounten mit POSIX-Spezifikationen
- nocto Unterdrückt holen Attribute für neue Datei
- noac kein Caching von Attributen
- nolock kein Locking verwenden
- tcp/udp Verwendung des jeweiligen Protokolls

## Hilfsprogramme:

showmount: - Abfrage von Serverinformationen (wie  
smbstatus,smbclient)

Optionen:

- a: - Zeigt alle Informationen an
- d: - zeigt nur Verzeichnisse, die von Clients gemountet  
sind
- e: - Exportliste des Servers anzeigen
- no-headers:
  - Unterdrückt Überschriften in der Ausgabe

nfsstat: - statische Informationen über den lokalen NFS-  
Server

### TCP-Wrapper und NFS:

- TCP-Wrapper ist zuständig für die Erlaubnis und die Sperrung einzelner Rechner zum Zugriff auf bestimmte Ports/Programme
- Einträge werden in /etc/hosts.allow vorgenommen (bei alten Versionen in /etc/hosts.allow und /etc/hosts.deny)
- Beispiel-Eintrag:  
    mountd : .foo.com : ALLOW  
    mountd : ALL : DENY

# LPI 201

## Thema 211: Systemwartung

Autor: Kai Riedmann

# System-Logbuch

## System-Logbuch

### Ziele:

- Konfiguration des syslogd als zentralen Logserver
- Konfiguration von syslogd
- Aufnehmen entfernter Verbindungen ins Logbuch
- Verwendung von grep und anderen Utils zur Analyse

### Schlüsseldateien, Begriffe und Hilfsmittel:

- syslog.conf
- syslogd
- /etc/hosts

## Aufbau eines zentralen Logservers

Starten des syslogd als Daemon, der remote-Logeinträge aufnimmt:

```
/sbin/syslogd -r
```

## Einstellung auf Clientseite:

```
/etc/syslog.conf
```

```
*.* @Logserver
```

Schon beim Bootvorgang entstehen SysLog-Meldungen

=> Aufnahme des Logservers in /etc/hosts

Eintrag in /etc/services:

```
syslog      514/udp
```

## Automatische Analyse der Logfiles

Verwendung der Textutils, wie grep oder awk, da der syslogd keine Programme selbst ausführt, wenn gewisse Ereignisse in die Syslog aufgenommen werden.

## Syslog-Messages selbst erzeugen

Erzeugen von Syslog-Messages mit *logger*:

*logger [-p Herkunft.Priorität] Meldungstext*

- ohne -p wird user.notice verwendet!
- statt *Meldungstext* kann man mit -f auch eine Datei angeben

Gültige Herkunftsbezeichner:

*auth, authpriv, cron, daemon, ftp, kern, lpr, mail, news, security, syslog, user, uucp, local0 bis local7*

Gültige Prioritätsbezeichner:

*alert, crit, debug, emerg, err(or), info, notice, panic, warn(ing)*

# Paketmanagement

## Paketmanagement

Ziele:

- eigenständiger Aufbau von Paketen
- Aufbau und Wiederaufbau von RPM und DEB-Paketen

Schlüsseldateien, Begriffe und Hilfsmittel:

- rpm
- SPEC-Dateiformat
- /debian/rules

## RPM

Vorgehensweise:

- einrichten des Quellcodes
- erstellen eines Patches für alle Veränderungen, die notwendig waren
- erstellen einer SPEC-Datei
- Sicherstellen, dass alle Dateien an den richtigen Orten installiert sind
- Aufbau des Paketes mit rpm

## SPEC-Datei

- enthält alle Informationen um ein RPM-Paket zu erstellen
- acht Abschnitte:
  - Präambel: alle Informationen, die beim Ausführen des rpm-Befehls angezeigt werden
  - Prep-Sektion: beginnt immer mit %prep  
einfaches Shell-Skript, das Quellcode-Dateibaum aufbaut
  - Build-Sektion: beginnt mit %build  
Shellskript um Quellcode zu kompilieren (*make*)
  - Install-Sektion: beginnt mit %install  
Shellskript (meist *make install*)
  - Files-Sektion: Angabe der Dateien, die in das RPM-Paket übernommen werden sollen  
Liste von Hand erstellen!  
alle Dateien mit vollem Pfad angeben

## SPEC-Datei

- install-/uninstall-Scripts
  - > %pre
  - > %post
  - > %preun
  - > %postun
- verify-Skript: Skript zum Überprüfen der Installation
- clean-Sektion: beginnt mit %clean  
Befehle, die nach dem Aufbau des Paketes ausgeführt werden sollen

## Beispiel:

Man möchte ein Programm a installieren, dessen Quellcode gepackt in a.tar.gz vorhanden ist.

1. Kopiere a.tar.gz nach /usr/src/rpm/SOURCES
2. erstelle SPEC-Datei, speichere dies unter /usr/src/rpm/SPECS
3. wechsle in das SPEC-Verzeichnis und führe folgenden Befehl aus:

rpm -ba a.spec (nächste Folie)

Während der Ausführung passiert dies:

- 3.1. Anweisungen aus %prep werden abgearbeitet (rpm -bp SPEC)
- 3.2. Anweisungen aus %build (rpm -bc SPEC)
- 3.3. Anweisungen aus %install (rpm -bi SPEC)
- 3.4. rpm erstellt das Binärpaket und legt es in /usr/src/rpms/RPMS
- 3.5. rpm erstellt das Source-Paket und legt es in /usr/src/rpms/SRPMS

```
Vendor: W.Eihnachtsmann
Distribution: None
Name: a
Release: 13
Packager: W.Eihnachtsmann <weihnachtsmann@nordpol.com>
Copyright: GPL
Group: unsorted
Provides: a
Autoreqprov: on
Version: 1.0.0
Summary: Eine einfache Hallo Welt Anwendung
Source: a-1.0.0.tar.gz
```

```
%description
```

```
    Diese Anwendung zeigt die grundlegende Funktionalitaet eines C-
    Compilers. Damit auch Nicht-Programmierer dieses Programm nutzen
    koennen, habe ich es hier als Binaerpaket veroeffentlicht.
```

```
# Die %prep Sektion entpackt das tar Archiv. (Statt der beiden
# Anweisungen haetten wir auch einfach %setup schreiben koennen.
```

```
%prep
```

```
    rm -rf $RPM_BUILD_DIR/hallo-1.0.0
    zcat $RPM_SOURCE_DIR/hallo-1.0.0.tar.gz | tar -xvf -
```

```
# Die %build Sektion kompiliert das Programm
```

```
%build
```

```
    make
```

```
# Die %install Sektion installiert es
%install make install

# Die %files Sektion enthält eine Liste der Dateien, die in das
# RPM-Paket integriert werden sollen.
%files
    /usr/local/bin/a

%doc
    README
```

## DEB-Pakete:

Vorgehensweise:

- einrichten des Quellcodes, Makefile und README erstellen
- aufrufen von `dh_make`
- editieren der neu erstellten Dateien in `debian/`
- Paket erstellen

## dh\_make

- fragt nach, welche Sorte des Pakets erstellt werden soll (single binary/multiple binary/library)
- Abfrage der errechneten Einstellungen
- Aufruf erstellt ein neues Verzeichnis debian/

Nun sollte man, um nach Debian-Richtlinien zu arbeiten das Makefile editieren und im “install”-Abschnitt die Dateien nach \$(DESTDIR) installieren lassen.

In debian/ sind folgende Dateien entstanden:

- README.Debian:  
Angabe der Änderungen zum Original-Paket
- copyright  
mindestens Name und Lizenz angeben, evtl. Originalquelle
- control  
Informationen über Programm, Autor, Sektion, Priorität des Paketes

## dh\_make

- rules:  
Aufbauanleitung, entspricht im wesentlichen *rules*-Datei
- menu:  
Menüeintrag für die verschiedenen Window-Manager, evtl. noch ein Icon angeben
- postinst, preinst, postrm, prerm:  
Skripte, zum installieren und deinstallieren

## Beispiel:

Man möchte ein Programm a installieren, dessen Quellcode in /usr/src/a-1.0.0 vorhanden ist.

1. Wechsel in /usr/src/a-1.0.0
2. Aufruf von dh\_make
  - 2.1. Welche Sorte des Pakets soll erstellt werden: s (single binary)
  - 2.2. Abfrage der errechneten Einstellungen:

```
Maintainer name : werauchimmer
Email-Address   : werauchimmer@irgendwo
Date            : Mon, 25 Mar 2002 16:52:35 +0100
Package Name    : a
Version         : 1.0.0
Type of Package : Single Hit <enter> to confirm:
```
3. Änderung der Makefile
4. Anpassen der Dateien in /usr/src/a-1.0.0/debian
  - 4.1. README.Debian
  - 4.2. copyright

## 4.3. control

```
Source: hallo
Section: utils
Priority: optional
Maintainer: Kai Riedmann <kr@online.de>
Build-Depends: debhelper (> 3.0.0)
Standards-Version: 3.5.2

Package: hallo
Architecture: any
Depends: ${shlibs:Depends}
Description: A programm.
    This is a simple example for the building process of debian
    packages.
    It contains one simple binary hello world programm which is
    mostly worthless for any user.
```

## 4.4. rules

```
Beispiel: http://www.linux-praxis.de/lpic2/lpi201/demo/rules
```

## 5. dpkg-buildpackage

```
5.1. a_1.0.0-1_i386.deb
5.2. a_1.0.0-1.diff.gz
5.3. a_1.0.0-1.orig.tar.gz
5.4. a_1.0.0-1_i386.changes
5.5. a_1.0.0-1.dsc
```

# Backup-Aufgaben

## Backup-Aufgaben

Ziele:

- Erstellen eines Backup-Speicherplans für ein Netz

Schlüsselfragen:

- Medienwahl
- Programmwahl
- Backupstrategie

## Backup-Strategie:

### 1. Vollbackup

Absicherung des kompletten Datenbestandes

### 2. partielles / differentielles Backup

Backup der Daten, die seit dem letzten Backup verändert wurden

#### 2.1. tar (Tape Archiver)

wichtige Funktionen:

2.1.1. erstellen: -c

2.1.2. Inhaltsverzeichnis: -t

2.1.3. entpacken: -x

weitere Flags:

2.1.4. komprimieren: -z / -j

2.1.5. Datei angeben: -f

2.1.6. verbose-Mode: -v

### 2.2. cpio (Copy I/O)

- dient der Bearbeitung von Archiven, darunter auch tar-Archiven
- flexibler, da sowohl stdin, als auch eine Dateiliste der Archiv-Dateien genommen werden kann

Modi von cpio:

copy-out:

Schreiben von Dateien in ein Archiv

copy-in:

extrahieren von Dateien aus einem Verzeichnis

copy-pass:

kopieren von Dateien innerhalb eines Archivs

### 2.3. dd (Disc Dump)

- Kopieren von Dateien von und auf Gerätedateien
- Kopieren zwischen zwei Geräten  
(Bsp. von /dev/hda nach /dev/hdc)
- Vorgehensweise:

```
dd if=<inputfile> of=<outputfile> ibs=<input-Blocksize>  
obs=<outputblocksize>
```

# LPI 201

ENDE, viel Erfolg bei der Prüfung

Kai Riedmann