Abschlussprojekt

Aufbau und Konfiguration eines Domänencontrollers mit Samba

Fachlehrer: Herr Scheib

BBSNW
Klasse: FSINF 06
Ramona Ludwig
Mai 2010

Inhaltsverzeichnis

1. Was ist ein Domaincontroller (DC) 2
2. Aufgaben eines Domaincontrollers 2
3. Warum ein Domaincontroller unter Linux? 3
4. Was ist Samba? 4
5. Samba 5
5.1 Installation der Software 5
5.2 Samba als Domaincontroller einrichten 5
6. Linuxuser anlegen 12
7. Sambauser anlegen 14
8.Windowsuser in Linux / Samba aufnehmen 16
9. Verzeichnisse anlegen 21
10. Rechtevergabe 22
10.1 Rechtevergabe serverseitig (Linuxrechte / ACL) 22
10.1.1 Linux - Rechte 22
10.1.2 Spezialbits (Setuid- Bit, Setgid- Bit, Sticky- Bit) 25
10.1.3 ACL (Access Control Lists) 27
10.2 Rechtevergabe clientseitig (über SMB) 33
11. Laufwerksmapping 36
12. Skripte 39
13. Roaming Profiles 40
14. Drucker anbinden 41
15. Fazit 44
16. Literaturverzeichnis 46
17. Anhang 49
17.1 Aufbau der Testumgebung 49
17.2 Problemstellung 52
17.3 Kommentar 54

1. Was ist ein Domaincontroller (DC)

Ein Domaincontroller (auch Domänencontroller) ist lt. Wikipedia, der freien Enzyklopädie, ein Server, der Computer und Benutzer in einem Rechnernetz zentral authentifiziert und autorisiert.[1]
Durch ihn ist eine zentrale Verwaltung einer Domain möglich.

> „Als Domain (Domäne) werden logisch abgegrenzte Bereiche eines Netzwerks bezeichnet. Domains können hierarchisch strukturiert sein.“[2]

Der DC wird auch als Primary Domain Controller (PDC) bezeichnet, d. h., dass er als Hauptserver fungiert. Für eine mögliche Sicherungskopie der Datenbank dient ein zweiter Server. Er wird Backup Domain Controller (BDC) genannt.[1]

2. Aufgaben eines Domaincontrollers

- Verwaltung der Benutzer, Dateien und Rechte
- Bereitstellen von Anmeldeskripten (Loginskripten)
- Benutzerprofile (Roaming Profiles)
- Ressourcen der Domäne über logische Laufwerke zur Verfügung stellen[3]

Ein Domaincontroller überwacht die Zugriffe von Usern auf den Server und auf die Ressourcen. Er überprüft dabei die Gültigkeit und die Berechtigung der Anmeldung des Users. Die dafür benötigten Daten sind in einer Datenbank gespeichert. Er weiß, welcher User welche Benutzerdaten (inkl. Passwort) hat und welche Ressourcen von wem benutzt werden dürfen.
Jeder Benutzer kann sich an jedem Rechner in der Domäne mit seinen eigenen Zugangsdaten anmelden, durch die zentrale Datenbank. Alle Daten liegen auf dem Server und sind für den jeweiligen Benutzer frei verfügbar, sobald er sich auf dem Server angemeldet hat.

---

[1] Vgl. Meißner in Wikipedia (2010)
[2] Vogel (2005)
[3.] Vgl. Röhl (2004)

Vorteile eines Domaincontrollers:

- Ein Benutzer kann sich nur noch mit gültigem Namen und Kennwort am Netzwerk anmelden und erhält dann gezielt Zugriff auf die benötigte Arbeitsumgebung
- Benutzer mit gleicher Aufgabenstellung können zu Gruppen zusammengefasst werden und erhalten über ihre Gruppenzugehörigkeit Ihre Rechte im Netzwerk
- Änderungen oder Neuerungen werden zentral an einer Stelle vorgenommen und nicht an jedem Rechner lokal
- Optimale Nutzung vorhandener Ressourcen
- Ein oder mehrere Administratoren verwalten die Benutzer und Gruppen zentral

3. Warum ein Domaincontroller unter Linux?

Die Testumgebung arbeitet mit der Linuxdistribution OpenSuse 11.2, einer 64-Bit Version. Darin ist die Software Samba 3.4.2 automatisch enthalten. Ist die Software bereits installiert, kann Samba als Domaincontroller konfiguriert werden, ohne weitere Softwarekosten.

Vorteile von Linux:

- Linux / Samba ist kostenlos erhältlich (z. B.: http://de.opensuse.org/, www.samba.org/), daher kostengünstige Serverlösung
- Open Source, d. h. der Quelltext ist frei verfügbar und darf erweitert oder geändert werden (GPL &Co.)[4]
- Keine teueren Lizenzen nötig wie bei Windowsnetzwerken
- Minimale Installation des Betriebssystems reicht, um als Server zu arbeiten.
- Durch wenig installierte Dienste kein Virenprogramm nötig, da
  – die meisten Viren für Microsoft - Systeme geschrieben sind
  – je weniger Dienste laufen, desto weniger Schwachstellen möglich

[4] Vgl. Kofler (2010) S. 33

  - Update meist schneller als Virenattacke, da jeder mithelfen kann, das System ständig zu verbessern
- Meist geringere Hardwareanforderungen als bei Windows
- Je nach Anforderungen ist es ausfallsicherer als Windowssysteme[5]
- Update im laufenden Betrieb möglich[6]

4. Was ist Samba?

Der Name Samba kommt von der Abkürzung SMB. Leider war die Bezeichnung „SMB“ schon vergeben und auf der Suche nach einem geeigneten Namen wurde aus SMB dann Samba.

SMB (Server Message Block) und auch das neuere, ähnlich funktionierende, CIFS (Common Internet File System) sind Entwicklungsprodukte von Microsoft. Da der Quellcode nicht offen gelegt wurde, sind die Funktionen nur nachgeahmt, was klar macht, weshalb Samba ständig ergänzt und verbessert wird.
SMB /CIFS sind Protokolle. Sie werden von WIN- Betriebsystemen zur Kommunikation von Dateisystemen benutzt, also zum Austausch von Daten. Damit Linux mit Windows – Clients kommunizieren kann, müssen sie die gleiche Sprache sprechen und Samba kann das.
Samba ist ein Programm, welches SMB / CIFS unter Linux ermöglicht und somit für den Datenaustausch sorgt. Dadurch dient das Programm als Datei- und Druckserver für, am Netzwerk, angeschlossene Clients. Es wird keine zusätzliche Software benötigt.
Aber Samba kann nicht nur Dateien und Verzeichnisse frei geben, sondern fungiert als Primary Domain Controller. Clients, die sich am Netzwerk anmelden wollen, werden durch Samba authentifiziert und autorisiert und bekommen ihre benutzerspezifische Oberfläche (Roaming Profiles) zugewiesen.
Samba funktioniert mit den Dämonen (Hintergrunddienste) smbd und nmbd.

---

[5] Vgl. Kofler (2010) S. 31
[6] Vgl. Kofler (2010) S. 72

Dabei bietet smbd die eigentlichen Datei- und Druckerdienste an. Nmbd ist für die Namensdienste zuständig. Auf der Konsole wird Samba mit den Befehl rcsmb start gestartet.
Siehe dazu auch Punkt 5.2 Samba als Domaincontroller einrichten.

5. Samba

5.1 Installation der Software

In Yast unter dem Menüpunkt Software lassen sich Softwarepakete, meiner Meinung nach, sehr einfach installieren. Die grafische Oberfläche ist selbst - erklärend. Im Untermenü Software installieren oder löschen nach Samba suchen. Für das Einrichten eines Sambaservers alle Pakete auswählen und installieren. Eine Abfrage von OpenSuse wegen eventuell zusätzlich zu installierenden Paketen, zur Auflösung von Abhängigkeiten, mit „Ja“ bestätigen. In OpenSuse 11.2 ist Samba bereits vorinstalliert.

Die Internetadresse „http://samba.sernet.de/“ bietet immer die aktuellste Version von Samba an, aber auch ältere Versionen werden unter dem Link „Dokumentation“ noch supportet.

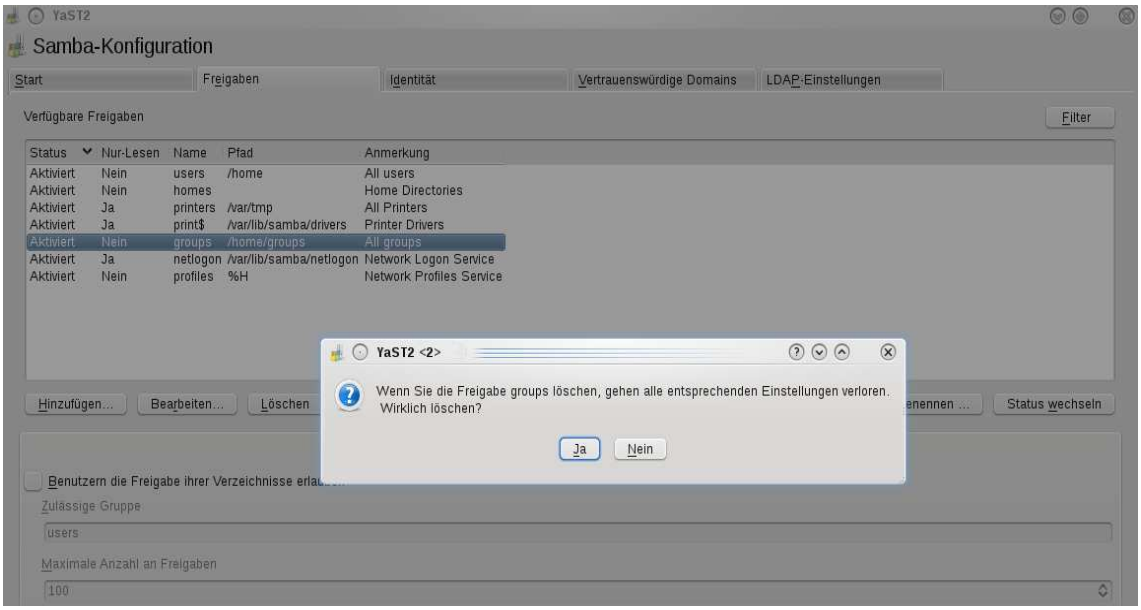
5.2 Samba als Domaincontroller einrichten

In Yast unter Netzwerkdienste im Untermenü Sambaservers auswählen. Zuerst muss man den Namen der Domain angeben (z. B. SAMBALU) und dann den Samba- Servertyp auswählen: Primary Domain Controller.

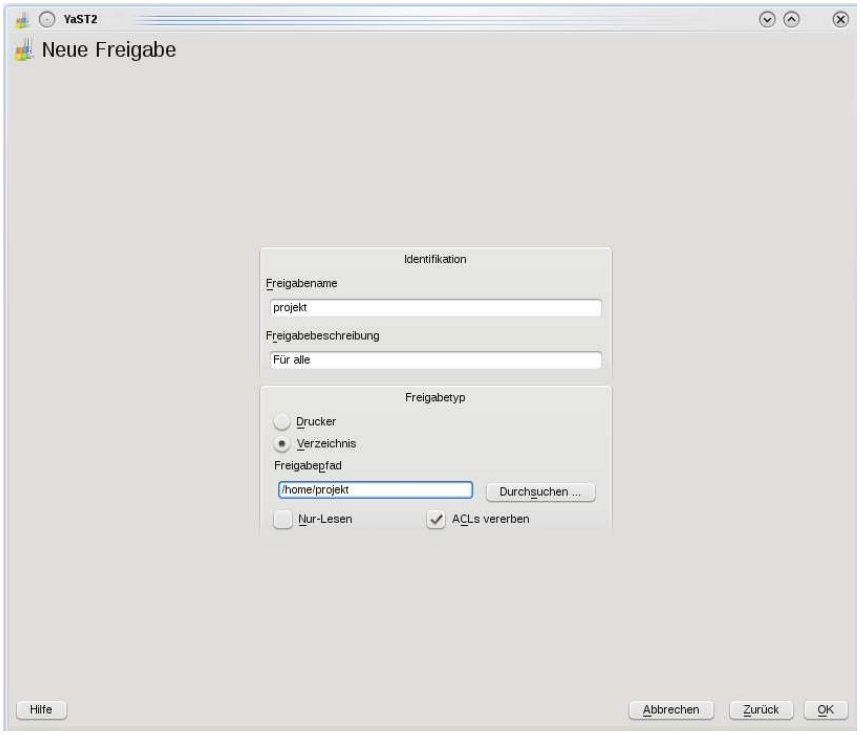
In der Samba – Konfiguration sind drei Karteireiter, für das kleine Testnetzwerk, von Bedeutung: Start, Freigaben und Identität. Im ersten Reiter findet folgende Einstellung statt:

- Beim Systemstart aktivieren, damit fällt das manuelle Starten weg.
- Firewall - Ports öffnen, damit die Windowsclients auf den PDC zugreifen können.

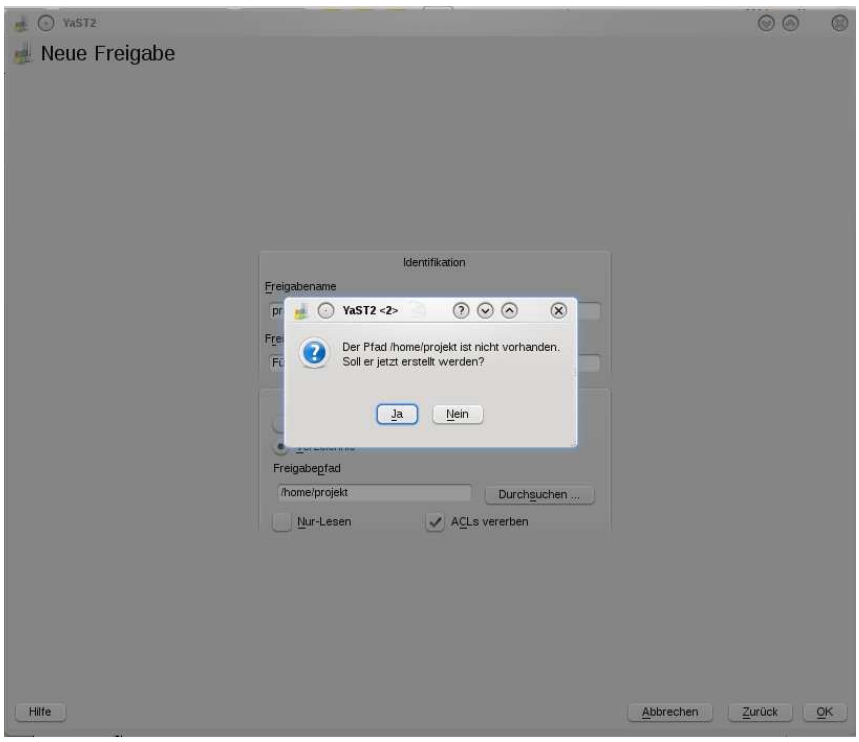
Im Register Freigaben hat Yast schon ein paar Einträge vorgegeben. Z.B. sind „printers“ und „print$“ nötig, um den Clients Zugriff auf den Netzwerkdrucker zugeben. Die vordefinierten Shares können geändert, ergänzt oder gelöscht werden. Wenn eine Freigabe gelöscht werden soll, fragt Yast noch einmal nach.



Um eine zusätzliche Freigabe hinzuzufügen, muss der Name, die Beschreibung und der Pfad angegeben werden.

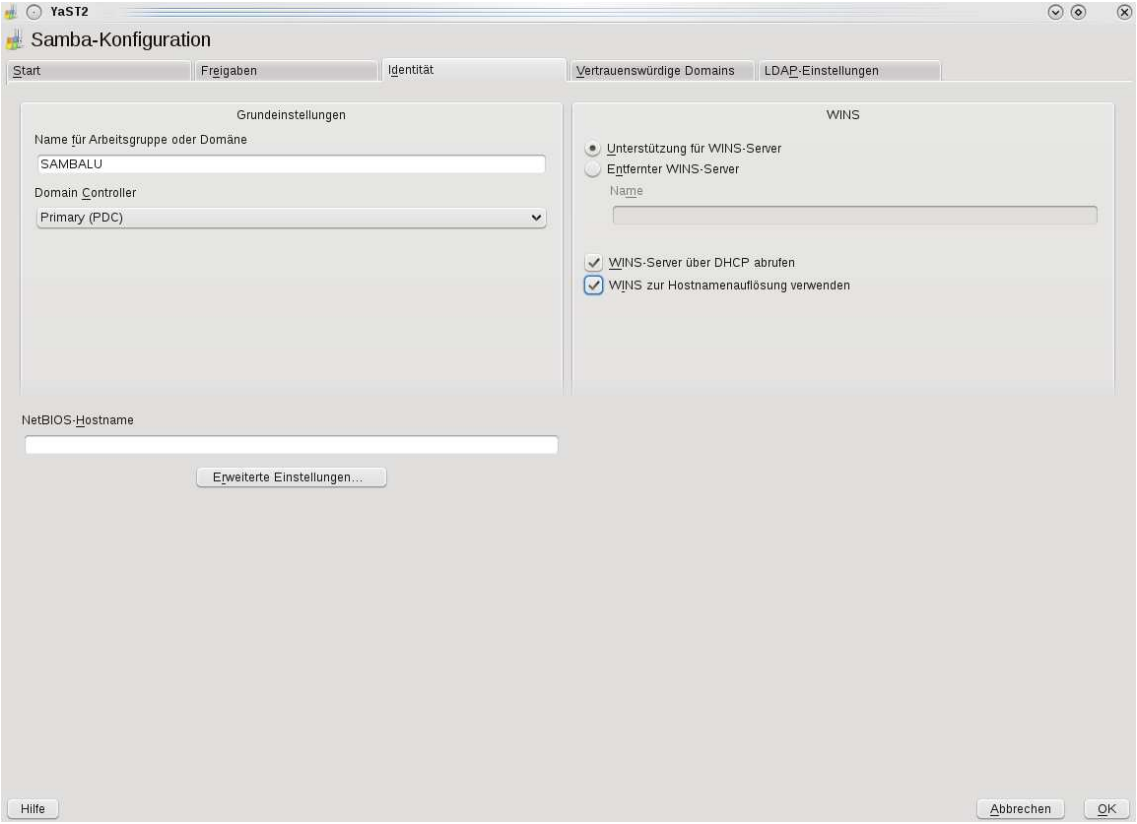


Da das Verzeichnis /home/dokumente nicht existiert, fragt Suse nach, ob es erstellt werden soll.

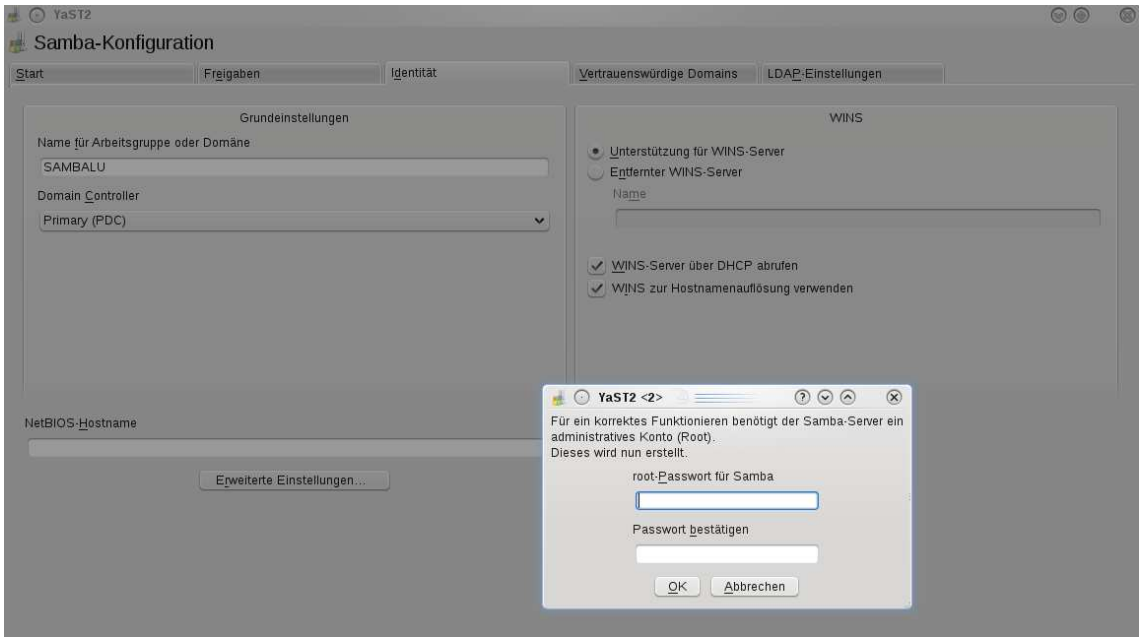


Im Register Identität sieht man noch mal die getroffenen Grundeinstellungen und muss rechts daneben folgende Einträge aktivieren:

- Unterstützung für WINS – Server (für das kleine Testnetzwerk reicht diese Einstellung)
- WINS – Server über DHCP abrufen
- WINS zur Hostnamenauflösung verwenden (Die Funktion des WINS – Server übernimmt der Samba – Daemon „nmbd“)



Alle getroffenen Einstellungen werden mit OK bestätigt. Zum Schluss wird nach dem Root – Passwort für den Sambaserver gefragt. Dieses Passwort braucht root später, um vom Windows- Client in die SAMBALU- Domäne eintreten zu können. Siehe auch Kapitel 8. Windowsuser in Linux / Samba aufnehmen.



Mit Eingabe des Passwortes ist die Konfiguration abgeschlossen und die notwendigen Hintergrundprozesse werden gestartet:

- **nmbd** dient zur internen Verwaltung und als Nameserver. Der Dämon kümmert sich auch um die Browsing – Funktion. Nmbd kann auch als Master – Browser oder als WINS – Server fungieren[7]
- **smbd** stellt die Schnittstelle für die Clients dar und gibt diesen Zugang zu Verzeichnissen, Druckern und zur aktuellen Browsing – Liste[8]

Damit **nmbd** die Namensauflösung durchführen kann, muss der Sambaserver wissen, welcher Rechner über welche IP- Adresse erreichbar ist. Diese Infor - mationen werden in die /etc/hosts - Datei geschrieben.
Der Sambaserver heißt „samba“, der XP- Client nennt sich „bie“ und der Windows7- Client „win7“. Siehe dazu auch Abschnitt 8. Windowsuser in Linux / Samba aufnehmen.

---

[7] Vgl. Kofler (2010) S. 962
[8] Vgl. Kofler (2010) S. 962

```
bbs:bash
Datei  Bearbeiten  Ansicht  Verlauf  Lesezeichen  Einstellungen  Hilfe
# hosts         This file describes a number of hostname-to-address
#               mappings for the TCP/IP subsystem.  It is mostly
#               used at boot time, when no name servers are running
#               On small systems, this file can be used instead of
#               "named" name server.
# Syntax:
#
# IP-Address  Full-Qualified-Hostname  Short-Hostname
#

127.0.0.1       localhost

# special IPv6 addresses
::1             localhost ipv6-localhost ipv6-loopback

fe00::0         ipv6-localnet

ff00::0         ipv6-mcastprefix
ff02::1         ipv6-allnodes
ff02::2         ipv6-allrouters
ff02::3         ipv6-allhosts
127.0.0.2       linuxlu.site linuxlu
192.168.178.20  samba
192.168.178.21  bie
192.168.178.22  win7
```

Die Konfiguration des Sambaservers, die auf der grafischen Oberfläche vorgenommen wurde, ist von Suse in die Datei /etc/samba/smb.conf geschrieben worden. Diese Datei sieht so aus:

```
bbs:bash
Datei  Bearbeiten  Ansicht  Verlauf  Lesezeichen  Einstellungen  Hilfe
linuxlu:/home/bbs # cat /etc/samba/smb.conf
# smb.conf is the main Samba configuration file. You find a full commented
# version at /usr/share/doc/packages/samba/examples/smb.conf.SUSE if the
# samba-doc package is installed.
# Date: 2009-10-27
[global]
        workgroup = SAMBALU
        passdb backend = tdbsam
        printing = cups
        printcap name = cups
        printcap cache time = 750
        cups options = raw
        map to guest = Bad User
        include = /etc/samba/dhcp.conf
        logon path = \\%L\profiles\.msprofile
        logon home = \\%L\%U\.9xprofile
        logon drive = P:
        usershare allow guests = Yes
        add machine script = /usr/sbin/useradd  -c Machine -d /var/lib/nobody -s /bin/false %m$
        domain logons = Yes
        domain master = Yes
        local master = Yes
        os level = 65
        preferred master = Yes
        security = user
        wins support = Yes
[homes]
        comment = Home Directories
        valid users = %S, %D%w%S
        browseable = No
        read only = No
        inherit acls = Yes
[profiles]
        comment = Network Profiles Service
        path = %H
        read only = No
        store dos attributes = Yes
        create mask = 0600
        directory mask = 0700
[printers]
        comment = All Printers
        path = /var/tmp
        printable = Yes
        create mask = 0600
        browseable = No
[print$]
        comment = Printer Drivers
        path = /var/lib/samba/drivers
        write list = @ntadmin root
        force group = ntadmin
        create mask = 0664
        directory mask = 0775

[netlogon]
        comment = Network Logon Service
        path = /var/lib/samba/netlogon
        write list = root

[projekt]
        comment = Für alle
        inherit acls = Yes
        path = /home/projekt
        read only = No
```

Diese Konfigurationsdatei besteht aus zwei Teilen. Dem [global]- Bereich, der zentrale und globale Einstellungen vorgibt und den Shares (Freigaben). Sie legen die Bedingungen fest, die für die einzelnen Freigaben gelten sollen, die nicht schon in der [global]- Sektion definiert wurden.

In der [global]- Share sieht man z. B., dass der Sambaserver SAMBALU heißt und in der Domäne der PDC ist (domain master = yes), der Drucker durch „CUPS“ (Common Unix Printing System) unterstützt wird, die User- Level-Security (security = user) eingestellt ist und die Namensauflösung (wins support = yes) unterstützt wird.

Die [home]- Share beinhaltet die Heimatverzeichnisse der User und gibt diese für sie frei. Die Homedirectories liegen meist in „/home/homedirectoyname“.

Die [profiles]- Share beinhaltet die Benutzerprofile. Für jeden Benutzer liegt ein Profil- Verzeichnis vor. Durch den Eintrag „store attributes = yes“, wird bewirkt, dass DOS- Attribute (wie z. B. versteckte Dateien) erhalten bleiben.[9]

Die Share [printers] und [print$] bewirken, dass die am Server angeschlossenen Drucker den Clients zur Verfügung gestellt werden. Das Dollarzeichen in [print$] veranlasst, dass diese Freigabe in der Netzwerkumgebung nicht sichtbar wird. Hier drin befinden sich die Druckertreiber.

In der [netlogon]- Share sind die Loginskripte, die bei der Anmeldung am Server ausgeführt werden. Hier können z. B. Skripte eingestellt werden, die den User bei der Anmeldung direkt mit einer Freigabe verbinden.

Die Share [projekt] soll eine Ordner mit dem gleichen Namen im Netzwerk für alle am Projekt arbeitenden Gruppenmitglieder frei geben.

Die Konfigurationsdatei kann mit einem Editor bearbeitet werden. Ob korrekt editiert wurde, kann mit dem Befehlt „testparm“ überprüft werden.
Den aktuellen Status des Sambaservers kann mit dem Befehl „rcsmb status“ oder „/etc/init.d/ status“ auf der Konsole anzeigt werden:

---

[9] Vgl. Flo84 (2007)

Weitere Befehle sind:

rcsmb start oder /etc/init.d/smb start = starten
rcsmb stop oder /etc/init.d/smb stop = anhalten
rcsmb restart oder /etc/init.d/smb restart = erneuter Start (Stop und Neustart)
rcsmb reload oder /etc/init.d/smb reload = erneuter Start (in einem Schritt)

Der Sambaserver läuft und jetzt können die Benutzer hinzugefügt werden. Damit Windows- Clients auf dem Sambaserver zugreifen können, müssen sie als Linux- und Samba- User angelegt sein.

6. Linuxuser anlegen

Linuxuser werden auf der Konsole mit dem Befehl „useradd –m Username“ angelegt. Durch „–m“ wird automatisch das Heimatverzeichnis mit angelegt. Das verschont das mühselige, nachträgliche Einrichten der Homedirectory mit dazugehöriger Besitzer- und Rechtezuordnung.
Eine Passworteingabe für den neu angelegten Benutzer ist optional.

Ob das Heimatverzeichnis angelegt wurde und welcher Zugehörigkeit und Rechte es besitzt kann mit 2 Befehlen kontrolliert werden:
„ls –lh /home“ listet alle Heimatverzeichnisse unter home auf und zeigt deren Benutzer und Gruppen mit den dazugehörigen Rechten auf das Verzeichnis.
Mit „cat /etc/passwd“ kann man einen Blick in die zentrale Benutzerdatenbank (von Linux!) werfen. Hier werden **alle** Benutzer des Systems aufgelistet. d. h. root ist in der Liste genauso aufgeführt wie die Testszenariobenutzer. Ein Eintrag kann z. B. so aussehen:
anton:x:1001:100:Anton Berger:/home/anton:/bin/bash

Erläuterungen:

anton = Benutzername

x = Das Passwort steht verschlüsselt in der „/etc/shadow“

1001 = Angabe der Benutzer- ID. Sie darf nur einmal vorkommen!

100 = Angabe der Gruppen- ID, der primären Gruppe des Users.

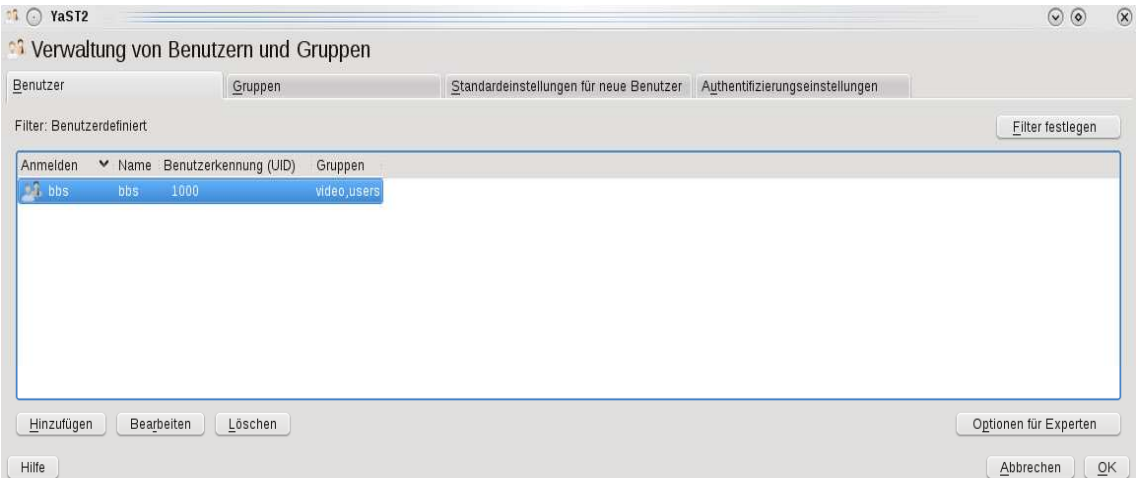
Anton Berger= Vollständiger Name des Benutzers

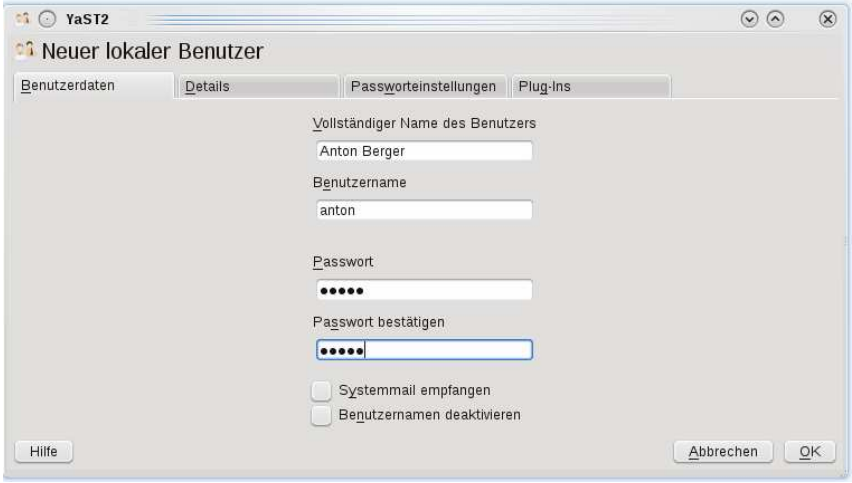
/home/anton = Das Heimat- bzw. Startverzeichnis des Benutzers nach dem Login

/bin/bash = Die Shell, die nach der Anmeldung gestartet werden soll.

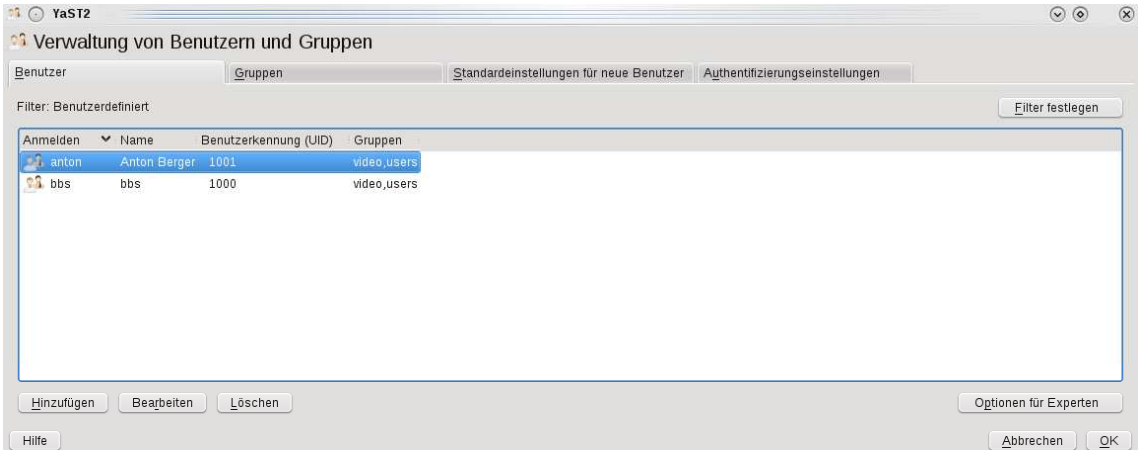
Die Shell ist auch der ausschlaggebende Punkt für diese kleine Exkursion: In einem Netzwerk ist es nicht gewünscht, dass der Benutzer die Möglichkeit hat sich am Server anzumelden. Das darf und soll nur root können. Den Eintrag /bin/bash einfach mit einem Editor auf /bin/false ändern und der Zugriff auf den Server wird verwährt. Eine weitere Option für die Zugriffsverweigerung ist, das Passwort für den Linuxaccount in ein, für den Benutzer, unbekanntes zu ändern oder erst gar keines anzulegen. In OpenSuse 11.2 ist die Anmeldung am System ohne Passwort nicht möglich, es sei denn es wurde in der Benutzer - verwaltung ausdrücklich definiert.

Ein anderes Verfahren Linuxuser ins System hinzuzufügen, funktioniert über das Kontrollzentrum unter Sicherheit und Benutzer im Untermenü Benutzer – und Gruppenverwaltung. Hier können auch Gruppen hinzugefügt werden:





Im Gegensatz zur Konsole muss hier ein Passwort vergeben werden, sonst ist das Speichern nicht möglich.



Es müssen alle Eingaben mit OK bestätigt oder in den Optionen für Experten „Änderung nun schreiben“ aktiviert werden, sonst kann User anton nicht als Sambauser aufgenommen werden. Auch in Yast kann die Anmelde- Shell in /bin/false geändert werden.

7. Sambauser anlegen

Nur Sambauser können sich am Server anmelden. Deshalb muss jeder Linuxuser, der auf den Server zugreifen will, zum Sambauser werden. Sambabenutzer werden auf der Konsole mit dem Befehl „smbpasswd –a Username“ angelegt. Dabei steht „-a“ für add (hinzufügen). Während beim

Anlegen eines LInuxusers, auf der Konsole, die Passwortvergabe optional war, wird beim Hinzufügen eines Sambauser die Eingabe eines Passwortes verlangt und muss bestätigt werden.

```
bbs : bash
Datei  Bearbeiten  Ansicht  Verlauf  Lesezeichen  Einstellungen  Hilfe
linuxlu:/home/bbs # smbpasswd -a anton
New SMB password:
Retype new SMB password:
Added user anton.
linuxlu:/home/bbs #
```

Mit dem Befehl „smbpasswd Username“ ist das Userpasswort veränderbar. Die Sambapasswörter werden in eine separate Datenbank gespeichert. In welcher Datei genau, wird in der „smb.conf“, in der [global]- Share, angezeigt:

```
bbs : bash
Datei  Bearbeiten  Ansicht  Verlauf  Lesezeichen  Einstellungen  Hilfe

linuxlu:/home/bbs # cat /etc/samba/smb.conf
# smb.conf is the main Samba configuration file. You find a full commented
# version at /usr/share/doc/packages/samba/examples/smb.conf.SUSE if the
# samba-doc package is installed.
# Date: 2009-10-27
[global]
        workgroup = SAMBALU
        passdb backend = tdbsam
        printing = cups
```

Die Samba- Datenbank lautet tdbsam (passdb backend = tdbsam). In früheren Distributionsversionen wurde dafür die Datei smbpasswd (passdb backend= smbpasswd) verwendet. Lt. Kofler ist der Vorteil von tdb (trivial database), dass neben dem Login- Namen und dem Passwort noch weitere Daten und Eigen - schaften gespeichert werden. Dazu gehören „SAM (Security Account Manager) – Attribute“, die eine höhere Kompatibilität mit aktuellen Windows- Versionen anbieten.[10]

Der Security Account Manager spielt vor allem in Windowsnetzwerken eine Rolle und wird hier nicht weiter erläutert.

In der Datei /etc/samba/smbusers wird festgelegt, welcher Benutzer unter Linux welchem Benutzer unter Windows entspricht. Solch ein Eintrag sieht wie folgt

[10] Vgl. Kofler (2010) S. 969

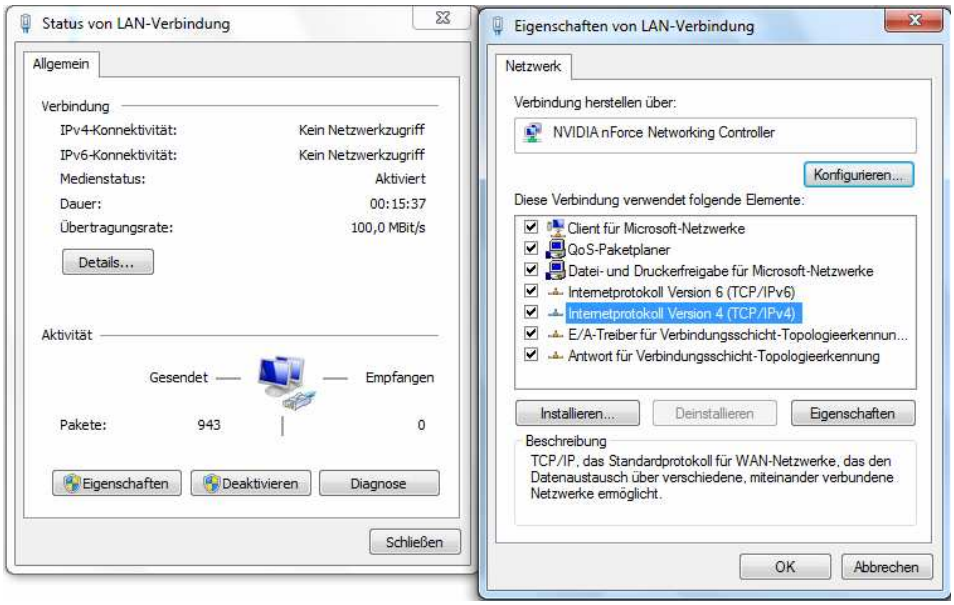
aus: root = administrator. Dieser Eintrag wurde von Suse bei der Einrichtung des Primary Domain Controllers automatisch erstellt. Er wird benötigt, um die Clients der Domäne beitreten zulassen. [11]

Eine Auflistung der Sambauser erfolgt mit „pdbedit -L“. Mit „pdbedit --help“ erhält man noch mehr Optionen, was die auszugebende Liste aufzeigen kann.

8.Windowsuser in Linux / Samba aufnehmen

Damit die unterschiedlichen Betriebssysteme miteinander kommunizieren können, müssen sie im selben Netzwerk liegen. Dafür wird jedem Windows-Client eine eigene, feste IP- Adresse zugeordnet. Es ist nicht vorgesehen, dass sich weitere PC 's ans Netzwerk anmelden, deshalb kann auf eine automatische IP- Adressvergabe verzichtet werden. Natürlich kann Samba auch als DHCP- Server fungieren bzw. konfiguriert werden, aber das ist nicht Bestandteil dieses Projekts.

In der Systemsteuerung die Eigenschaften der LAN- Verbindung öffnen. Dort die Eigenschaften der IP- Konfiguration (TCP/IPv4) anklicken:



[11] Vgl. Klimke (2010) S.34/35

Im Reiter Allgemein (für den jeweiligen Client) folgende Einstellungen vornehmen:

IP- Adresse: 192.168.178.21 (für den WinXP Client „bie“)

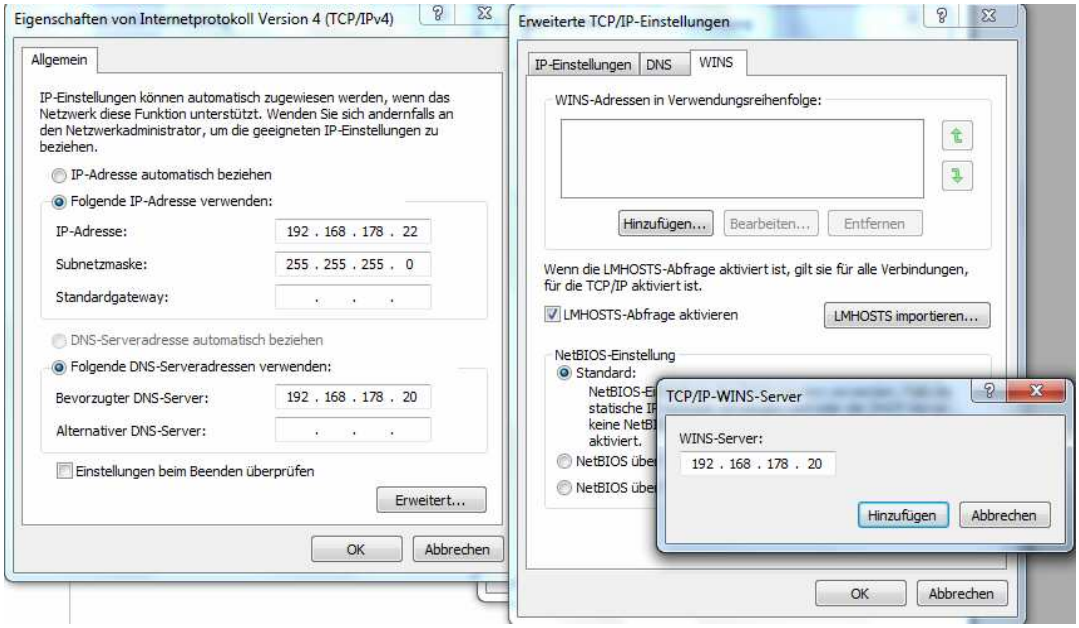
IP- Adresse: 192.168.178.22 (für den Win7 Client „win7“)

Subnetzmaske: 255.255.255.0 (jeweils)

Standardgateway: das Feld kann leer bleiben, da wir ohne Router arbeiten.

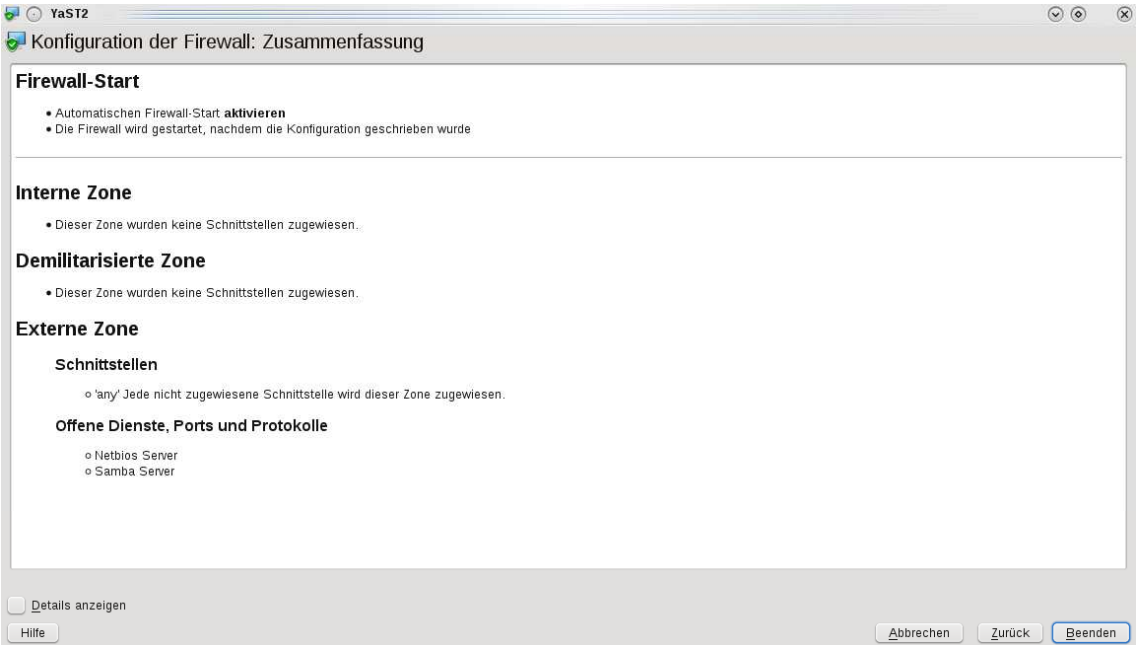
DNS- Server: 192.168.178.20 (entspricht der IP- Adresse des Servers).

In den erweiterten Einstellungen für den WINS- Servers die IP- Adresse 192.168.178.20 hinzufügen. Es ist die gleiche Adresse wie der DNS- Server, denn Samba (bzw. der nmbd – Dämon) ist für die Namensauflösung zuständig. Damit dies funktioniert wurde die Datei /etc/hosts editiert. Siehe dazu auch Abschnitt 5.2 Samba als Domaincontroller einrichten.



Damit weiß jeder Windowsclient wer für die Namensauflösung zuständig ist. Jetzt muss noch dafür gesorgt werden, dass sie auch auf den Server zugreifen können und nicht von der Firewall blockiert werden. Die notwendigen Ein - stellungen werden in Yast im Menüpunkt Sicherheit und Benutzer im Unter - menü Firewall getroffen. Als erlaubter Dienst Netbios Server hinzufügen:

Bevor das Fenster geschlossen wird noch einmal die Übersicht der getroffenen Einstellungen:



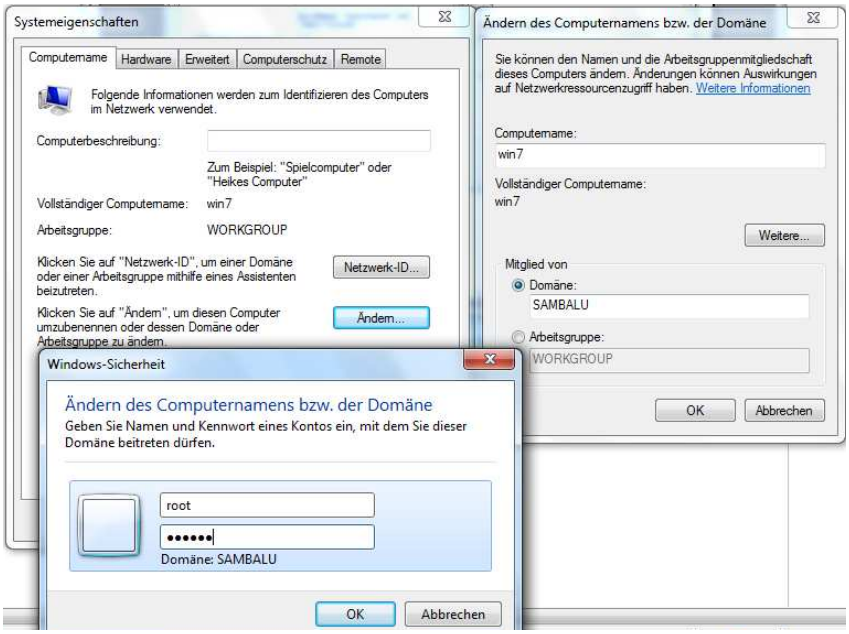
Soweit ist alles für die Aufnahme der Clients in SAMBALU vorbereitet. Doch zuvor sollte noch einmal die IP- Adresse des Sambaservers geprüft werden. Diese muss 192.168.178.20 sein, ansonsten wird sie auf der Konsole mit dem Befehl „ifconfig eth0 192.168.178.20 netmask 255.255.255.0“ eingerichtet und kann mit dem Befehl „ifconfig“ kontrolliert werden. Win7 benötigt für den Beitritt in die Domäne noch einen Patch für die Registry Datenbank in Form einer REG- Datei. Durch einen Doppelklick wird diese ins System eingespielt.

Danach muss Windows neu gestartet werden. Mit dem folgenden Link kann die Datei sofort herunter geladen werden:

https://bugzilla.samba.org/attachment.cgi?id=4988&action=view[12]

Für den Beitritt in die Domäne wird beim Windowsclient im Kontextmenü des Computers (bei XP heißt es Arbeitsplatz) die Eigenschaften aufgerufen. Im Register Computernamen auf „Ändern“ gehen und im neuen Fenster den Domänennamen (SAMBALU) und den jeweiligen Computername (für XP=bie und für WIN7=win7) angeben. Bevor man speichern kann, wird man nach einem Benutzernamen und Kennwort gefragt. Und jetzt kommt der Eintrag in „/etc/samba/smbusers“ ins Spiel. Wie im vorherigen Kapitel (Kap. 7.Sambauser anlegen) bereits erwähnt, wird der User, der in dieser Datei steht, dafür benötigt die Windowsclient der Domäne beitreten zulassen:

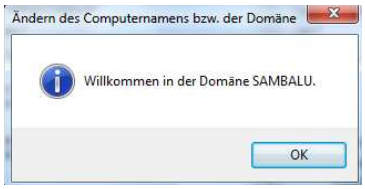
Benutzername = root
Kennwort = das Kennwort, mit dem der PDC erstellt wurde.



So funktioniert es mit dem XP- Client.

[12] Vgl. Klimke (2010) S. 36

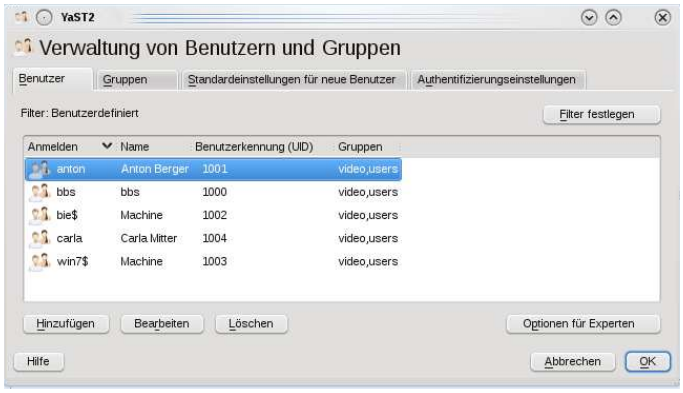
Win7 benötigt dafür root als Sambauser (smbpasswd –a root). Das bei „smbpasswd“ angegeben Passwort sorgt dann zum Beitritt in die Domäne. Ist alles korrekt gelaufen erhält man ein Willkommensfenster als Bestätigung zum Beitritt in die Domäne SAMBALU.



Win7 gibt noch eine Fehlermeldung aus, wegen der DNS- Namensänderung, die aber nicht berücksichtigt werden muss. [13]

Windows muss neu gestartet werden, damit die Anmeldung an der Domäne wirksam wird. Nach dem Neustart muss Strg+ Alt+ Entf. gedrückt werden, um den Anmeldebildschirm zu sehen. Der Dialog zeigt SAMBALU/root und fordert zur Passworteingabe auf. Mit „Benutzer wechseln“ kann sich Anton Berger, der Sambauser ist, mit seinem Sambapasswort, am Server anmelden.

Auf der Serverseite kann man in der Benutzerverwaltung sehen, dass die Windowsclients aufgenommen wurden. Sie sind gekennzeichnet durch ein $ - Symbol, das auf ein Maschinenkonto hinweist.



[13] Vgl. Klimke (2010) S. 36

Mit der Auflistung der Sambauser durch den Befehl „pdbedit –L“ lässt sich erkennen, dass auch hier die Maschinenaccounts hinzugefügt wurden.

9. Verzeichnisse anlegen

Verzeichnisse werden auf der Konsole mit dem Befehl „mkdir Verzeichnisname“ indem Ordner angelegt, indem der Befehl gegeben wurde. Dabei sind die Berechtigungen zum Erstellen eines Ordners zu berücksichtigen. Wichtig ist also welcher Benutzer gerade angemeldet ist und wo sich dieser befindet (in welchem Ordner). Root darf überall beliebig Verzeichnisse anlegen und löschen.
Die Heimatverzeichnisse sind normalerweise (wenn die Zugriffberechtigungen nicht geändert wurden) den dazugehörigen Besitzern zugeordnet, die darauf volle Zugriffsberechtigung haben, d. h. sie können innerhalb dieser neue Ordner erstellen und vorhandene löschen.
Näheres zu Berechtigungen siehe auch Kapitel 10.1 Rechtevergabe serverseitig. Falls angelegte Ordner wieder gelöscht werden sollen, hier noch zwei hilfreiche Befehle:

rmdir Verzeichnisname = Löscht Verzeichnisse ohne Inhalt
rm –r Verzeichnisname = Löscht Verzeichnisse mit Inhalt

Dabei ist zu beachten, dass man sich nicht in dem, zu löschenden, Ordner befinden darf.

Eine weitere Möglichkeit Verzeichnisse zu erstellen besteht auf der grafischen Oberfläche, im Konquerer oder Arbeitsplatz. Auch hier gelten die Berechtigungen wie auf der Konsole.

10. Rechtevergabe

10.1 Rechtevergabe serverseitig (Linuxrechte / ACL)

10.1.1 Linux - Rechte

Standardmäßig gibt es drei Rechte in Linux:

r = read (Leserecht bzw. das Recht den Ordnerinhalt zu sehen)
w = write (Schreibrecht bzw. das Recht Ordner zu erstellen / löschen)
x = execute (Ausführrecht bzw. das Recht in den Ordner zu wechseln).

Diese werden Dateien und Verzeichnissen zugeordnet, nicht dem User. Somit wirken die Berechtigungen auf die Ressourcen, auf die man zugreifen will.
Die drei Rechte werden pro Datei bzw. Verzeichnis dreimal vergeben. Einmal für den dazugehörigen Besitzer, dann für die zugeordnete Gruppe und für „alle Anderen“ (others).
Der dazugehörige Besitzer ist im Normalfall der Erzeuger. Seine primäre Gruppe ist die zugeordnete Gruppe der Datei bzw. des Verzeichnisses. „Alle Anderen“ sind die Restlichen, die weder Besitzer noch Gruppenmitglieder sind.
Die Darstellung der Rechte kann auf unterschiedliche Weise erfolgen:
rwxrwxrwx (das sind neun Zugriffsbits) oder 777. Jede Einzelziffer resultiert aus:

read = 4
write = 2
execute = 1.

Ein vollständiges Zugriffsrecht „rwx“ ergibt 7. Diese Zahl wird auch als Okaler Wert bezeichnet. Der erste Wert wird dem Besitzer zugeordnet, der zweite der Gruppe und der dritte den others. Ein Beispiel dazu:
Der User Anton legt in seinem Heimatverzeichnis /home/anton mit „mkdir“ ein neues Verzeichnis „vz1“ an. Er hat „vz1“ erstellt und ist somit der Besitzer. Er

erhält automatisch die vollen Zugriffsrechte darauf. Die Gruppe und „alle Anderen“ erhalten nur eingeschränkte Leserechte (r-x oder 5). Zusammen ergibt das „rwxr-xr-x“ oder755. Mit dem Befehl „ls -lh“ im Ordner /home/anton wird der Eintrag, über den zuvor erstellten Ordner, aufgelistet:

drwxr-xr-x 2 anton users 4,0K 14. Mai 14:17 vz1

Erläuterung:

d = Angabe des Dateityps, directory (=Ordner / Verzeichnis)
rwx = Zugriffsrechte für den Besitzer
r-x = Zugriffsrechte für die zugehörige Gruppe
r-x = Zugriffsrechte für „alle Anderen“
2 = Anzahl der Unterverzeichnisse
anton = Name des Besitzers
users = Name der Gruppe
4,0K = Größe des Verzeichnisses
14. Mai= Erstellungstag
14:17 = Erstellungsuhrzeit
vz1 = Name des Verzeichnisses

Die Auflistung einer Datei sieht nur unwesentlich anders aus:

-rw-r--r-- 2 anton users 4,0K 14. Mai 14:19 datei1.txt

Das ersten Zeichen (-) weist auf eine Datei hin. „datei1.txt“ ist eine Textdatei, deshalb ist das erste „x“ deaktiviert. Hier ist entscheidend, ob es Lese- oder Schreibrechte gibt.

Standardmäßig vergibt Suse an neu erstellte Textdateien die Zugriffsrechte 666 (rw-rw-rw-), d. h. jeder kann lesen und schreiben. An neue Verzeichnisse die Rechte 777 (rwxrwxrwx), d. h. jeder hat volle Berechtigungen. Da diese Bedingungen (jeder darf jede Datei ändern, löschen etc.) unrealistisch sind, gibt es die umask- Einstellung. Dieser Eintrag enthält einen Zahlenwert, der von dem Oktalwert der Zugriffsbits abgezogen wird. Mit „umask“ kann der eingestellte Wert angezeigt werden: „0022“(die erste 0 bezieht sich auf die Spezialbists, siehe Kapitel 10.1.2). Im Klartext bedeutet das:

Rechte für eine Datei = 666	Rechte für ein Verzeichnis = 777
-022	-022
= 644	= 755

Somit ist zu erklären, warum eine neue Datei automatisch „rw-r--r--“ und ein neues Verzeichnis „rwxr-xr-x“ erhält. „umask“ gibt die Möglichkeit mehreren Usern auf einmal „Rechte zu geben“ bzw. ihnen eine Maske überzustülpen.

Natürlich lassen sich diese Zugriffsberechtigungen spezifisch an das jeweilige Netzwerk anpassen. Dafür sind am besten Administratorrechte (root- Rechte) geeignet. Der Besitzer darf normalerweise nur innerhalb seines Heimat - verzeichnisses an und in seinen Dateien und Verzeichnissen Änderungen vornehmen. Z. B. die Gruppenzugehörigkeit neu definieren oder die Rechte der Gruppe bzw. others ändern.
Mit den folgenden Befehlen können die Zugehörigkeiten und Rechte geändert werden, dabei kann äquivalent zum Verzeichnisnamen (VZ) der Dateiname verwendet werden:

chown Besitzername VZ	= Ändern des Besitzers
chgrp Gruppenname VZ	= Ändern der Gruppe
groupadd Gruppenname	= Anlegen einer neuen Gruppe
usermod –g Gruppenname	= Ändern der primären Gruppe des Users
usermod –G Gruppenname	= Ändern der anderen Gruppe(n) des Users
chmod 755 VZ	= Ändern der Zugriffrechte dabei stellt 755 nur ein Bsp. dar und ist dem Bedarf anzupassen
chmod u (g, o, a) +/-r (w, x) VZ	= Alternative Schreibweise zu „chmod 775 VZ“

Sollen sich diese geänderten Rechte auch auf vorhandene Unterverzeichnisse und Dateien auswirken, dann muss auf „chmod“ ein „-R“ folgen (chmod –R 775 VZ, dabei steht „–R“ für rekursives ändern).
Beim Ändern von Rechten ist es entscheidend in welchem Ordner man sich gerade befindet. Eine Fehlermeldung von Suse: „chmod: Zugriff auf „vz1“ nicht möglich: Datei oder Verzeichnis nicht gefunden“, deutet darauf hin, dass man

sich nicht in dem Ordner befindet, der das genannte Verzeichnis enthält. Mit der Angabe des absoluten Pfades des Verzeichnisses wird das Problem behoben.

Das Erstellen von Gruppen dient dazu, Gruppenmitgliedern den Zugriff auf gemeinsam genutzte Dateien, Verzeichnisse und Programme zu geben.
Es ist daher sinnvoll, Usern mit gleichen Aufgabenstellungen, in eine Gruppe zusammen zufassen, um ihnen gleiche Rechte zu geben.

10.1.2 Spezialbits (Setuid- Bit, Setgid- Bit, Sticky- Bit)

Außer den drei bisher genannten Zugriffsberechtigungen können noch drei weitere, spezielle Zugriffsbits (erweiterte Rechte) vergeben werden. Diese sind Setuid- Bit, Setgit- Bit und Sticky- Bit.

Setuid- Bit

Setuid- Bit, auch SUID- Bit genannt, bewirkt, dass das Programm mit den Rechten des Besitzers der Datei ausgeführt wird. Normalerweise würde es mit den Rechten des Benutzers ausgeführt werden. Damit Suse dieses Zusatzrecht auswertet, muss das Ausführrecht (x) gesetzt sein, sonst bleibt es wirkungslos. Kofler erklärt in seinem Buch, dass das Bit gesetzt wird, um zusätzliche Rechte, die nur bei der Ausführung des Programms wirken, an gewöhnliche Besitzer weiter zugeben. Er sieht es als ein mögliches Sicherheitsrisiko an, da viele Pro - gramme root gehören und diese durch das Zugriffsbit wie unter root ausgeführt werden. Deshalb sollte die Anwendung des Setuid- Bits vermieden werden.[14]
Kommandos für das SUID- Bit:

chmod u+s VZ = zum Hinzufügen des Rechts
chmod u-s VZ = zum Entfernen des Rechts.

Der Oktalwert des Bits ist 4000. Alternativ kann der Befehl auch so lauten: chmod 4755 VZ.

[14] Vgl. Kofler (2010) S. 346

Ist das Setuid- Bit gesetzt sieht die Auflistung mit „ls –lh“ folgendermaßen aus:

drw**s**r-xr-x 2 anton users 4,0K 14. Mai 14:17 vz1

Bei den Besitzerrechten steht anstelle von „x“ ein „s“. Wird das Setuid- Bit ohne das x- Recht vergeben, wird anstelle von „s“ ein „S“ dargestellt, um darauf hinzuweisen, dass das Ausführrecht fehlt.

Setgid- Bit

Setgid- Bit ist ähnlich wie Setuid- Bit. Das Programm wird mit den Rechten der Gruppe ausgeführt, anstatt mit denen des Benutzers. Auch hier muss das Aus - führrecht (x) gesetzt sein, damit das Recht beachtet wird.
Das Setgid- Bit bewirkt in Verzeichnissen, dass dort neu erzeugte Dateien der Gruppe des Verzeichnisses zugeordnet werden und nicht wie üblich der Primärgruppe des Erzeugers. Das ist ein großer Vorteil für Teamarbeit. Die, durch die Gruppe, verbundenen Mitglieder haben so Zugriff auf die neu erstellten Dateien.

Das Kommando zum Setzen bzw. Entfernen des Setgid- Bit ist ähnlich wie das zuvor, nur anstelle des „u“ für User, wird ein „g“ für group verwendet:

chmod g+/- s VZ.

Der Oktalwert des Bits ist 2000. Alternativ kann der Befehl auch so lauten: chmod 2755 VZ.

Mit „ls –lh“ aufgerufen, wird der Eintrag so aufgelistet:

drwxr-**s**r-x 2 anton users 4,0K 14. Mai 14:17 vz1

Bei den Gruppenrechten steht anstelle von „x“ ein „s“. Auch hier gilt: wird das Setgid- Bit ohne das x- Recht vergeben, wird anstelle von „s“ ein „S“ dargestellt.

Sticky- Bit

Das Schreibrecht an einer Datei oder Verzeichnis bedeutet, dass diese verändert werden dürfen. Neue Ordner können erstellt und sogar gelöscht werden. Dies ist aber nicht immer gewünscht. Abhilfe schafft das Sticky- Bit.

Es bewirkt, dass nur der Besitzer seine Dateien löschen darf. Auch das Sticky-Bit benötigt das x- Recht, um Wirkung zu haben. Das Kommando dafür ist:

chmod +t VZ = zum Hinzufügen des Rechts
chmod -t VZ = zum Entfernen des Rechts

Der Oktalwert des Bits ist 1000. Alternativ kann der Befehl auch so lauten: „chmod 1755 VZ“.

Ist das Sticky- Bit gesetzt sieht die Auflistung mit „ls –lh“ folgendermaßen aus:

drwsr-xr-**t** 2 anton users 4,0K 14. Mai 14:17 vz1

Bei den Zugriffsrechten für „alle Anderen“ steht anstelle von „x“ ein „t“.
Wird das Sticky- Bit ohne das x- Recht vergeben, wird anstelle von „t“ ein „T“ dargestellt, um darauf hinzuweisen, dass das Ausführrecht fehlt.

10.1.3 ACL (Access Control Lists)

ACL steht für Access Control Lists. Man braucht sie, um die Windows-Rechtevergabe besser abbilden zu können, denn mit den ACLs können weitaus mehr Rechte vergeben werden und diese Möglichkeit wird von Samba genutzt:

> „ACLs ermöglichen es, für jede Datei bzw. für jedes Verzeichnis beliebig viele Regeln aufzustellen, welche Benutzer und Gruppen die Datei bzw. das Verzeichnis lesen oder verändern dürfen und wer das – abweichend von den Unix- Zugriffsrechten – *nicht* darf. ACLs wirken also ergänzend zu den Standardzugriffsrechten und können zusätzliche Rechte einräumen oder vorhandene Rechte entziehen.“[15]

Mit dem Kommando „getfacl vz1“ bekomme ich eine übersichtliche Auflistung der, für das Verzeichnis „vz1“, geltenden Zugriffsrechte. Es stellt im Wesentlichen die gleichen Inhalte dar, wie beim „ls –lh“ Befehl: der Name vom Verzeichnis, Besitzer und Gruppe mit den jeweilig gültigen Rechte:

[15] Kofler (2010) S. 349

file: vz1
owner: anton
group: users
user::rwx
group::r-x
other::r-x

Durch ACL besteht die Möglichkeit Dateien und Verzeichnissen weitere Besitzer und Gruppen zuzuordnen. Mit „setfacl –m“ werden ACL- Rechte vergeben. Ein Beispiel: Carla und die Gruppe „fsinf“ sollen volle Zugriffsrechte auf das Verzeichnis „vz1“ erhalten:

setfacl –m u:carla:rwx,g:fsinf:rwx (absoluter Pfad)vz1

Erläuterung:

setfacl	= Befehl zum Verändern der zusätzlichen Rechte
-m	= modify, verändert ACL- Rechte an Dateien/ Verzeichnissen
u:carla:rwx	= u= user**:** Name des User**:** Definition der Rechte
g:fsinf:rws	= g= group**:** Name der Gruppe**:** Definition der Rechte
vz1	= Name des Verzeichnisses (bzw. Datei, mit evtl. Pfadangabe)

Das ein ACL- Recht gesetzt wurde sieht man in der Auflistung mit „ls –lh vz1“ anhand eines zusätzlichen „+“:

drwxrwxr-x**+** 2 anton users 4,0K 15. Mai 14:17 vz1

Die Ausgabe des Befehls „getfacl vz1“ zeigt detaillierter auf, wie die Vergabe der Zugriffsrechte von Suse umgesetzt wurde:

file: vz1
owner: anton
group: users
user::rwx
user:carla:rwx
group::r-x
group:fsinf:rwx
mask::rwx

other::r-x

Erläuterung der unteren Zeilen:

user::rwx	= Rechte des zugehörigen Besitzers (owner)
user:carla:rwx	= Rechte des zusätzlichen Besitzers (named user), mehrere named user möglich
group::r-x	= Rechte der zugehörigen Gruppe (owning group)
group:fsinf:rwx	= Rechte der zusätzlichen Gruppe (named group), mehrere „named group“ möglich
mask::rwx	= Maske**,** Werte, mit denen maskiert wird
other::r-x	= Rechte für alle Anderen

Die vergebenen Berechtigungen werden anhand einer Maskierung (mask) ausgewertet, welche bei der Vergabe von ACLs automatisch erstellt wird. Diese Maskierung betrifft nicht die Rechte von „owner“ und „others“, deren Rechte sind immer wirksam. Wie diese Maskierung funktioniert, wird am besten an einem Beispiel deutlich:

named user	carla	rwx
mask		rwx
wirksame Rechte = rwx		

Hätte Carla nicht die vollen Rechte, sondern nur Leserechte (r-x) erhalten und die Maskierung würde „rw-“ vorsehen, dann wäre die Rechnung wie folgt:

named user	carla	r-x
mask		rw-
wirksame Rechte = r--		

Nur das Leserecht käme zur Wirkung. D. h. der jeweilige Wert in dem Eintrag (named user/group, owning group) muss gleich sein mit dem Wert in mask, damit die gesetzten ACLs wirken. Masken werden benutzt, um filtern zu können und damit Grenzen festzulegen.
Die Maske lässt sich mit „setfacl –m m:rwx VZ“ oder „chmod“ ändern und wird dann neu berechnet, außer es ist die Option „-n“ gesetzt, die dies unterbindet. Wie diese Neuberechnung aussieht, lässt sich gut am nächsten Beispiel

erkennen: Gesetzte ACL- Berechtigungen werden durch „chmod g-w vz1“ eingeschränkt und mit dem Kommando „getfacl vz1“ ausgegeben:

```
# file: vz1
# owner: anton
# group: users
user::rwx
user:carla:rwx          #effective:r-x
group::r-x
group:fsinf:rwx         #effective:r-x
mask::r-x
other::r-x
```

Der „ls –lh“ Eintrag sieht wie folgt aus:

drwxr-xr-x**+** 2 anton users 4,0K 15. Mai 14:17 vz1

Dadurch, dass der mask- Eintrag nur noch „r-x“ beinhaltet, sind auch die Rechte von named user und named group auf „r-x“ geschrumpft. Durch den Kommentareintrag „#effective :r-x“ klar zu erkennen.

Andreas Grünbacher unterscheidet in seiner Dokumentation zwischen den Zugriffsrechten (Access ACL) für Benutzer und Gruppen auf Dateien und Verzeichnisse und den Default ACL (Vorgabe- ACL), die nur auf Verzeichnisse angewandt werden können und dabei festlegen, welche ACL Rechte auf neu erzeugte Dateien und Verzeichnisse vom übergeordneten Verzeichnis vererbt werden.[16]

Dafür muss zu dem Befehl „setfacl “ ein „-d“ hinzukommen, „-d“ für default (Standard). Mit „setfacl –d –m g:fsinf:rwx vz1“ wird der Gruppe „fsinf“ in neu erzeugten Dateien und Verzeichnissen die vollen Zugriffsrechte zugewiesen. Dabei werden diese Rechte in neu erzeugten Verzeichnissen auch wieder als Standard gesetzt und weitervererbt.

---

[16] Vgl. Grünbacher (2003) S. 559

Mit „ls –lh“ ist die Veränderung nicht zu sehen. Erst der Aufruf von „getfacl vz1“ macht den Unterschied deutlich:

```
# file: vz1
# owner: anton
# group: users
user::rwx
user:carla:rwx          #effective:r-x
group::r-x
group:fsinf:rwx         #effective:r-x
mask::r-x
other::r-x
default:user::rwx
default:group::r-x
default:group:fsinf:rwx
default:mask::rwx
default:others::r-x
```

Die Auflistung zeigt die Zugriffsrechte und die „default- Rechte“. An den ersten neun Einträgen hat sich nichts geändert. Die Gruppe „fsinf“ hat in „vz1“ nach wie vor nur Lese- und Ausführrecht (r-x). Es fällt auf, dass ein „default mask“-Eintrag erstellt wurde. Dieser gilt für die „default“- Berechnung.
Die fünf Defaulteinträge werden in neu erzeugten Dateien und Verzeichnisse dafür sorgen, dass „fsinf“ volle Zugriffsrechte zugewiesen bekommen. Hierbei ist zu beachten, dass der Unterschied zwischen Dateien und Verzeichnissen darin liegt, dass Verzeichnisse diese Rechte weitervererben können, Dateien dies aber nicht tun.

Verdeutlichung an einem Beispiel: Erstellen eines neuen Verzeichnisses mit „mkdir vz1/neu1“ und dessen Einträge mit „getfacl vz1/neu1“aufzeigen:

```
# file: vz1/neu1
# owner: anton
# group: users
user::rwx
```

```
group::r-x
group:fsinf:rwx
mask::rwx
other::r-x
default:user::rwx
default:group::r-x
default:group:fsinf:rwx
default:mask::rwx
default:others::r-x
```

Vergleicht man diese Auflistung mit der von „vz1“ fällt auf, dass die Defaultrechte komplett übernommen wurden. Nur sie wurden bei der Vererbung berücksichtigt und haben Auswirkungen auf die Zugriffsrechte von „neu1“. Die zugehörige Gruppe „users“ hat nur „r-x“ (Linuxstandard), wohingegen die Gruppe „fsinf“ volle Rechte hat, da sich auch der mask- Eintrag geändert hat und dementsprechend neu berechnet wurde.
Erstellt man eine neue Testdatei sind die vererbten Rechte ähnlich denen des Verzeichnisse. Die Ausgabe einer mit „touch vz1/datei1“ erzeugt Testdatei mit „getfacl vz1/datei1“ sieht wie folgt aus:

```
# file: vz1/datei1
# owner: anton
# group: users
user::rw-
group::r-x                    #effective:r--
group:fsinf:rwx               #effective:rw-
mask::rw-
other::r--
```

Bei einer Textdatei vergibt Suse die Rechte 666. Es fehlt das x- Recht, weshalb der mask- Eintrag auch nur „rw-“ beinhaltet. Alle weiteren Rechte für die Datei ergeben sich dann aus der Berechnung.

Noch mehr Informationen über ACLs bietet die PDF- Datei von Andreas Grünbacher unter: http://www.suse.de/~agruen/acl/chapter/fs_acl-de.pdf.

Noch etwas Grundsätzliches:
Die ACLs müssen beim Rechnerstart mit gestartet werden, damit diese zur Verfügung stehen. Dies kann in der Konfigurationsdatei „etc/fstab“ oder mit dem Befehl „mount“ kontrolliert werden. Der Eintrag mit dem Kommando „mount“ sieht dann so oder so ähnlich aus:

/dev/sda6 on / type ext4 (rw,acl,user_xattr)

Die hier, für ACLs, beschriebenen Befehle und Ausgaben wurden alle auf der Konsole ausgeführt. Natürlich kann man auf der grafischen Oberfläche in den Eigenschaften der Ordner, unter Berechtigungen, die Benutzer (nur root!)– und Gruppenzugehörigkeit mit ihren Zugriffsrechten und in den Erweiterten Berechtigungen auch die erweiterten Rechte ändern.

10.2 Rechtevergabe clientseitig (über SMB)

Es besteht die Möglichkeit über Gruppenzugehörigkeit dem Windowsclient eine Gruppe zuzuordnen, die bestimmte Rechte hat. Z. B. eine Gruppe mit Administratorenrechten. Dieser Gruppe wird im Gegenzug eine Gruppe auf dem Sambaserver zugeordnet. Dafür muss Samba gesagt bekommen, wie die jeweilige Gruppe heißt und wer sich auf welchem System befindet. So wie in der Datei „/etc/samba/smbusers“ festgelegt wird, welcher Benutzer unter Linux welchem Benutzer unter Windows gleichkommt, so kann dies auch mit Gruppen geschehen. Prinzipiell kann jede Linuxgruppe mit jeder Windowsgruppe ver - bunden werden. Dafür muss die benötigte Gruppe(n) mit „groupadd Gruppen - name“ erstellt werden, bzw. falls es auf der Windowsseite schon eine Gruppe gibt, muss nur noch das Gegenstück geschaffen werden. Mit „useradd – m“ die Benutzer anlegen und sie zu Gruppemitgliedern machen (usermod –g/-G). Wenn z. B. der Windowsgruppe A1 eine Linuxgruppe B1 zugeordnet werden soll, dann lautet der Befehlt dafür:

net groupmap add ntgroup=A1 unixgroup=B1 type=d

Somit entspricht die Gruppe A1 der Gruppe B1 auf dem Server. Der Eintrag lässt sich mit dem Befehl „net groupmap list“ ausgeben:

A1 (S-1-5-21-2739396525 – 2695154064 – 2931498878 – 1000) -> B1

Am Ende der Zahlenkette steht ein Wert (1000). Es ist ein RID- Wert, der bei der Zusammenführung mitgegeben wurde. Über diesen Wert können Rechte gesteuert werden. Es gibt eine Liste mit vordefinierten bekannten Einheiten und den zugehörigen RIDs:

Well-Known User Default RIDs / Bekannte Standardbenutzer-RIDs

Well-Known Entity	Bekannte Einheit	RID	Type	Notwendig
Domain Administrator	Domänen-Administrator	500	User	Nein
Domain Guest	Domänen-Gast	501	User	Nein
Domain KRBTGT	Domäne KRBTGT	502	User	Nein
Domain Admins	Domänen-Administratoren	512	Group	Ja
Domain Users	Domänenbenutzer	513	Group	Ja
Domain Guests	Domänengäste	514	Group	Ja
Domain Computers	Domänencomputer	515	Group	Nein
Domain Controllers	Domänencontroller	516	Group	Nein
Domain Certificate Admins	Domänen-Certificate-Administratoren	517	Group	Nein
Domain Schema Admins	Domänen-Schema-Administratoren	518	Group	Nein
Domain Enterprise Admins	Domänen-Enterprise-Administratoren	519	Group	Nein
Domain Policy Admins	Domänen-Policy-Administratoren	520	Group	Nein
Builtin Admins	Builtin-Administratoren	544	Alias	Nein

Well-Known Entity	Bekannte Einheit	RID	Type	Notwendig
Builtin users	Builtin-Benutzer	545	Alias	Nein
Builtin Guests	Builtin-Gäste	546	Alias	Nein
Builtin Power Users	Builtin-Hauptbenutzer	547	Alias	Nein
Builtin Account Operators	Builtin-Zugriffsoperator	548	Alias	Nein
Builtin System Operators	Builtin-System-Operator	549	Alias	Nein
Builtin Print Operators	Builtin-Drucker-Operator	550	Alias	Nein
Builtin Backup Operators	Builtin-Backup-Operator	551	Alias	Nein
Builtin Replicator	Builtin-Replikator	552	Alias	Nein
Builtin RAS Servers	Builtin-RAS-Server	553	Alias	Nein

Das heißt, wenn man möchte, dass eine Gruppe volle Zugriffsberechtigungen erhält, dann wird zu der Gruppenverbindung der RID - Wert = 512 mitgegeben.

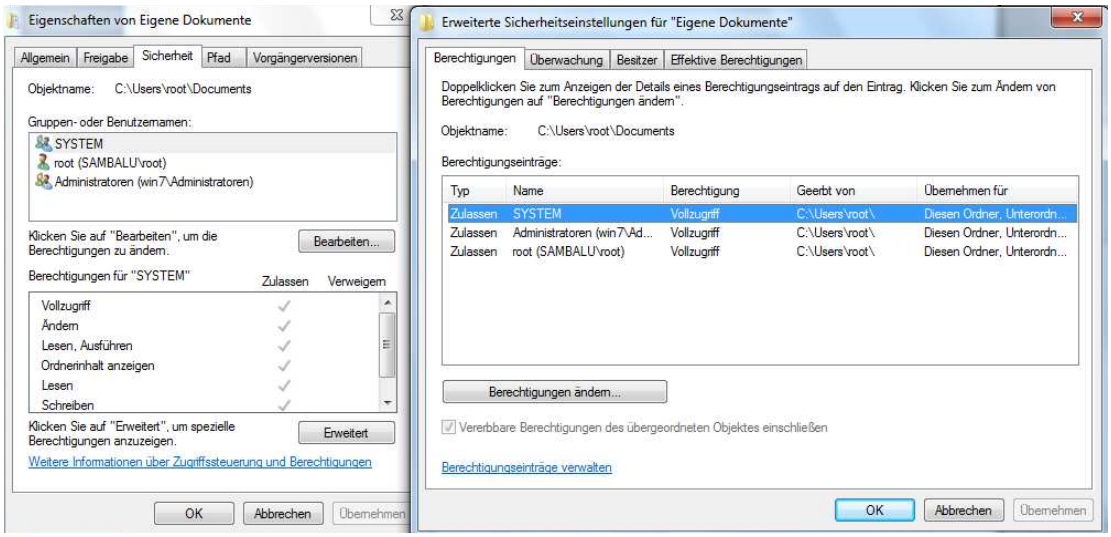
net groupmap add rid=512 ntgroup="Domain Admins" unixgroup=fsinf type=d

Mit dem Auflistungsbefehl „net groupmap list" sieht der Inhalt dann so aus:

A1 (S-1-5-21-2739396525 – 2695154064 – 2931498878 – 1000) -> B1
Domain Admins (S-1-5-21-2739396525 – 2695154064 – 2931498878 – 512) -> B1

Somit haben alle Mitglieder der Gruppe „fsinf" durch „rid=512" die vollen Zugriffsrechte erhalten und können sich am Client anmelden für Verwaltungs - tätigkeiten.

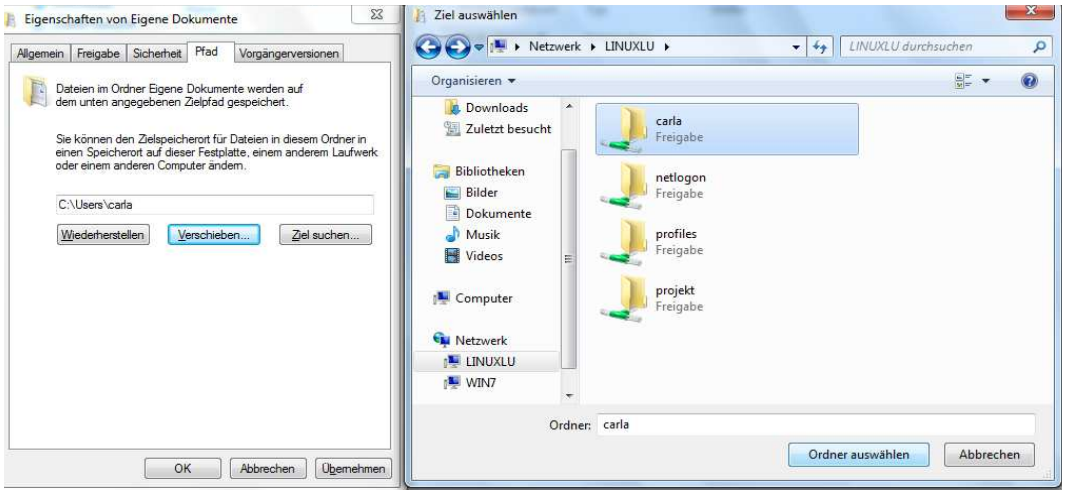
Windows hat eine Vielzahl von Rechten und Einstellmöglichkeiten. Sie werden im Kontext des Verzeichnisses aufgerufen:



Unter dem Reiter Effektive Berechtigungen sind noch weitere Einstellungen möglich, auf die ich hier aber nicht eingehe.

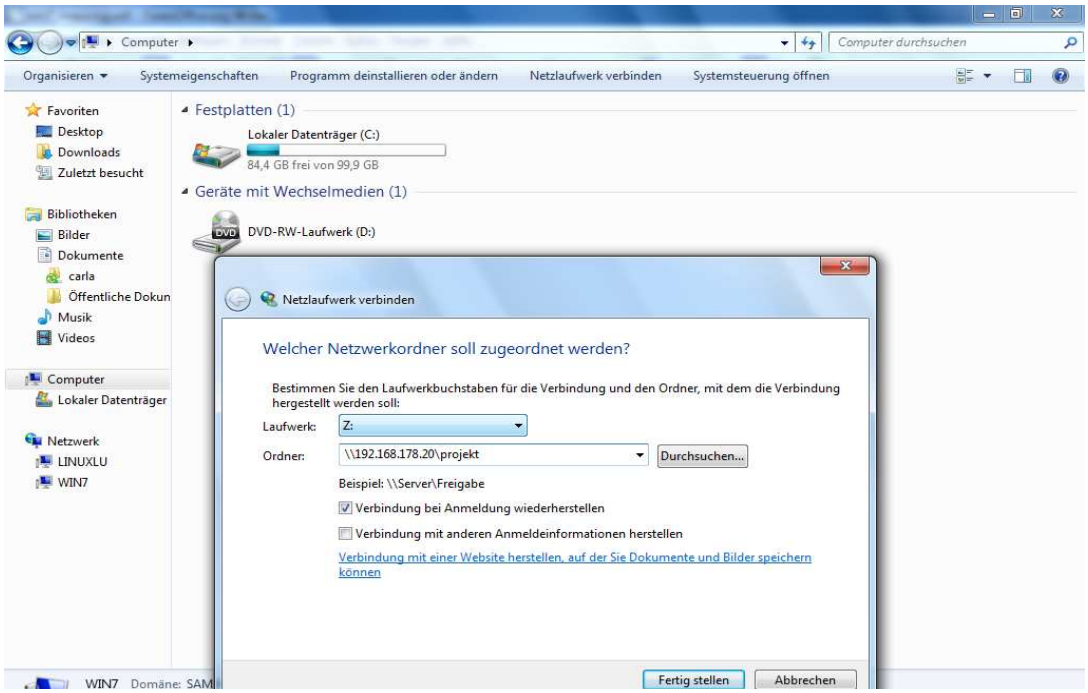
11. Laufwerksmapping

Das Homedirectory eines jeden Benutzers liegt in der Share [home]. Es wird vom Server automatisch gemappt und stellt es dem User zur Verfügung. Die neu erzeugten Daten gelangen wie in der Share [profiles] unter „path = %H“ angegeben in das Heimatverzeichnis der Users auf dem Server. Um auch die Windows- Daten aus dem Ordner „Eigene Dokumente“ (Win7) bzw. „Eigene Dateien“ (XP) auf dem Server bereitzustellen, wird in den Eigenschaften des Ordners „Eigene Dokumente“/ „Eigene Dateien“ der Pfad zum Server eingetragen. Somit werden die Daten vom lokalen Rechner auf den Server verschoben.

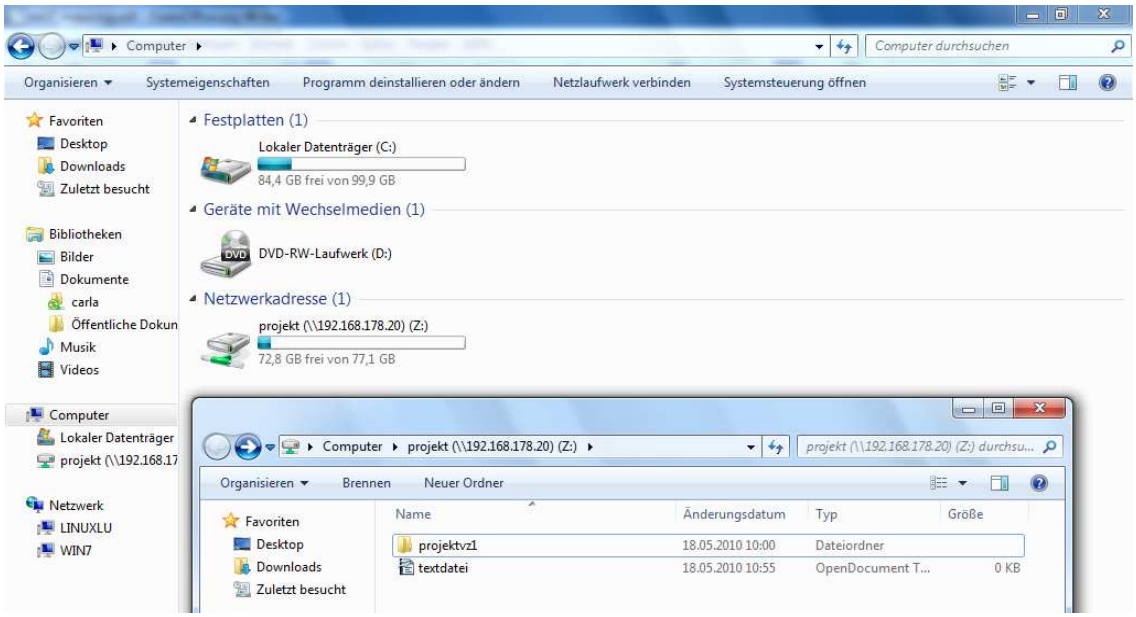


Windows fragt noch einmal nach: „Möchten Sie alle Dateien vom alten Ort an den ausgewählten neuen Ort verschieben?“ Ein Klick auf „Ja“ verschiebt die Daten. Nach erfolgreichem Transferieren, kann Carla beliebig in ihrer Arbeitsumgebung Verzeichnisse und Dateien erstellen und gleichzeitig kann dieses Vorgehen auf dem Server verfolgt werden.

Beim Erstellen des Primary Domain Controllers wurde eine neue Freigabe [projekt] erzeugt. Um einem User Zugriff darauf zugeben, muss diese Share gemappt werden. Dies geschieht von der Clientseite aus unter Computer, Netzlaufwerk verbinden. Im neuen Fenster wählt man einen freien Laufwerksbuchstaben aus und gibt den Pfad zum Verzeichnis an:



Mit „Fertig stellen“ ist man mit dem Laufwerk verbunden. Dabei gelangt man direkt in den Ordner „projekt“, indem zuvor serverseitig ein Unterverzeichnis (projektvz1) und eine Textdatei (textdatei) erstellt wurden. In der Share [home] steht „read only = no“. Das bedeutet, dass man nicht nur lesend auf das Verzeichnis zugreifen kann. Da diese beiden Dateien aber von root erstellt wurde (root = Besitzer+ Gruppe) und andere (group + others) nur Leserechte haben, gelingt es dem User nicht diese Dateien zu verändern. Dafür müssen die Zugriffsrechte angepasst werden mittels den Linux- oder ACL- Rechten.



Das Laufwerksmapping beim XP- Client sieht ähnlich aus und wird unter Arbeitsplatz, Extras, Netzlaufwerk verbinden aufgerufen.

Es besteht auch die Handhabe ein Laufwerksmapping, über ein Skript, zu automatisieren. Dazu mehr im nächsten Abschnitt, Kapitel 12. Skripte.

12. Skripte

Es gibt die Möglichkeit eine Laufwerksverbindung über ein Skript automatisch ablaufen zu lassen. Bei der Anmeldung wird dann, über das Skript, die Laufwerkszuordnung mit ausgeführt. Diese Skripte liegen in der Share [netlogon]. Damit sie mit ausgewertet werden, benötigt die [global]- Share der „smb.conf“ einen zusätzlichen Eintrag: „logon script = Name der Batch- Datei“. Dieser Zusatz wird mit einem Editor (z. B. joe) einfach eingefügt. Danach muss der Server neu gestartet werden (rcsmb restart), damit die Veränderung greift. Alternativ dazu, kann in Yast im Sambaserver im Reiter „Identität“, erweiterte Einstellungen, Globale Einstellungen für Experten, der Eintrag „logon script“, als Option, mit dem Namen der Datei, als Wert, hinzugefügt werden. Durch Bestätigung der geänderten Einstellungen werden diese umgesetzt.

Ist solch ein Eintrag in der [global]- Share gesetzt, wird im Verzeichnis path der [netlogon]- Sektion eine derartige Batch- Datei auch erwartet. Sie muss in das Verzeichnis „/var/lib/samba/netlogon“ gelegt werden. Dort wird der Client, bei der Anmeldung, es suchen, um es dann ausführen zu können.

Beim Entwickeln einer Batch- Datei muss darauf geachtet werden, dass jede Zeile mit einem Wagenrücklauf/ Zeilenvorschub (CR+LF) beendet wird, damit die Datei auch für den Client lesbar ist. Dafür benutzt man am besten in Windows einen Editor und achtet beim Speichern darauf, eine „.bat“- Datei zu erstellen.
In Linux enden die Zeilen normalerweise nur mit einem Wagenrücklauf. Der Editor „vi“ kann mit dem Befehl „set ff=dos“ dafür sorgen, dass die Zeilen mit

CR+LF beendet werden sollen. Dann kann mit dem Kommando „vi logonanton.bat“ eine Batch- Datei geschrieben werden, mit dem Inhalt:

```
@echo off
net use Z: \\samba\projekt
```

Mit den Zeichen „Esc +**:**wq“ wird der Insert- Modus von „vi“ beendet und die Datei gespeichert. Nach dem Ablegen der Datei in das benötigte Verzeichnis wird der User Anton bei der Anmeldung am Server direkt mit dem Ordner projekt verbunden.
In diese Skripte können verschiedene Anweisungen geschrieben werden, die dann beim Start mit ausgeführt werden. Es ist auch denkbar, eine Begrüßung mit aufzunehmen.

13. Roaming Profiles

Roaming Profiles bedeutet, dass die für den Benutzer spezifischen Einstellungen auf dem Server gespeichert werden und bei der Anmeldung, am Server, von dort geladen werden. Der Vorteil davon ist, dass der Benutzer sich an jedem Rechner, der am Server angeschlossen ist, anmelden kann. Wären seine Daten lokal auf einem Rechner gespeichert (Local Profiles), dann könnte sich der User nur an diesem einem Computer anmelden. Der Verwaltung-saufwand wäre enorm, da jeder User auf seinem Rechner verwaltet werden müsste. Da der Server die ganzen Daten hat muss nur noch zentral an einem Ort der Aufwand betrieben werden. Beim Anmelden bekommt der Benutzer dann automatisch, die für ihn zugeschnittene Arbeitsumgebung. D. h. der Desktophintergrund mit Icons ist immer gleich, seine Lesezeichen, seine E-Mails usw. alles was speziell für diesen Benutzer angelegt wurde, wird bei der Anmeldung am Server geladen. Samba stellt diese Voraussetzungen in der Konfigurationsdatei „smb.conf“:

```
[global]
        workgroup = SAMBALU
        passdb backend = tdbsam
        printing = cups
        printcap name = cups
        printcap cache time = 750
        cups options = raw
        map to guest = Bad User
        include = /etc/samba/dhcp.conf
        logon path = \\%L\profiles\.msprofile
        logon home = \\%L\%U\.9xprofile
        logon drive = P:
        usershare allow guests = Yes
        add machine script = /usr/sbin/useradd  -c Machine -d /var/lib/nobody -s /bin/false %m$
        domain logons = Yes
```

„logon path=\\%L\profiles\.msprofile“ gibt den Ort an, wo die Benutzerprofile liegen.

„logon home=\\%L\%U\.9xprofile“ ist die Angabe des Standortes des Heimatverzeichnisses.

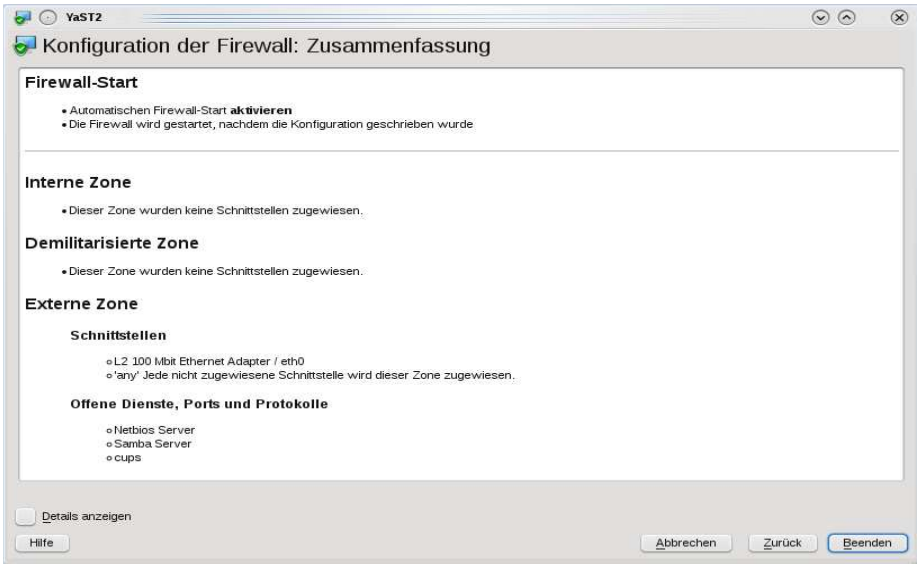
„logon drive=P:“ sagt, wie das Laufwerk heißt, auf welches das Heimat - verzeichnis gemountet wird.

„domain logons=Yes“ ermöglicht die Anmeldung der Clients am Server.

14. Drucker anbinden

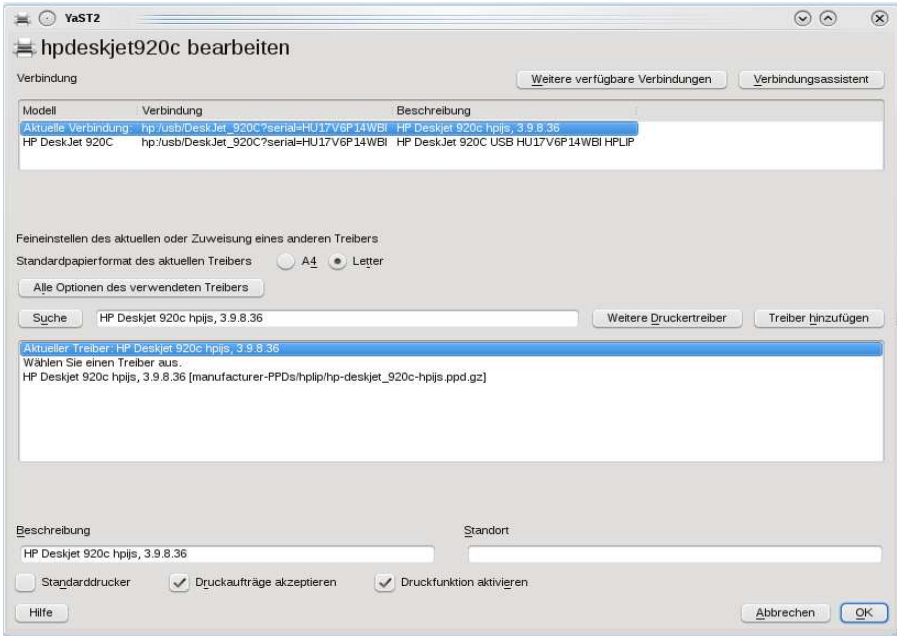
Damit die Druckereinbindung funktioniert muss in der Firewall der entsprechende Dienst (CUPS) aktiviert werden:

CUPS (Common Unix Printing System) ist für die Verarbeitung und Zwischen - speicherung der Druckdaten zuständig. Es sorgt dafür, dass der Drucker die Daten lesen, sprich drucken kann. Dabei wird auf PostScripts aufgebaut. Eine Programmiersprache, die den geschriebenen Text für den Drucker umwandelt. Eine weitere Aufgabe von CUP ist, die Druckaufträge in Empfang zu nehmen und sie in einer Warteschlange solange zu behalten, bis der Drucker frei ist diese auszudrucken. Der älteste Druckauftrag wird dann zuerst ausgedruckt.



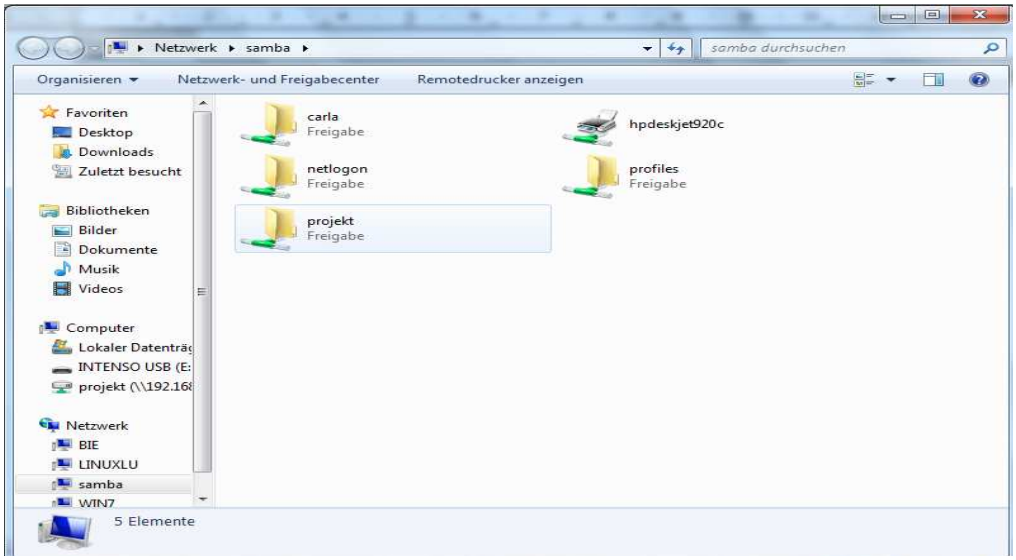
Bevor das Fenster geschlossen wird, können noch mal die getroffenen Einstellungen kontrolliert werden. Um herauszufinden welche Drucker am System angeschlossen sind, geht man in Yast ins Menü Hardware in Drucker. Dort werden alle erkannten Drucker von Suse aufgelistet.

Hier können verschiedene Einstellungen getroffen werden. Z. B. über den Button „Bearbeiten“ können Treiber gesucht und gefunden werden:



Durch den „smb.conf- Eintrag“ „printcap name = cups“ und der Freigabe „printers“ weiß Samba, dass alle Clients Zugriff, auf die am System angeschlossen Drucker, haben sollen.

Auf der Clientseite kann man mit der Angabe „\\samba“ im Browserfenster die Freigaben des PDC sehen. Darunter befinden sich auch die, am System angebundenen, Drucker. Bei Erteilung eines Druckauftrages gibt Win7 eine Fehlermeldung aus, dass kein Windows- Treiber auf dem Server zu finden sei.



Genauer beschreibt das Klimke in seinem Artikel „Tanzschule“:

> „Die Druckertreiber erhalten Sie aus der Windows-7-Installation; es handelt sich um die Dateien psSui.dll, pscript.hlp, pscript.ntf, pscript5. dll. Sie finden sie in Unterverzeichnissen von C:\WINDOWS\SYSTEM32\SPOOL\DRIVERS\: In W32C86\3 liegen die 32- Bit Treiber, in X64\3 die 64-Bit-Treiber. Sie kopieren die Dateien nach usr/share/cups/drivers respektive usr/share/cups/drivers /x64, wobei Sie die Dateinamen noch in Kleinbuchstaben konvertieren müssen, damit Samba sie korrekt anbieten kann. Mit dem Kommandozeilenbefehl cupsaddsmb –a fügen Sie nun alle Druckertreiber in den unter [print$] in der Samba-Konfiguration angegebenen Pfad ein…“.[17]

Gibt der Win7 –Client jetzt einen Druckauftrag, werden die Treiberdaten aus der Freigabe [print$] geholt und der Druckauftrag ausgeführt.

15. Fazit

Linux wird immer bekannter. Samba als ein „Vielkönner“, wird schon lange in Netzwerken verwendet, was wir als „Normaluser“ gar nicht als solches identifizieren können, da es im Hintergrund abläuft und die Arbeitsoberfläche meist Windows ist. Linux sollte noch mehr Bekanntheitsgrad gewinnen, als Gegenpol zur Monopolstellung von Microsoft. Es geht nicht nur darum, dass Linux kostenfrei aus dem Internet zu erwerben ist und Samba automatisch beinhaltet, sondern auch um die Art der Veröffentlichung durch die General Public License (GPL), die Linux und damit auch Samba nicht nur kostenlos, sondern „frei“ macht. Die Chance den Kernel weiterzuentwickeln, beinhaltet auch die Gelegenheit ein System aufzubauen, ganz nach meinen Vorstellungen oder betriebsspezifischen Vorgaben.
Zwar ist Linux mit seiner Konsolenanwendung für Windows- gewöhnte Augen und Hände eine echte Umstellung (das Lernen der Befehle, die Art und Weise wie sie funktionieren...), aber meines Erachtens geht die Rechnung auf, durch die vielen positiven Möglichkeiten, die Linux bietet. Schließlich ist man ein Gewohnheitsmensch und hat man sich einmal daran gewöhnt fällt es einem wahrscheinlich gar nicht mehr auf.

---

[17] Klimke (2010) S. 38

Mein Fazit: OpenSuse ist eine echte Alternative zu Windows. Mit Samba lässt sich ein Heimnetzwerk aufbauen, das Drucker-, Datei-, Internet- und noch viele andere Dienste zur Verfügung stellt.

16. Literaturverzeichnis

Behlert, Bodammer u. a. (2005):SUSE LINUX Administrationshandbuch. In: Max- Plank –Gesellschaft zur Förderung der Wissenschaften (Hrsg.) <http://www.mpipks-dresden.mpg.de/~mueller/docs/suse9.3/suselinux-adminguide_de/html/cha.samba.html> Download vom 19.05.2010.

Eckstein, Collier- Brown u. a. (1999): Using Samba. In: Linux User Krems (Hrsg.): Samba 2.0 Deutsch von Wolfgang Wald. <http://lug.krems.cc/docu/samba/ch06_06.html > Download vom 18.05.2010.

Emmerich, Gappel u. a. (2002): Konfiguration eines Samba-File-Servers unter Linux zur Anbindung von Windows-Clients. Projekt im Fach Kommunikations-technik. In: Dipl. - Ing. Alexander Scheib (Hrsg.), BBS-NW: Konfiguration eines Samba-File-Servers unter LINUX zur Anbindung von Windows-Clients. <http://www.scheib.info/downloads/Samba2002.pdf>. Download vom 24.04.2010.

Flo84 (2007):Smb.conf. In: Thomas Mönkemeier (Hrsg.): Linupedia, dem Wiki des LInuxclub. <http://www.linupedia.org/opensuse/Smb.conf>. Download vom 15.04.2010.

Goldmann, Martin (Hrsg.) (1999-2010): Samba als Primären Domänencontroller (2003). <http://www.goldmann.de/samba-als-pdc_tipp_390.html>. Download vom 23.03.2010.

Goldmann, Martin (Hrsg.) (1999-2010): Samba-Server mit Suse 10.1. <http://www.goldmann.de/samba-suse-pdc-linux_tipp_431.html>. Download vom 23.03.2010.

Grünbacher, Andreas (2003): Access Control Lists unter Linux. Administrator - handbuch. In: Novell. com (Hrsg.), Fachgebiet Suse <http://www.suse.de/ ~agruen/acl/chapter/fs_acl-en.pdf >. Download vom 15.04.2010.

Hansch, Jochen u.a. (2005): Aufbau eines Public Domain Server unter Linux mit Hilfe von Samba. Studienarbeit bei Prof. Dr. A. Grebe, WS 2004/05. In: Fachhochschule Köln, Fachgebiet Datennetze (Hrsg.). < http://www.dn.fh-koeln.de/download/arbeiten/WS0405_PDC_Samba.pdf>. Download vom 30.03.2010.

Klimke Markus (2010):Windows 7 im Samba-Netzwerk, Tanzschule. In: linuxuser (Hrsg.): Schwerpunkt: Heterogenes Netz, Ausgabe 02|2010, Seite 32 – 39.

Kofler, Michael (2010): Linux 2010, Debian, Fedora, openSUSE, Ubuntu, 9., überarbeitete und erweiterte Auflage. Addison-Wesley Verlag, München 2010.

Meißner, Jens (aktualisiert 2010): Domain Controller. In: Wikimedia Foundation Inc. die freie Enzyklopädie (Hrsg.). <http://de.wikipedia.org/wiki/Domain_Controller>. Download vom 31.03.2010.

Röhl, Sebastian (2004): PDC (Primary Domain Controller). In: Regionales RechenZentrum Erlangen (Hrsg.). <http://www.rrze.uni-erlangen.de/ausbildung/berufsausbildung/pdf/pdc.pdf.> Download vom 03.04.2010.

Rubner, Stefan (2004): Linux als PDC mit Samba 3. In: IDG Business Media GmbH (Hrsg.). <http://www.tecchannel.de/storage/nas/402321/linux_als_pdc_mit_samba_3/>. Download vom 03.04.2010.

Scheib, Alexander, Skripte Unterricht BBSNW.

Stelter, Nick und Vargovic, Predag (2007): Samba - Helper für VisualMedia, Freies E-Book. In: Dipl. - Ing. Alexander Scheib (Hrsg.), BBS-NW: Samba als PDC unter SuSE 10.2 (2009). <http://www.scheib.info/downloads/Samba_helper_for_visualmedia.pdf> Download vom 23.03.2010.

Terpstra, Micouleau u. a.: The Official Samba 3.5.x HOWTO and Reference Guide. Chapter 12. Group Mapping: MS Windows and UNIX. In: SAMBA, opening windows to a wider world (Hrsg.). <http://www.samba.org/samba/docs/man/Samba-HOWTO-Collection/groupmapping.html> Download vom 18.05.2010.

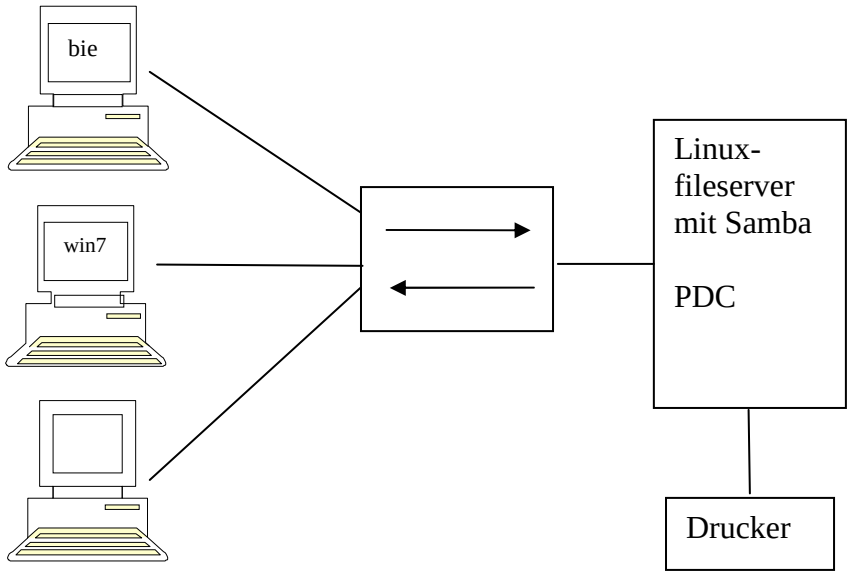
Vernooij, Terpstra u. a (2005): Samba- HOWTO – Sammlung, Kapitel 12: DAS GRUPPEN – MAPPING - ZWISCHEN MS WINDOWS UND UNIX. In Samba (Hrsg.). Deutschsprachige Übersetzung der Dokumentation zu Samba 3. < http://gertranssmb3.berlios.de/Samba-HOWTO-Sammlung.pdf>. Download vom 18.05.2010.

Vogel, Martin (Hrsg.) (2005): Domain, EDV – Lexikon. <http://lexikon.martinvogel.de/domain.html>. Download vom 02.04.2010.

Weidel, Thomas und Janina Masser (2002): Samba – Linux im Windows – Netzwerk. Projekt im Fach Mehrplatzbetriebssysteme. In: Dipl. - Ing. Alexander Scheib (Hrsg.), BBS-NW: Samba Einfacher Domänencontroller SuSE 9.3. <http://www.scheib.info/downloads/Samba_Einfacher_Domaenencontroller_suse93.pdf>. Download vom 24.04.2010.

17. Anhang

17.1 Aufbau der Testumgebung



Der Server ist ein Multibetriebssystemrechner. Auf ihm waren bereits Windows XP Home und Vista installiert. Auf einer noch freien, nicht mit NTFS formatierten Partition, wird Linux OpenSuse 11.2, eine 64- Bit- Version, installiert. OpenSuse 11.2 beinhaltet die Software Samba 3.4.2, mit deren Hilfe der Rechner zum Fileserver und PDC wird.

Der Festplattenspeicher beträgt ca. 100 GB. Suse macht daraus 3 Teilbereiche: eine Swappartition, diese ist zu vergleichen mit der Auslagerungsdatei unter Windows, ein Teilbereich für „ / “ und ein weiteres für „ /home “. Diese Bereiche werden mit Ext4 formatiert und in „/“ bzw. „/home“ eingehängt (gemountet).

Das ganze sieht dann so aus:

Gerät	Größe	F	Typ	Fs-Typ	Einhängepunkt
/dev/sda	335.35 GB		ST3360320AS		
/dev/sda1	50 GB		HPFS/NTFS	NTFS	/windows/C
/dev/sda2	150 GB		HPFS/NTFS	NTFS	/windows/D
/dev/sda3	35 GB		HPFS/NTFS	NTFS	/windows/E
/dev/sda4	100.35 GB		Extended		
/dev/sda5	2.01 GB	F	Linux Swap	Swap	Swap
/dev/sda6	20.00 GB	F	Linux native	Ext 4	/
/dev/sda7	78.34 GB	F	Linux native	Ext 4	/home

Da es zwei Windows- Betriebssysteme und eine weitere, freie, NTFS-formatierte Partition auf dem Rechner gibt, macht OpenSuse in jedes Laufwerk einen Mountpunkt. Dies beinhaltet die Möglichkeit von Suse aus auf die Laufwerke (C; D, E) zugreifen zu können. In der Konfigurationsdatei /etc/fstab ist zu sehen, welche Laufwerke beim Rechnerstart in den Verzeichnisbaum gemountet werden. Mit dem Befehl mount bzw. umount können z. B. Laufwerke eingebunden bzw. entfernt („ausgehängt“) werden, geben somit die Möglichkeit auf Zugriff oder nicht.

Der Client PC ist ebenfalls ein Multibetriebssystemrechner. Er war genauso ausgestattet wie o. g. Server- Rechner, da aber WIN XP Home und Vista sich nicht in eine Domäne einbinden lassen, müssen domänfähige Betriebssysteme installiert werden. Dafür wird die komplette Festplatte neu formatiert.

In die erste Partition kommt Windows XP Professionell. Bei der Installation von der CD wird eine neue Partition mit 80 GB Speicher erstellt. Dann wird WIN7 installiert, auf eine zweite ca. 100 GB große Partition (die Partitionsgrößen sind frei gewählt). Beide Partitionen wurden von Windows mit NTFS formatiert. Windows konfiguriert das meiste selbst. Bis auf die Angabe der Uhrzeit, evtl.

der Sprache und Tastaturbelegung, geht alles automatisch. Am Anmeldebildschirm muss man den Benutzer und Rechner definieren. Nachdem beide Betriebssysteme laufen, wird jeweils Adobe Reader, Adope Flash Player, OpenOffice, Firefox und Avira Antivirenprogramm eingerichtet. Die meisten Pakete sind aber für die Testumgebung nicht wichtig.

Damit die Kommunikation zwischen den Rechnern funktioniert, werden diese mit einem Patchkabel (Kat 5, cross wiring, for Gigabit Ethernet) über die Netwerkkarten verbunden. Ein Switch ist dazwischen geschaltet.
Und schon kann es losgehen. OpenSuse starten, Samba als Primary Domain Controller konfigurieren, die Windows- Clients in die Samba- Usern Datenbank aufnehmen….
Die Benutzer dieses Testszenarios sind die User bbs, anton und carla.
Das Sambakennwort für die Benutzer ist zur Vereinfachung für alle „geheim“.

17.2 Problemstellung

Mehrere Clients, die auf unterschiedlichen Betriebssystemen, mit unterschiedlichen Oberflächen arbeiten (z. B. Windows, Mac Os, Linux), sollen die Möglichkeit haben ein und denselben File - Server zunutzen. Dies soll möglich sein, damit Benutzer mit gleichen Aufgaben mit gleichen Ressourcen und Verzeichnisse arbeiten können. Damit soll auch die Verwaltung der Dateien, Verzeichnisse und Profile vereinfacht und zentral vom Server aus geschehen.

Dabei sollen die Kosten für Lizenzen eingespart werden, deshalb soll ein Open Source Programm (Linux mit integriertem Samba 3.4) gewählte werden.
Im dargestellten Beispiel wird der Server unter Linux Open Suse 11.2 laufen und es werden zwei Client in die Beispieldomäne einbezogen, voraussichtlich einer unter Win XP und ein weiterer unter Win7.

Auf diesem Server soll die zentrale Rechteverwaltung durchgeführt werden. Diese Rechteverwaltung bezieht sich auf die Benutzer, genauso wie auf Dateien und Verzeichnisse. Clients sollen nur auf bestimmten Verzeichnisse und Dateien zugreifen können und diese Einstellungen sollen sich bei Bedarf weiter vererben. Änderungen oder Anlegen von Benutzerkonten sollen zentral mit dem Domaincontroller geschehen.
Die Rechteverwaltung ist z. B. über Gruppenzugehörigkeit des jeweiligen Clients zu bewerkstelligen. So können Benutzer mit gleicher Aufgabenstellung gleiche Rechte zugewiesen bekommen.

Login- Skripte geben den Usern die Möglichkeit ihre speziellen Benutzerprofile von Rechner zu Rechner mitzunehmen, denn diese Einstellungen sind nicht mehr auf einem Rechner lokal installiert, sondern auf dem Server. Sie werden

geladen, sobald sich der User am Server anmeldet. Damit meldet sich der User an und gelangt sofort auf die benötigte Arbeitsumgebung.

Es wird ein Drucker am Sambaserver angeschlossen, damit die Clients über das Netzwerk zentral drucken können.

17.3 Kommentar

Motiviert durch den Unterricht bei Herrn Scheib wollte ich in meinem Abschlussprojekt unbedingt etwas mit EDV tun. Linux, und was man mit Samba machen konnte (Laufwerksmapping, Freigaben), hat mir gut gefallen. Deshalb entschloss ich mich für dieses Projekt, ohne wirklich zu wissen, was ein Domaincontroller ist. Oft bin ich auf Verständnishürden gestoßen (Linux ist eben nicht zu vergleichen mit Windows und private Erfahrungen mit Linux hatte ich nicht). Bei den Problemen war das Schulwissen teils hilfreich, reicht aber oft nicht aus, um sich in der Linuxwelt zurechtzufinden. Viele Linuxseiten arbeiten mit Fachbegriffen, die ich dann wieder separat nachschlagen musste, was die Arbeit am Projekt erschwerte. Ich habe mir oft ein Team gewünscht, das sich gegenseitig im Projekt unterstützt!

Wenn dann, nach langer Mühe, einzelne Funktionen wie z. B. die Client-einbindung in die Domäne gelangen, war das sehr motivierend. Vor allem deshalb, weil ich die Fehler selbst fand, korrigierte und dann zu meinem Ziel kam. Fehler sind und bleiben die besten Lehrmeister!
Viele Beschreibungen, z. B. die Kontrolle der IP- Adresse auf dem Server, resultieren aus diesen Fehlern. Vielleicht helfen sie anderen, sie nicht zutun.

Dieses Script ist mit Sicherheit nicht vollständig. Es ist ein Versuch in verständlicher Weise zu erklären wie man mit einigem Vorwissen Samba einsetzt, um einen Domaincontroller aufzubauen. Ich hätte gerne noch so vieles ausprobiert und hinzugefügt, aber leider reichte die Zeit nicht.

Ich möchte all denen Mut machen, die sich auch etwas zu weit aus dem Fenster gelehnt haben und vielleicht ins Zweifeln kommen, weil das Ziel scheinbar in weiter Ferne liegt und die Zeit so knapp wird. Wenn das Resultat da ist, sieht man es mit anderen Augen.

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich diese Arbeit selbständig und nur mit den angegebenen (erlaubten) Hilfsmitteln angefertigt habe. Alle Stellen, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen sind, sind durch Angabe der Quellen kenntlich gemacht worden.

Neustadt an der Weinstraße, den 21.05.2010

Ramona Ludwig