

VHS – Schwabach

Einführung in Apache, CUPS und Samba

Kurs-Nr: P2824-SS07
Datum: 22.2.2007

Inhaltsverzeichnis

1	Einführung in Apache.....	3
1.1	Was ist Apache.....	3
1.2	Woher bekomme ich Apache.....	3
1.3	Installation.....	4
1.4	Konfiguration.....	4
2	Einführung in Samba.....	8
2.1	Was ist Samba.....	8
2.2	Woher bekomme ich Samba.....	8
2.3	Installation.....	8
2.4	Grundkonfiguration.....	8
2.5	Einfache Freigaben mittels Konqueror.....	9
2.6	Konfiguration „zu Fuß“.....	9
2.7	Samba-Benutzer und Zugriffsrechte.....	11
2.8	Literatur und weiterführende Weblinks.....	12
3	Einführung in CUPS.....	13
3.1	Was ist CUPS.....	13
3.2	Woher bekomme ich CUPS.....	13
3.3	Konfiguration eines lokalen Druckers.....	13
3.4	Konfiguration über das Web-Interface.....	14
3.5	Freigabe eines lokalen Druckers.....	14
4	Hilfreiche Tipps und Quellen.....	16
4.1	Linux und Hardware.....	16
4.2	Weblinks.....	16

1 Einführung in Apache



„Linux is like a wigwam – no windows, no gates, but an Apache inside“

1.1 Was ist Apache

Apache ist ein Open-Source http-Server, dient also der Auslieferung von Webseiten auf dem www-Server. Mit Hilfe der Apache-Software, ist es möglich, seinen eigenen Webserver im Intra- oder z.B. mittels DynDNS auch im Internet zu betreiben.

Ist die eigene Homepage bei einem externen Provider (z.B. 1&1, Strato, etc.) gehostet und besteht sie nur aus statischen Seiten (d.h. einzelnen html-Dateien), kann man sie ohne weiteres lokal mit Hilfe eines Webeditors erstellen und per ftp zum Provider hochladen. Gestaltet man die Website jedoch mit Hilfe von php oder benutzt ein php-basierendes CMS (Content-Management-System (z.B. Mambo/Joomla o.ä.), und man will die Website lokal testen, dann tritt Apache auf den Plan. Gleiches gilt, wenn man die Website auf dem eigenen Rechner hosten will.

Hinweis:

Wird mittels Apache ein Webserver im Internet betrieben, so sind einige Dinge zu beachten:

- Um einen zuverlässigen Dienst anbieten zu können, sollte der Rechner 24/7 durchlaufen
- Da der Rechner damit aber auch ständig vom Internet aus erreichbar ist, steigt auch das Risiko von Hackerangriffen. Der Rechner sollte also besonders abgesichert sein. **Insbesondere sollten auf ihm keine (privaten) Daten liegen.** Es bietet sich daher an, als Webserver eine eigene Maschine (am Besten in einer DMZ) zu nutzen (muss kein „Geschoss“ sein ;)
- Weiteres Studium der entsprechenden Literatur zur Absicherung des Servers ist unerlässlich (man sollte wissen, was man tut)!

Apache ist seit 1996 mit nunmehr mehr als 58% Marktanteil und über 63 Millionen Sites die am weitesten verbreitete Webserversoftware¹ und stellt damit die erfolgreichste Open-Source-Software dar. Apache läuft unter Linux/Unix, Windows, OS/2 und Netware.

Bezüglich der Herkunft der Bezeichnung „Apache“ gibt es zwei Auslegungen:

- Aus Respekt vor den Apache-Indianern.
- Apache ist ein „**A PATCH**y server“, d.h. Er besteht aus einem existierenden Kern und diversen Patch-Dateien.

1.2 Woher bekomme ich Apache

- <http://www.apache.org>
Hier gibt es stets die neueste Version (aktuell V. 2.2.4)
- Distributions-CD/DVD
Bei OpenSuSE 10.2 (V. 2.2.3-20) sind dies die Pakete:
apache2-prefork
apache2-mod_* (* steht für perl, php4, python)

¹ Quelle: Netcraft – Web Server Survey (http://news.netcraft.com/archives/web_server_survey.html), Stand 02/07

evtl: apache2-doc und apache2-example-pages
+ abhängige Pakete (werden automatisch mit installiert)

1.3 Installation

- YAST → Software installieren → Filter: Suche → „apache“
- oder: YAST → Software installieren → Filter: Schemata → Web- und LAMP-Server (hier werden zusätzlich weitere Pakete, z.B. zur Datenbankbindung mit MySQL mit installiert)

1.4 Konfiguration

1.4.1 Konfiguration über YAST

Seit SuSE 9.2 gibt es für Apache ein eigenes Modul im Kontrollzentrum (YAST2-Module → Netzwerkdienste → http-Server), das eine einfache Grundkonfiguration ermöglicht.

Zunächst genügt es, den http-Dienst zu aktivieren und ggf. **die Firewall für Port 80 freizuschalten**. Ein erster Test im Konqueror (Adresse: `http://localhost/`, alternativ: `http://<hostname>/`) sollte bereits eine Testseite zeigen.

Alternativ kann man den Apache auch als root mittels eines rc-Skriptes starten:
rcapache2 start

Für die weitere Anpassung des Indianers empfiehlt sich aber der Weg „zu Fuß“ durch die Konfig-Dateien.

1.4.2 Konfiguration durch Bearbeitung der conf-Dateien

Der Apache ist eine sehr mächtige Software mit vielen Konfigurationsmöglichkeiten. Diese erschließen sich bedeutend besser bei der direkten Manipulation der gut dokumentierten config-Dateien, als über ein Frontend wie YAST. Hinzu kommt, dass andere oder ältere Distributionen die Variante über ein Frontend gar nicht bieten.

Bei Apache 1.x gab es im wesentlichen nur eine Konfigurationsdatei (`httpd.conf`), die zwar sehr gut dokumentiert, aber ellenlang war. Mit apache2 hat sich dies grundlegend geändert. Die Konfiguration wurde nach Kategorien ausgelagert und findet sich nun verteilt in mehreren Unterverzeichnissen und Konfigurationsdateien.

Die wichtigsten Einstellungen befinden sich nach wie vor in `/etc/apache2/httpd.conf` bzw. in `default-server.conf`:

1.4.2.1 `httpd.conf`

Hier werden alle weiteren Konfigurationsdateien eingebunden. Direktiven sind hier nur wenige zu setzen.

- *DirectoryIndex index.html ~~index.html.var~~*
Nach diesen Dateien wird gesucht, wenn nur das Verzeichnis des Webservers in der Webadresse angegeben ist.
Anmerkung: `index.html.var` wird nur für die installierten Testseiten benötigt und kann nach dem Test gelöscht werden.

1.4.2.2 default-server.conf

Alle hier gesetzten Einstellungen dienen als Standard-Werte für weitergehende Konfigurationen (virtual-hosts).

- *DocumentRoot* `"/srv/www/htdocs"`
Das „Stammverzeichnis“ des Webservers. In diesem Verzeichnis befindet sich der Besucher der Webseite zunächst. Aus diesem kann er nicht heraus.
- `<Directory "/srv/www/htdocs"> </Directory>`
Hier stehen einige Direktiven für das DocRoot.
- Aliases
Mit Hilfe von Aliases kann man ein beliebiges Verzeichnis auf das DocRoot „linken“. Beispiel: *Alias /mp3/ "/home/foo/MeineMP3Sammlung/"*
Mit `http://<hostname>/mp3/` hat man (d.h., wenn der Webserver nach außen offen ist: **jeder**) Zugriff auf den Inhalt von `/home/foo/MeineMP3Sammlung`
Hinweis: Die Dateien in `/home/foo/MeineMP3Sammlung` müssen mindestens lesbar für „Others“ sein.

1.4.3 Eine kleine Test-Webseite

Um nicht in `/srv/www/htdocs` herumschreiben zu müssen (dies ist root vorbehalten), nutzen wir das Verzeichnis `/public_html` in unserem Homeverzeichnis. Dieses besondere Verzeichnis dient dazu, dass User einfach Webseiten erstellen und diese im eigenen Homeverzeichnis pflegen können. Zugriff auf diese Seiten erfolgt über die Syntax `http://<domain>/~<benutzername>`.

In der Datei `apache2.conf` bzw. in der entsprechenden Datei im Unterverzeichnis `/etc/apache2/mods-enabled` kann das Verhalten bzgl. des `public_html`-Verzeichnisses eingestellt (und auch abgeschaltet) werden.

Im Verzeichnis `public_html` erstellen² wir nun eine Datei `„index.html“` mit etwa folgendem Inhalt:

```
<html><body><h1>
  Willkommen auf der Testwebsite von foo!
</h1></body></html>
```

Die so erstellte kleine Miniwebseite ist nun schon im Konqueror mit der Adresse `http://localhost/~<username>/` bzw. `http://<ip-Adresse>/~<username>` aufrufbar!

Hinweis: Die Sicherheitseinstellungen sind zunächst zu restriktiv gesetzt. Folgendes ist zu tun, um das `userdir` zu aktivieren (als root):

- In `/etc/apache2/mod_userdir.conf`
Kommentarzeichen („#“) vor der Zeile `UserDir /home/*/public_html` entfernen
- In `/etc/apache2/httpd.conf`:
Am Ende der Datei einfügen: `Include /etc/apache2/mod_userdir.conf`
- Eventuell: Um Links auf einen Ort außerhalb des `docroots` zu ermöglichen, muss in der `httpd.conf` im Bereich `<Directory />` die Direktive von **Deny from all** auf **Allow from all** geändert werden.

² Das Bearbeiten von html-Dateien kann mit einem beliebigen Texteditor (z.B. vi, Kate), einem html-Editor (Bluefish, Eclipse) oder z.B. Openoffice Writer erstellt werden (hier: Speichern unter ... HTML-Dokument). Auch MS-Word ist hierzu in der Lage. Beim Vergleich des erzeugten html-Codes stellt man jedoch schnell fest, dass MS-Word für diesen Zweck denkbar ungeeignet ist.

1.4.4 Kleiner Exkurs: DNS

Im Netzwerk besitzt jeder Rechner eine eigene Adresse (IP-Adresse). Diese besteht aus vier, durch Punkte getrennte Bytes. Für private Netze ist z. B. der Bereich 192.168.0.0 bis 192.168.255.255 häufig anzutreffen. Nachteil: Diese Adressen kann man sich nur schwer merken. Abhilfe schafft das DNS (Domain Name System). Im Prinzip erfolgt dabei nichts anderes als eine Zuordnung von IP-Adressen zu leicht merkbaren Domainnamen.

Beispiel: 212.69.172.113 ↔ google.de.

Für kleine Netze ist ein DNS-Server aber sicher überdimensioniert. Hier kommt nun die Datei `/etc/hosts` (Windows: `C:\WINNT\system32\drivers\etc\hosts`) zum Zuge. Diese macht prinzipiell nichts anderes als ein DNS-Server – nur eben nicht zentral, sondern verteilt auf allen Rechnern im Netz.

Bei kleinen, statischen Netzen funktioniert dies auch recht gut. Überschreitet das Netzwerk aber eine gewisse „kritische Masse“ an Hosts oder kommen häufig neue hinzu, wird man die Installation eines DNS-Servers in Betracht ziehen.

Beispiel:

Wir ergänzen die Datei `hosts` um die Zeile

```
192.168.0.2      berta
```

Damit ist der Rechner Berta nicht nur über `http://192.168.0.2`, sondern – einfacher – über `http://berta` erreichbar.

Alternativ ist die Bearbeitung der Datei `/etc/hosts` auch über YAST → Netzwerkdienste → Hostnamen möglich.

1.4.5 Ausblick: Virtuelle Hosts (vhosts)

Bis jetzt ist nur eine Website unter der gegebenen IP-Adresse nutzbar. Will man mehrere Webseiten unter **einer** IP-Adresse anbieten (denn IP-Adressen sind teuer), so geschieht dies über die sogenannten „virtuellen Hosts“.

Beispiel:

Die Domains „www.websaid1.de“ und „www.websaid2.de“ sollen auf derselben Maschine gehostet werden. Folgendes bitte als „Root“ an der Konsole:

1. Wir erweitern die Datei `/etc/hosts` um diese beiden Domains (bei Angabe der gleichen IP!)
2. Wir kopieren die Datei `/etc/apache2/vhosts.d/vhost.template` und erzeugen neue (identische) Dateien mit den Namen
`www.websaid1.de.conf`
`www.websaid2.de.conf`
 Die Dateinamen sind willkürlich – wichtig ist aber die Endung `conf`.
3. Wir legen unterhalb von `/srv/www` zwei Verzeichnisse `websaid1` und `websaid2` an. Auch diese Verzeichnisnamen sind willkürlich. Die bilden das docroot unserer beiden neuen Webseiten.
4. Wir legen in die beiden neuen docroots jeweils eine Datei `index.html` mit aussagekräftigem Inhalt.
5. Zurück zu den Dateien `www.websaidX.conf` im Verzeichnis `/etc/apache2/vhosts.d`:
 In ihnen müssen zunächst zwei Einstellungen vorgenommen werden:

Direktive *Servername* *www.websaidX.de*
 Direktive *DocumentRoot* */srv/www/*
 Bereich `<Directory „/srv/www/websaidX“>`

6. Zu guter Letzt noch vhosts aktivieren: In der Datei *listen.conf* den letzten Eintrag *NameVirtualHost* * einkommentieren.
7. Server restarten mit *rcapache2 restart*
8. Das wars – zumindest die Grundkonfiguration. Für tiefergehende Kenntnisse lege ich die man apache2 sowie die Dokumentation auf apache.org ans Herz.

1.4.6 Der eigene Webserver im Internet

Gleich nochmals der Hinweis:

Wer einen Webserver im Internet (also nicht nur im Intranet zum Entwickeln und Testen) betreibt, sollte wissen, was er tut. Insbesondere sollte er (oder sie) mit Sicherheitsaspekten sowie mit der rechtlichen Seite vertraut sein.

Leider ist es bei den üblichen Providern (NefKom, T-Online, ...) nicht üblich, dass man seine eigene feste IP-Adresse erhält. Stattdessen teilt man sich dieselbe IP-Adresse mit vielen anderen Benutzern. Bei jeder Einwahl erhält man so aus einem Pool von IP-Adressen eine dynamisch zugewiesen. Während der Online-Sitzung ist man dann unter dieser auch erreichbar. Bei der nächsten Einwahl (bzw. der „Zwangstrennung“ nach 24 Stunden) ist das dann wieder eine andere. Folge: Der eigene Webserver ist nicht mehr erreichbar.

Lösung 1: Feste IP beantragen (NefKom: 14,90€ / Monat)

Lösung 2: Dienste wie z.B. DynDNS beanspruchen.

Zu DynDNS (es gibt auch noch andere):

Man holt sich (kostenlos) bei DynDNS (www.dyndns.org) eine „virtuelle Subdomain“ (z.B. websaid1.dyndns.org) und aktualisiert mittels eines Tools die unter dieser Subdomain abgelegte IP-Adresse, wenn sich diese geändert hat. Für Linux und Windows gibt es eine Vielzahl dieser Tools. Näheres bei DynDNS.org.

Tipp: Einige DSL-Router haben eine dyndns-Unterstützung eingebaut (z.B. Linksys, Netgear). Damit kann man auf eine Softwareinstallation verzichten.

Ist das Konto bei einem DynDNS-Provider eingerichtet und der eigene DSL-Router für den Abgleich der IP-Adresse konfiguriert, muss nur noch dafür gesorgt werden, dass Anfragen auf dem Port 80 (http) nicht von der routereigenen Firewall geblockt werden, sondern an die lokale IP des Webserver durchgereicht werden. Auch hierfür findet man in den meisten Routerkonfigurationen einen entsprechenden Menüpunkt.

2 Einführung in Samba



2.1 Was ist Samba

Samba ist eine OpenSource/Freie Software, die das SMB-Protokoll für Unix/Linux implementiert. Es stellt File- und Druckerdienste für Windows-Clients zur Verfügung, indem Samba einen Win-NT-Server „simuliert“. Damit ist es möglich, von einem Windows-Rechner aus auf freigegebene Verzeichnisse und Drucker des Linux-Rechners zuzugreifen.

2.2 Woher bekomme ich Samba

- <http://www.samba.org>
Hier gibt es stets die neueste Version (aktuell V.3.0.24)
- Distributions-CD/DVD
Bei SuSE 10.2 (V.3.0.23d-6) die Pakete
 - samba
 - samba-client
 - evtl. samba-doc

2.3 Installation

- YAST → Software installieren → Filter: Suche → „samba“
- oder: YAST → Software installieren → Filter: Schemata → Dateiserver
(hier werden zusätzlich weitere Pakete mit installiert)

2.4 Grundkonfiguration

Ähnlich wie bei Apache bieten sich seit SuSE 9.2 „eineinhalb“ Wege an.

2.4.1 Konfiguration über YAST (der „halbe“ Weg ;)

Die Grundkonfiguration verläuft in wenigen Schritten:

1. Namen für die (Windows!-)Arbeitsgruppe bzw. (Windows!-)Domäne eingeben. Dies ist unter Windows per Default „ARBEITSGRUPPE“. Wir übernehmen jedoch den vorgeschlagenen Namen *TUX-NET*.
2. Auswahl, ob PDC³ (Primary Domain Controller) oder nicht. Wir wollen lediglich Dateien austauschen, also wählen wir Option 2 (kein PDC).
3. Samba-Konfiguration
 - Start: Manuell oder automatisch bei Systemstart
 - Firewall-Port öffnen (**Wichtig!**)
 - NetBios-Name: Unter diesem ist der Rechner im Netzwerk sichtbar
 Der Rest kann zunächst übernommen werden.

³ Primary Domain Controller (PDC)

Dieser Server verwaltet eine Windows NT-Domäne. In einem Netzwerk können mehrere Windows NT/2000/XP Computer zu einer Domäne zusammengefasst werden. Dieses Konzept ähnelt den Arbeitsgruppen von Windows 9x/ME, ist aber nur bedingt vergleichbar. Es ermöglicht die vollständig zentrale Verwaltung des Netzwerks. Es kann direkt am PDC festgelegt werden, welche Benutzer sich mit welchem Passwort anmelden dürfen, auf welche Dateien bzw. Netzwerkfreigaben sie Zugriff haben und noch vieles mehr. Die Änderungen gelten für alle Computer, die Mitglied der Domäne sind und werden sofort wirksam.
aus Wikipedia (URL: http://de.wikipedia.org/wiki/Primary_Domain_Controller)

2.4.2 Start/Stop des Samba-Servers

Hierzu dienen bei SuSE die rcXXX-Scripte. Diese sind eigentlich nur symbolische Links (etwa: Verknüpfungen unter Windows), liegen in /usr/sbin (also im Pfad) und zeigen auf ein Script in /etc/init.d.

Ein Aufruf von rcsmb als root zeigt die möglichen Argumente. Die wichtigsten:

```
rcsmb status      Lläuft der Server oder nicht?
rcsmb start       Startet den Server
rcsmb stop        ...
```

2.4.3 Testen des Servers

Wir geben im Konqueror *smb://<NetBiosNameDesRechners>* ein und sollten die drei Standardshares groups, profiles und users sehen⁴. Im Netzwerk können wir statt des NetBios-Namens auch die IP-Adresse setzen. Sinnvoll ist ein Eintrag in /etc/hosts bzw. /etc/samba/lmhosts (Linux), C:\WINNT\system32\drivers\etc\hosts bzw. lmhosts (Win2k) nach dem Schema

```
<IP-Adresse>    <NetBiosName> (lmhosts)
<IP-Adresse>    <Hostname> (hosts)
```

Alternative: Einrichtung eines DNS-Servers, siehe 1.4.4

2.5 Einfache Freigaben mittels Konqueror

Über Rechtsklick auf einem (eigenen) Verzeichnis, lässt sich dieses per Mausklick freigeben. Das Recht hierzu muss nur einmalig mit root-Rechten vergeben werden. Dann ist die Freigabe aber auch fast schon trivial und das Vorgehen selbsterklärend.

Diese Vorgehensweise ist vielen aus Windows bekannt und wird für viele einfache Belange ausreichen. Samba kann aber noch viel mehr. Interessant wird die folgende Konfiguration „zu Fuß“ dann, wenn es um Nutzergruppen und -berechtigungen geht. Allerdings schadet es auch nicht, etwas „unter die Haube zu schauen“ ;)

2.6 Konfiguration „zu Fuß“

2.6.1 Die Dateien in /etc/samba

- **lmhosts**
Spielt die gleiche Rolle wie die entsprechende Datei unter Windows: Eine Zuordnung von NetBios-Namen zu IP-Adressen (ähnlich wie die Datei hosts)
- **secrets.tdb**
Hier steht die Domänen-SID, also das Maschinenkennwort, mit dem sich der Server dem Client gegenüber (und umgekehrt) ausweist.
- **smbfstab**
Will man unter Linux auf eine Netzfreigabe zugreifen, so geschieht dies z.B. an der Shell mit dem Befehl *mount* bzw. *smbmount*. Nachteil: Ist die entfernte Freigabe mit einem Benutzernamen/Passwort abgesichert, muss man diese Kombination dem Befehl *smbmount* als Argument übergeben. In der smbfstab kann man gebräuchliche Freigaben bei Serverstart automatisch mounten lassen. Das Passwort bekommt dann niemand mehr zu Gesicht.

⁴ Hinweis: Unter Windows muss man evtl. eine Weile warten, bis der neue Rechner in der Netzwerkumgebung auftaucht. Schneller geht es, wenn man im Windows-Explorer \\<NetbiosNameDesRechners> eingibt, oder ihn im Netzwerk sucht.

- **smbpasswd**
Hier stehen die Samba-Benutzer mit ihren verschlüsselten Passwörtern. Genaueres hierzu später.
- **smbusers**
Der Name der Datei führt leicht in die Irre. Es stehen hier nämlich **nicht** die Usernamen der Samba-Benutzer, sie dient vielmehr der Abbildung (Mapping) von Windows-Usern auf die Linux-Umgebung.
Beispiel: root = Administrator admin
Der Windows-Administrator-User wird also dem Linux-Root-User zugeordnet.
- **smb.conf**
Die Kern-Konfigurationsdatei für den Samba. In ihr werden globale Deklarationen vorgenommen und die Freigaben (Shares) mit den entsprechenden Direktiven z.B. bzgl. Zugriffsberechtigung definiert.

2.6.2 Der Aufbau der smb.conf

2.6.2.1 Grundsätzliches

Wie üblich, ist die smb.conf eine reine Textdatei, kann also mit jedem beliebigen Texteditor bearbeitet werden. Sie gliedert sich in zwei (eigentlich drei) Abschnitte. Der [global]-Sektion, den Standard-Shares und selbstdefinierten Shares.

Es gibt sehr viele Direktiven für die smb.conf, die hier unmöglich behandelt werden können. Ein Hinweis auf die fast 6000 Zeilen lange *man smb.conf* sei erlaubt. ;) Beim händische Tippen schleichen sich manchmal Tippfehler ein. Der Befehl *testparm* überprüft die smb.conf auf Syntaxfehler (nicht auf semantische!)

2.6.2.2 Die [global]-Sektion

Hier stehen die grundlegenden Einstellungen, wie der NetBios-Name, der Name der Windows-Workgroup, ob PCD oder nicht, etc. Auch Druckerfreigaben über CUPS sind bereits vorbereitet, falls man Samba über YAST installiert hat. Beispielsweise findet sich hier die oben mittels YAST definierte Workgroup.

2.6.2.3 Besondere Freigaben

- **[homes]**
Stellt eine Freigabe auf das Home-Verzeichnis auf dem Linux-Server zur Verfügung. Natürlich müssen Benutzername in den beiden Welten übereinstimmen oder mittels smbuser aufeinander gemappt sein.
- **[profiles]**
Diese Share ist wichtig, wenn der Samba als PDC fungiert und die User ihre Benutzerprofile auf dem Server ablegen. (Hier obsolet)
- **[users]**
Eine Freigabe für das /home-Verzeichnis.
- **[groups]**
Eine Freigabe für Gruppenverzeichnisse (bei uns nicht vorhanden, also obsolet).
- **[printers] und [print\$]**
Wird bei CUPS wichtig.

2.6.2.4 Wichtige Direktiven

- path = <Existierendes Verzeichnis>
Auf dieses Verzeichnis zeigt die Freigabe
- readonly = [yes|no]
- browseable = [yes|no]
Freigabe wird gelistet oder muss in Kenntnis ihres Namens aufgerufen werden
- valid users = <Liste mit Usern oder @Usergruppen>
Gibt bestimmten Usern oder -gruppen Zugriff auf eine Share.
- create mask = <Bitmaske> (z.B. 660)
Definiert Berechtigungen für neu erstellte Dateien in dieser Share
- directory mask = <Bitmaske>
dito für Verzeichnisse
- guest ok = [yes|no]
Zugriff auf Share auch ohne Benutzername/Passwort?

2.6.2.5 Einfache Freigaben

Will man einfache Freigaben ohne besondere Zugriffsbeschränkungen definieren, gibt es zwei Wege:

1. Man definiert in der [global]-Sektion die Direktive *security = share*. Damit definiert man die Shares auf Freigabeebene, d.h. Die Benutzer haben auf die Share OHNE Passwordeingabe Zugriff.
2. Man erlaubt einen Gast-Zugriff mittels der Direktive *guest ok = yes* in der einzelnen Share.

Eine schöne Dokumentation hierzu ist 2.8 [4].

2.6.2.6 Komplexe Freigaben

Der Samba-Server ist eigentlich dafür vorgesehen, z. B. in Firmen seinen dienst als Fileserver zu verrichten. In diesem Umfeld sind natürlich wesentlich komplexere Freigabekonfigurationen zu finden – schließlich gilt es, die Firmenstruktur auf eine Ordnerstruktur abzubilden. Einen kleinen Einblick bietet das folgende Kapitel. Auf weiterführende Literatur sei an dieser Stelle wieder einmal verwiesen!

2.7 Samba-Benutzer und Zugriffsrechte

2.7.1 Samba-Benutzer anlegen

Grundsätzlich gilt: Für jeden Samba-User muss ein Linux-User mit gleichem Benutzernamen existieren. Die Passwörter dürfen sich aber unterscheiden.

Ein Samba-User wird angelegt mittels
`smbpasswd -a <username>`

Hinweis: Leider gibt es hier eine Namensgleichheit zwischen dem Programm `smbpasswd` und der Datei `/etc/samba/smbpasswd`. Das Programm `smbpasswd` manipuliert die Datei `/etc/samba/smbpasswd`.

2.7.2 Unix-Rechte und Samba-Rechte

Samba bringt seine komplette eigene Rechteverwaltung für den Zugriff auf Shares (benutzer- oder maschinenbezogen) wie auch für die Dateien und Verzeichnisse im Share selbst mit.

Dabei gilt grundsätzlich: Die effektiven Rechte, die eine Datei hat, ist eine bitweise AND-Verknüpfung der Samba- und Linux-Berechtigungen. Mit anderen Worten: Was das eine System an Restriktionen erteilt hat, kann das andere System nur verschärfen, nicht aber aufweichen.

2.7.3 Freigaben mit Benutzerrechten

Zum Testen erstellen wir nun eine Freigabe, über die wir Dateien im Netz bereitstellen. Schreibzugriff sollen nur wir und die Benutzergruppe Kumpels haben. Die Chefetage sowie die neugierige Sekretärin bleiben ausgesperrt.

1. Erstellen des Verzeichnisses `/home/foo/tauschverzeichnis` unter Linux
2. Definition der Share

```
[tausch]
path = %H/tauschverzeichnis
readonly = yes
invalid users = bigboss @chefetage
browsable = yes
create mask = 0611
#Das Folgende überschreibt die Direktive readonly
write list = meinereiner @kumpels
```
3. Damit Samba die Änderungen übernimmt, muss man ihn mittels `r smb restart` neu starten, oder ihn zumindest dazu bringen, mit `r smb reload` die Konfiguration neu einzulesen.

2.8 Literatur und weiterführende Weblinks

1. KÜHNEL, JENS: *Samba 3*, Mitp-Verlag; Auflage: 1 (Februar 2004), ca. 30 €
2. Das deutsche Samba-Howto
<http://gertranssmb3.berlios.de/output/>
3. Samba in 21 Tagen – Webdokument
<http://www.64-bit.de/linux.html> → Netzwerk → Samba
4. Sehr schöne Dokumentation zur Konfiguration
http://www.linuxhilfen.org/netz/samba_konfig.html
5. <http://samba.sernet.de/>
6. Natürlich die entsprechenden man-Pages

3 Einführung in CUPS

3.1 Was ist CUPS

Früher gab es unter Unix im wesentlichen zwei Drucksysteme, den LPD (Berkeley Line Printer Daemon) sowie das AT&T Line Printer System. Wie aus den Bezeichnungen hervorgeht waren dies Systeme, die für Lineprinter (also Nadel- oder Typenraddrucker) entwickelt wurden.

Als Nachfolger wurde das Common Unix Printing System entwickelt, das sich seit einigen Jahren zum Standard für das Drucken unter Unix/Linux entwickelt hat. Es verwendet u. a. das IPP-Protokoll (Internet Printing Protokoll) zum Ansteuern von Druckern.

3.2 Woher bekomme ich CUPS

Die Heimat von CUPS ist <http://www.cups.org>. Hier gibt es die neuesten Releases. Bei praktisch allen aktuellen Distributionen ist CUPS jedoch dabei bzw. kann problemlos von CD nachinstalliert werden.

3.3 Konfiguration eines lokalen Druckers

3.3.1 Erkannter Drucker

Die Installation eines lokalen Drucker via YAST ist einfach:
YAST → Hardware → Drucker

Fall ein bekannter Drucker angeschlossen und eingeschaltet ist, erkennt ihn YAST wahrscheinlich korrekt. In diesem Fall ist die Konfiguration damit abgeschlossen.

TIPP:

Falls der Drucker trotz korrekter Erkennung nicht drucken sollte, liegt es wahrscheinlich an einer BIOS-Einstellung. Dort sollte die das Protokoll für den Parallelport unbedingt auf *Normal*, *Bidirektional* oder *EPP* eingestellt sein. **Keinesfalls** jedoch auf *ECP*.

Update: SuSE 10.2 kann wohl auch mit ECP umgehen (nicht persönlich getestet).

3.3.2 Nicht erkannter oder ausgeschalteter Drucker

Andernfalls (wie auch bei Netzwerkdruckern) muss man den Drucker von Hand konfigurieren:

Nach Eingabe des Namens der Warteschlange gelangt man zur Auswahl des Druckers bzw. des Druckertreibers (Knopf *PPD-Datei* rechts unten). Für viele Drucker existieren mehrere Druckertreiber für unterschiedliche Anforderungen.

An dieser Stelle ist es auch leicht möglich, einen neuen Treiber (PPD-Datei) in das System einzuspielen. Als Quelle für PPD's bieten sich an

- Webseiten der Druckerhersteller (Epson, hp, ...)
- **Die** Anlaufstelle für Drucken unter Linux schlechthin:
<http://www.linuxprinting.org>

3.4 Konfiguration über das Web-Interface

Durch Eingabe von `http://localhost:631/admin` gelangt man zum Webinterface von CUPS. Hier kann man z. B. Druckaufträge überwachen, neue Drucker anlegen, bestehende konfigurieren, etc.

Hinweis für ältere SuSE-Distributionen (< 10.2): Die Sicherheitseinstellungen sind hier sehr restriktiv gesetzt. Nicht einmal root darf hier alles. Um dennoch die Drucker komplett über das Webinterface administrieren zu können, ist gibt man an der Konsole folgendes ein:

```
lppasswd -g sys -a root
```

und anschließend ein Passwort (muss nicht identisch dem root-Passwort sein).

3.5 Freigabe eines lokalen Druckers

3.5.1 Vorbereiten der smb.conf

In der smb.conf wurde bereits zwei Shares eingerichtet, die für das Netzwerkdrucken benötigt werden. Hier ist also eigentlich nichts zu tun.

- [print\$]
Diese Share zeigt auf `/var/lib/samba/drivers`. Hier stehen werden die Windows-Druckertreiber hineingelegt, falls man von einem Windows-Rechner aus auf einen Linux-Drucker zugreifen möchte und die Windowstreiber zentral lagern möchte.
- [printers]
Dies ist eine besondere Share, die nicht auf ein Verzeichnis zeigt (`path = ...`) zeigt auf das Spool-Verzeichnis. Durch die Direktive `printable = yes` ist die Share als Druckershare deklariert. Diese Share holt sich aus der Datei `/etc/printcap` die definierten lokalen Drucker und stellt sie über Samba im Netzwerk bereit.

3.5.2 Testen der Konfiguration

Die Eingabe von `smbclient -L //<NetBiosName>` sollte einen Ausdruck ähnlich diesem bringen

```
Domain=[TUX] OS=[Unix] Server=[Samba 3.0.7-5-SUSE]

      Sharename      Type      Comment
      -
      profiles      Disk      Network Profiles Service
      users          Disk      All users
      groups         Disk      All groups
      print$         Disk      Printer Drivers
      tausch         Disk      Tauschverzeichnis
      IPC$           IPC       IPC Service (Samba 3.0.7-5-SUSE)
      ADMIN$        IPC       IPC Service (Samba 3.0.7-5-SUSE)
      testdrucker1   Printer   Epson Stylus Color 760 Foomatic/gimp-print
(recommended)
Domain=[TUX] OS=[Unix] Server=[Samba 3.0.7-5-SUSE]

      Server          Comment
      -
      Workgroup       Master
      -
```

3.5.3 Zugriff vom entfernten Rechner aus

Über YAST → Hardware → Drucker → Hinzufügen.

Hier gibt es mehrere Möglichkeiten, an seinen Drucker zu kommen. Wir wählen die einfache, von Windows gewohnte Methode: *Über SMB-Netzwerkserver drucken*.

Anschließend werden wir nach einigen Angaben gefragt. Bei dem Namen der Arbeitsgruppe und der „entfernten Warteschlange“ (gemeint ist der „Sharename“ des Druckers) erweist sich der vorangegangene Test per smbclient als nützlich.

Anschließend sollte der Drucker über das Netz ansprechbar sein.

4 Hilfreiche Tipps und Quellen

4.1 Linux und Hardware

Aufgrund der Tatsache, dass viele Hardwarehersteller nur Windows-Treiber ausliefern, ist es sinnvoll, sich **vor dem Kauf** neuer Hardware darüber zu informieren, ob diese unter Linux unterstützt wird.

Idealerweise kann man die Hardware direkt beim Händler mit Hilfe einer Knoppix-CD testen. Wo dies nicht möglich ist, muss man die diversen Foren im Internet bemühen.

4.2 Weblinks

- **Linux User Schwabach - <http://www.lusc.de/>**
Stammtisch jeden 1. und 3. Donnerstag im Monat (Gaststätte Gartenlaube)
April 2007: Schwabacher Linxstage
Weitere Veranstaltungen siehe Website

4.2.1 Bei Hardwareproblemen

- SuSE Hardwaredatenbank
<http://cdb.novell.com>
- <http://www.tuxhardware.de/>
- Speziell zum Thema Drucken
<http://www.linuxprinting.org/>
- Zum Thema Notebooks unter Linux:
<http://www.linux-laptop.net/>
<http://tuxmobil.org/>

4.2.2 Softwarequellen

- <http://www.links2linux.de/>
- <http://sourceforge.net/>

4.2.3 e-books

- Linux-Fibel
<http://www.linuxfibel.de/default.htm>
- Linuxbu.ch online
<http://www.linuxbu.ch/>

4.2.4 Sonstiges

- Linux-Befehlsübersicht für die Shell
<http://www.tnt-computer.de/yanip/lbefehle.html>
- Hilfe-Foren
Pro-Linux: http://www.pl-forum.de/t_system/
- Linuxhilfen: <http://www.linuxhilfen.org/>
dort gibt es eine gut verständliche Samba-Erklärung
- Und schließlich - die Linux-Suche von Google:
<http://www.google.de/linux/>