Die WCM Linux-Box als Fileserver

Die Linux-Box lernt Samba

Mit einigen wenigen heißen Tanzschritten kann die WCM-Linux-Box anderen Rechnern im Netz als zentraler Massenspeicher dienen. Samba, das Fileserver-Paket der freien Entwicklergemeinde, verhilft unserer Box zu neuen Einsatzmöglichkeiten.

von Martin Müller

Im vierten Teil unseres Workshops zur Konfiguration der Linux-Box, beschäftigen wir uns mit den Fileservices im LAN. Für die zugrunde liegende Distribution Debian ist es kein Problem, sämtliche gängigen Fileservices anzubieten. Egal ob Network-File-System (NFS) für Linux-Clients, AppleTalk für MacIntosh-Clients oder Samba für die wohl am meisten verwendeten Windows-Clients - mit wenigen Handgriffen stehen die entsprechenden Dienste schnell zur Verfügung.

Linux 2 Linux

Der Server läuft, wir sind als root an einer Konsole angemeldet. Durch *apt-get-install nfs-kernel-server nfs-common* wird der NFS-Server installiert und seine Konfigurationsdatei unter */etc/exports* angelegt. In */etc/exports* geben Sie nun die freizugebenden Verzeichnisse, gefolgt von Rechnername oder IP-Adresse sowie der Berechtigung an (siehe Kasten NFS-Server). In der ersten Zeile geben wir das Verzeichnis */usr/local* für alle Rechner aus dem internen Netz frei. Der Zugriff darf nur lesend erfolgen (ReadOnly), Der Zusatz async bewirkt, dass etwaige Schreibzugriffe vom Server frühzeitig als erfolgreich an den Client zurück gemeldet werden. Dies ist zwar unsicher, denn der Server könnte vor der tatsächlichen Sicherung abstürzen, der Geschwindigkeitszuwachs gegenüber der Option sync beträgt jedoch circa Faktor 10, weshalb wir dennoch diese Option wählen sollten.

Achtung: Wenn der Linux-Box die Rechnernamen der Clients bekannt sind, müssen die Rechnernamen explizit in der exports-Datei angegeben werden! Hier gilt: Namensauflösung kommt vor der IP-Adresse zur Anwendung. Dem Server sind die Rechnernamen dann bekannt, wenn diese entweder in der *hosts*-Datei eingetragen sind oder wenn sie im DNS eingetragen sind. Nähere Angaben zur *hosts*-Datei und dem Dynamic-Name-Service finden Sie im Workshop der letzten WCM-Ausgabe.

So was von verboten ...

In der Datei */etc/hosts.allow* können Sie außerdem explizit den Zugriff für Clients freigeben. Die Konfigurationsdatei */etc/hosts.deny* sorgt für das Gegenteil – Rechnern wird der Zugriff verweigert. Die Clients die in dieser Datei angegeben werden, haben keine Berechtigung den angeführten Dienst zu nutzen – auch wenn der Client in *exports* angegeben ist. Den genauen Syntax der allow/host-Dateie entnehmen Sie bitte wieder dem Kasten NFS-Server. Das Beispiel für *allow* gewährt Zugriff auf den Dienst portmap (NFS-Dienst) für alle Rechner aus dem Netz 192.168.123.X außer für den Rechner mit der IP 192.168.123.95. Das Beispiel *deny* verbietet den Zugriff auf den ssh-Dienst für alle Rechner außer für den Client mit der IP-Adresse 192.168.100.100. Diese beiden Dateien bieten eine Vielzahl von Einschränkungsmöglichkeiten, weshalb Sie äußerste Sorgfalt walten lassen müssen. Ungewollte Einschränkungen in dieser Datei lassen die entsprechenden Dienste für Sie anscheinend als nicht funktionierend erscheinen. Tatsächlich haben Sie nur den Zugriff verboten, der Dienst hingegen läuft einwandfrei. Brauchen Sie die Möglichkeit des Sicherheitsgewinns durch dieses Dateipaar nicht, so lassen Sie die Dateien unverändert.

Um die konfigurierten NFS-Shares der Öffentlichkeit zugänglich zu machen, setzten wir das Kommando *exportfs –a* ab. Anschließend können Sie mit *rpcinfo –p* prüfen, ob der Daemon auch läuft. In der Liste erscheint das Programm portmapper, der NFS-Server läuft.

Die lokalen Dateirechte werden durch den Export mit übergeben. Es ist also nötig, den User des zugreifenden Clients auch am Server angelegt zu haben. Und es ist notwendig, dieselbe Benutzer- und Gruppeninformation (UID und GID) für die User an den betroffenen Maschinen zu haben. Die entsprechenden Benutzereigenschaften auf allen Maschinen gleich zu halten, ist bei größeren Netzwerken ein großer Aufwand, weshalb ein zentraler Login-Server notwendig wird. So einen Server werden wir zu einem späteren Zeitpunkt konfigurieren.

NFS-Server

```
#/etc/exports

/usr/local192.168.123.*(ro, async)
/usr/sharebeispielrechner(rw,async)

                    ***

#hosts.allow

portmap:192.168.123.0/24 : ALLOW EXCEPT 192.168.123.95

                    ***

#hosts.deny

sshd:DENY ALL EXCEPT 192.168.123.100

                    ***

Ausgabe von rpcinfo –p

program vers proto   port
100000   2     tcp   111  portmapper
```

Äpfel und Pinguine

Damit auch MacIntosh-Clients an der Datenvielfalt teilhaben können, benötigen wir das Paket *netatalk*. Der übliche Installationssyntax *apt-get install netatalk* holt das Paket vom vorgegebenen Installationsort, installiert den Server und legt die Konfigurationsdateien unter */etc/netatalk/* an.

Wir schreiben ans Ende der Datei */etc/netatalk/AppleVolumes.default* die Zeile */Apple „Delicious" allow:@wcm* wodurch wir das Verzeichnis */Apple* unter dem Namen *Delicious* für alle Benutzer die in der Linux-Benutzergruppe *wcm* zugänglich machen.

In der Datei */etc/netatalk/atalkd.conf* schreiben wir ans Ende das Netzwerkinterface auf dem atalk seine Dienste anbieten soll (Beispiel *eth0*). Nun noch netatalk neu starten (*/etc/init.d/netatalk restart)* und schon kann man auf den Server zugreifen. Bitte beachten Sie, dass Benutzername und Passwörter direkt aus dem Linux-System verwendet werden. Sie müssen also die User und Gruppen die Zugriff haben sollen auf der Box anlegen. Wenn Sie mehr Kontrolle über den Netatalk-Server brauchen, schauen Sie sich */etc/netatalk/afpd.conf* genauer an. Netatalk ist

mittlerweile sehr ausgereift, die unzähligen Konfigurationsmöglichkeiten entnehmen Sie direkt aus den sehr gut dokumentierten Konfigurationsdateien.

Mystische Rhythmen

Zu guter letzt wollen wir es auch Windows-Clients ermöglichen, Dateien auf der WCM-Linux-Box zu speichern oder zu lesen. Dazu brauchen wir das Paket *samba* sowie die entsprechenden Client-Pakete *sbmclient*.

Um das Windows-Netzwerk ein wenig besser verstehen zu können, erst mal ein paar Begriffe:

SMB (Server Message Block) ist das von Microsoft entwickelte Netzwerkprotokoll, das von Samba nachgebildet wird. Da Microsoft den freien Entwicklern keinerlei Information dazu zur Verfügung stellt, müssen diese in mühsamer Kleinarbeit den Netzwerkverkehr zwischen den Windowsrechnern abhören und das gewonnene Wissen in die Programme einfließen lassen. Da es sich bei Samba demnach um eine völlige Neuprogrammierung handelt, laufen einige Funktionen sogar sicherer und stabiler als beim Original. Andererseits ist noch nicht der volle Funktionsumfang des SMB in der aktuellen Sambaversion verfügbar.

Das *NetBIOS*, ein von IBM entwickeltes Protokoll, kümmert sich um den Datentransport im Netzwerk. Es stellt unter anderem den *NetBIOS-Nameservice* (NBNS) zur Verfügung. Dieser Nameservice hat wie DNS die Aufgabe, dass sich Rechner untereinander mit Bezeichnungen ansprechen können. Anders als beim DNS muss diese Aufgabe nicht zwingend von einem Server übernommen werden, da ein Client beim Hochfahren seiner Netzwerkverbindung jedem Rechner im Netz mitteilen kann, wie er heißt.

Des Weiteren gibt es zwei Kommunikationsmechanismen innerhalb des NetBIOS. *Datagramm* ist ein verbindungsloser Datenaustausch. Er ermöglicht es, Daten an mehrere Clients gleichzeitig zu versenden, hat aber den Nachteil dass es keine Empfangsbestätigung für die Daten gibt.

Der zweite Mechanismus, *Sessions* genannt, baut feste Verbindungen von Rechner zu Rechner auf, mit allen Vorteilen, wie Empfangsbestätigung und Fehlerkorrektur der Daten.

Nicht zu verwechseln mit dem *NBNS* ist das *WINS*, das Windows Internet Name Service. Auch dieses Nameservice braucht keinen zwingenden Server, da es die Daten mit Hilfe von Datagramm-Paketen an alle Rechner im Netzwerk sendet. Leider ist dieses Verfahren in großen Netzwerkwerken sehr ineffizient, da es eine Zeit dauert, bis alle Rechner gegenseitig die richtigen Namen ausgetauscht haben.

Das Match der Browser

Um diesen Vorgang zu beschleunigen, setzen wir Samba als Masterbrowser ein. Somit teilen alle Windowsclients unserer WCM-Linux-Box ihren Namen mit und fragen die Box, welche anderen Rechner es sonst im Netz gibt. Es ist wichtig der Linux-Box einen möglichst hohen Browser-Wert zu geben, da ein Masterbrowser nicht von vorn herein fest steht. Jeder Windows-Client könnte sich selbst als Masterbrowser definieren – es werden so genannte Matches um den Browserstatus ausgetragen. Damit unsere Box jedes Match gewinnt, weisen wir ihr später den Wert 64 zu.

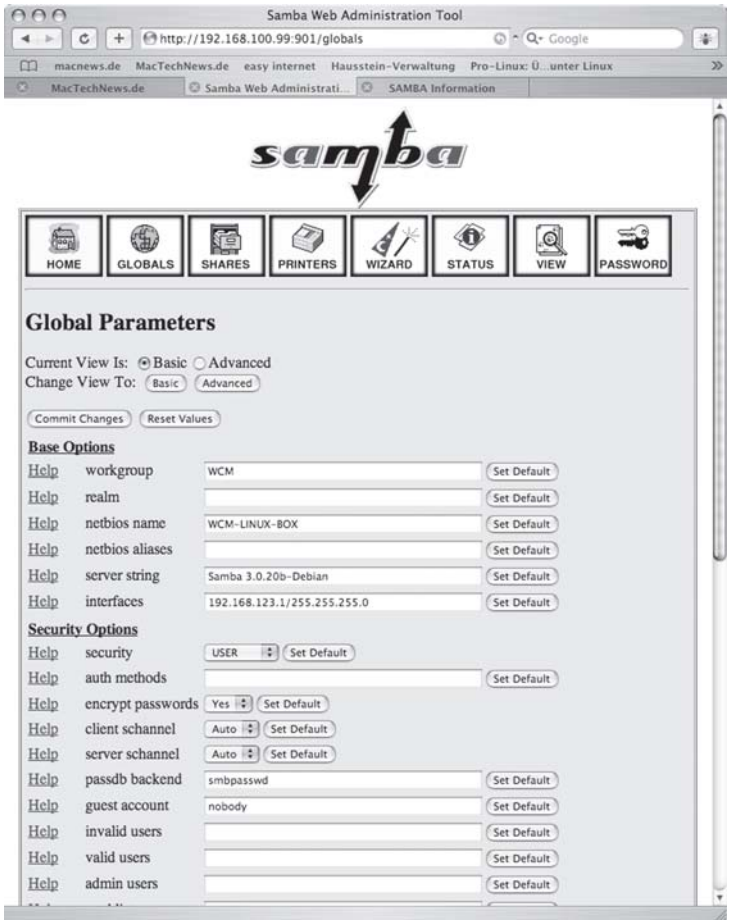
Da wir die WCM-Linux-Box als WINS-Server und gleichzeitig als Masterbrowser betreiben, sind im Regelfall die verteilten Browsinglisten aktuell und die Clients sehen auch nur den tatsächlichen Zustand des aktuellen Windows-Netzwerkes.

Zugriffsrechte und Freigaben

Shares kennen eingefleischte Windows-Nutzer unter dem Begriff Freigabe. Jedes frei gegebene Objekt wird als Share bezeichnet, egal ob es eine Dateifreigabe ist oder ein Drucker. Wir wollen auch die Bezeichnung Share verwenden, weil für die Samba-Benutzeroberfläche SWAT nur eine englischsprachige Oberfläche zur Verfügung stellt. Samba kann auf drei Arten Shares anbieten die sich durch unterschiedliche Sicherheitsstufen unterscheiden. Bei der *Share-Level-Sicherheit* wird jedes Share durch ein eigenes Passwort geschützt. Wird kein Passwort vergeben steht der Zugriff für alle Benutzer offen. Dieses System funktioniert auch nur dann, wenn sich die zugreifenden Clients in derselben SMB-Arbeitsgruppe befinden.

Die *User-Level-Sicherheit* geht einen Schritt weiter und setzt ebenfalls dieselbe Arbeitsgruppe voraus. Shares werden nun mit dem aktuell am Client verwendeten Benutzernamen und Passwort angesprochen. Sind Sie am Windows-Rechner mit dem Benutzernamen *Elias* und dem Passwort *Engel* angemeldet, so wird dieses Identifikationspaar an den Server übergeben. Trifft in der SMB-Serverdatenbank kein Muster auf dieses Paar zu, wird der Zugriff verweigert. Vorteil: Keine zusätzlichen Passwörter nötig, Sie können von jedem Windowsclient mit denselben Rechten auf den Server zugreifen. Nachteil: Damit sie tatsächlich „wandernde User" verwenden können, also von jedem Rechner mit den selben Rechten auf den Server zugreifen können, müssen Sie auf jedem Client den entsprechenden Benutzer angelegt haben und idente Passwörter vergeben haben. Und Sie müssen die Benutzer-Datenbank am Server mit den Benutzern an den Clients immer im Gleichstand halten. In kleinen Netzwerken ist dies kein Problem und der Sicherheitsgewinn durch die User-Level-Sicherheit enorm. Eine weitere Anmeldungsart ist die Domain-Level-Sicherheit. Der Client greift bei der Anmeldung auf die Benutzerdatenbank am Server zu, lokale Benutzer werden nicht beachtet. Es besteht die Möglichkeit Ressourcen des Clients trotzdem über die User-Level-Sicherheit frei zu geben. Der fundamentalste Unterschied besteht jedoch hinter den Kulissen: Die Zugriffsrechte werden über Tokens vergeben die für das gesamtes Netzwerk gelten. Für den User bedeutet dies keinen Unterschied, für den Server ist diese Art der Anmeldung jedoch bedeutend effizienter zu handhaben. Leider kann Samba diese Art der Anmeldung noch nicht eigenständig nutzen, es nimmt nur die Authentifizierungsanfragen entgegen und reicht sie an einen

SWAT im Einsatz: Das Bild zeigt die Konfigurationsmöglichkeiten für die globalen Einstellungen.



primären oder sekundären Windows-Domain-Controler weiter.

Um die Anmeldemöglichkeiten vollständig erwähnt zu haben, muss auch noch das Active Directory in die Liste. Es ist eine Weiterentwicklung des Domain-Level-Konzepts, steht aber in der aktuellen stabilen Version von Samba leider noch nicht zur Verfügung.

Genug der grauen Theorie, wir entscheiden uns aus Sicherheitsgründen für das User-Level-Sicherheitskonzept. Dazu installieren wir Samba, das SMB-Client-Paket, ein Paket dass mit dem Netzwerkdateisystem umgehen kann und das sambaeigene Verwaltungstool SWAT (*apt-get install samba-common smbclient smbfs swat*).

Bei der Installation werden wir nach dem Arbeitsgruppennamen gefragt (vorerst egal) und ob wir verschlüsselte Passwörter verwenden wollen (*JA*). DHCPD3 wollen wir installieren und verwenden und Samba wollen wir als *daemon* starten. Die Samba-Passwort-Datenbank soll im vorgeschlagenen Verzeichnis erstellt werden und SWAT darf unsere *smb.conf* getrost überschreiben. Die Installation ist fertig, wir gehen ans konfigurieren.

Damit SWAT auch benutzbar wird, muss in */etc/services* der Port 901 für das TCP Protokoll freigeschalten sein. Dies ist normalerweise der Fall, eine Überprüfung kann jedoch nicht schaden (*pico /etc/services*). Der Eintrag in Zeile 421 darf keine Raute (#) am Beginn stehen haben. Die Freischaltung von SWAT finalisieren wird durch Entfernung der Kommentarzeichen *#<off>#* am Ende der Datei */etc/inetd.conf*. Die beiden letzten Zeilen müssen nun mit *netbios-ssn* und *swat* anstatt mit den Kommentarzeichen beginnen.

Danach starten wir den *inetd* neu *(killall –HUP inetd)* und schon können wir mit einem beliebigen Browser aus dem lokalen Netz auf SWAT zugreifen. Einfach die IP-Adresse des Servers, gefolgt vom Port 901, an dem SWAT lauscht, angeben (*http://192.168.123.1:901*). Wir können uns als root mit dem root-Passwort anmelden um in der nun bunten Welt von SWAT das Sambapaket zu konfigurieren.

SWAT– Einsatzkommando

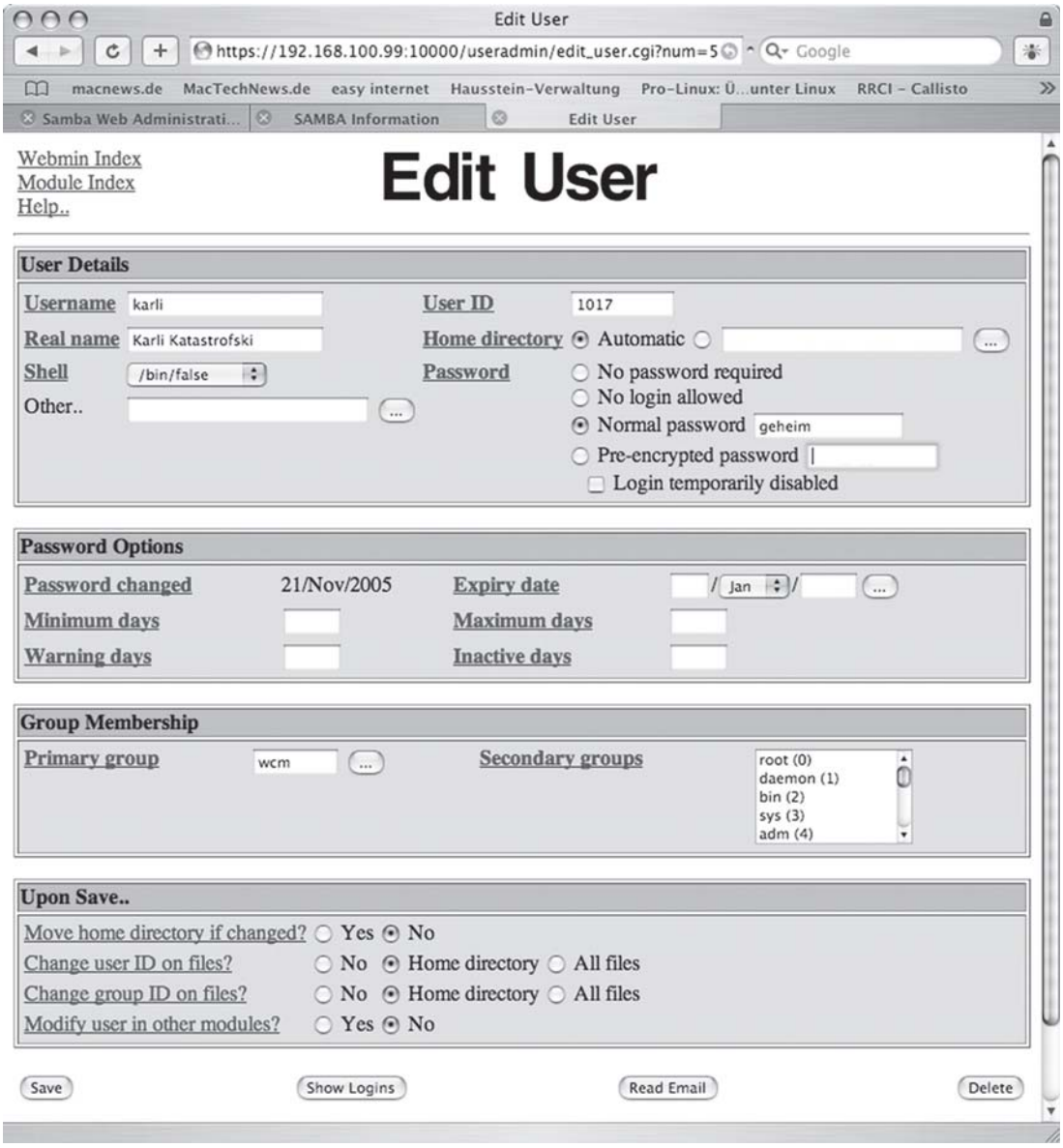
Nach dem Anmelden wird einem die hervorragende Dokumentation zu Samba präsentiert. Die integrierte Doku ist bei SWAT allerdings nur in englischer Sprache verfügbar. Deutschsprachige Hilfe dafür findet man unter http://samba.sernet.de/info.html.

Navigiert wird bei SWAT mit Hilfe der Iconleiste direkt unter dem großen Samba-Logo.

Im Abschnitt„*GLOBAL* geht's an die primären Eigenschaften von Samba die für alle Shares Gültigkeit haben und das allgemeine Serververhalten betreffen. Als erstes muss im Abschnitt *BASE OPTIONS* der Arbeitsgruppennamen unter *Workgroup* angegeben werden. Im Beispiel entscheiden wir uns für WCM, der Netbios-Name unter dem die Box erscheinen soll, wird zwei Kästen weiter unten angegeben. Danach kann ein Kommentar vergeben werden, der in der Netzwerkumgebung neben dem Servernamen zu sehen ist. In der nächsten Box *Interfaces* geben wir nun die Geräte oder IP-Adressen an, über die Samba Netzwerkinformationen verteilt.

In den *SECURITY-OPTIONS* stellen wir die Sicherheit auf *User* und lassen die Auth-Methode unverändert. Somit werden User an der lokalen (Server)-Datenbank authentifiziert. *Encrypted Passwords* sind ab Windows 98 kein Problem, verwenden Sie ältere Windows-Clients, müssen Sie die Passwortverschlüsselung auf *NO* stellen. Der nächste für uns wichtige Eintrag lautet *passdb backend,* mit dem wir die Benutzerdatenbank angeben. Zunächst geben wir hier *smbpasswd* an, in einer späteren Ausgabe werden wir die Benutzerdatenbank an die MySQL-Datenbank übergeben. Den *Guest Account* belassen wir beim User *nobody*. Dadurch erlangen Gäste die sich am Server anmelden nur die Rechte des Unix-Users *nobody*. *Nobody* hat von Grund auf fast keine Rechte auf der Box und gilt deshalb als sehr sichere Option. Die weiteren Einträge in dieser Sektion lassen wir unbeachtet.

Erst bei den *Browser Options* legen wir wieder Hand an, indem wir den *Os Level* stellen um als Masterbrowser zu gewinnen. Außerdem wollen wir *Preferred Master* und *Local Master* werden, beide Einträge auf *YES* setzen. Domain-Master lassen wir auf *AUTO*.



Die Webmin-Edit-User-Seite bietet Einstellungen zum Anlegen eines Windows-Users

Bei den *WINS Options* setzen wir nur den *wins support* auf *YES* um als WINS-Server fungieren zu können. Der Eintrag *wins server* muss unbedingt leer bleiben!

So, die wichtigsten Einträge sind getan, scrollen Sie nach oben und bestätigen Sie die Einstellungen mit der Schaltfläche *Commit Changes*.

Die Einzelheiten

Als nächstes wollen wir die Freigaben selbst erstellen. Klicken Sie auf den *SHARE*-Knopf in der Titelleiste und geben Sie danach im Feld neben *Create Share* einen frei wählbaren Freigabenamen an (Beispiel: *WCM-Share)*. Danach legen Sie die Freigabe durch den Knopf *Create Share* an, wir werden zum Share-Konfigurationsfenster weiter geleitet. Sämtliche Einstellungen die hier getroffen werden, gelten nur für diese Freigabe. Bei *comment* können Sie Bemerkungen eingeben, die der User in der Netzwerkumgebung zu Gesicht bekommt. Der Eintrag *path* gibt das frei zu gebenden Server-Verzeichnis an. Wir legen auf der Konsole testweise das Verzeichnis WCM-FREI an (*mkdir /WCM-FREI)* und geben dieses bei SWAT bei *path* an.

Die *SECURITY OPTIONS* bieten uns Möglichkeiten, einzelnen Usern oder Gruppen den Zugriff auf dieses Share zu erlauben oder zu verbieten. Mehrere Benutzer geben Sie durch ein Leerzeichen getrennt an, Gruppen werden durch den Klammeraffen vor dem Gruppennamen definiert (zB @TESTGRUPPE). Wir definieren einen Eintrag im *Valid User*-Feld durch *@wcm*. Dadurch erlangt die Gruppe *wcm* Zugriff auf dieses Share. Weiters setzen wir den *READ ONLY* Eintrag auf *NO,* da wir generell auch auf dem Share schreiben wollen. *GUEST OK* belassen wir auf *NO* um einen möglichst hohen Sicherheitslevel zu erreichen.

Wollen Sie das Share in der Windows-Netzwerkumgebung verstecken, brauchen Sie nur den *BROWSABLE* Eintrag auf *NO* zu setzen, bei *YES* wird die Freigabe aufgelistet. Um das Share komplett vom Netz zu nehmen stellen Sie den Eintrag *AVAILABLE* auf *NO.* Dies kann sinnvoll sein, wenn Sie zum Beispiel das Verzeichnis am Server von einer Festplatte auf eine anderen verschieben wollen oder sonstige Wartungsarbeiten am Server durchführen. Somit kann kein User an den Daten arbeiten. *COMMIT CHANGES,* am Seitenanfang zu finden, schreibt die Einstellungen in die Datei /etc/smb.conf. Jetzt wechseln wir durch Betätigen des Knopfes *ADVANCED* in den erweiterten Konfigurationsmodus. Suchen Sie das Feld *create mask* im Bereich *SECURITY OPTIONS* und ändern Sie den Eintrag auf *0777*. Dies bewirkt, dass Samba die Gruppenrechte für jede neue Datei auf lesen und schreiben setzt. Bestätigen Sie wieder durch Drücken auf *Commit Changes.* Die ersten Einstellungen zur Freigabe sind erfolgt, wir starten Samba im *STATUS*-Bereich durch Betätigen der *RESTART ALL*-Schaltfläche neu.

Unix vs Windows

Nun geht's zurück ans Terminal. Wir brauchen auf der Box dieselben Benutzer, wie sie auch unter Windows existieren. Beispiel: Sie melden sich unter Windows XP mit dem Benutzernamen „Karli" und dem Passwort „geheim" an. Folglich brauchen wir auch auf der WCM-Linux-Box einen Benutzer „Karli" idealer Weise mit demselben Passwort.

Den Benutzer legen Sie am besten mit der grafischen Administrationshilfe Webmin an. Die Einstellungen für den Benutzer entnehmen Sie der Abbildung 2 (*Edit User)*. Auf Webmin und dessen User-Funktionsfeld sind wir in einem der vorangegangenen Workshopteile ausführlich eingegangen. Alternativ können Sie auch den Terminalbefehl *adduser* verwenden. Den neuen User legen wir gleich im Webmin-Userfeld in die neue Gruppe *wcm*, die von nun an vollen Zugriff auf die Freigabe *WCM-Share* haben soll. Administratoren die die Kommandozeile benutzen, verwenden das Kommando *addgroup*.

Nun müssen wir Samba noch mitteilen, dass es einen neuen Windows-User gibt, der auf unsere Shares zugreifen darf. *smbpasswd –a karli* legt den neuen User karli an, wir werden um das *smb*-Passwort gefragt. Das *smb*-Passwort muss nun unbedingt so lauten wie das Passwort das zur Anmeldung am Windows-Client dient! In unserem Beispiel also *geheim*.

Nun noch die Zugriffsrechte für das Verzeichnis richtig setzen: *chgrp wcm /WCM-VERZ* und *chmod g+rw /WCM-VERZ* weist das Verzeichnis der *wcm*-Gruppe zu und gibt Gruppenberechtigungen für lesen und schreiben. Wir wechseln ins Freigabeverzeichnis mit *cd /WCM-VERZ* und vergeben die letzten Rechte *chgrp wcm ..*

Client Setup

Nun sind wir in der Lage mit den Windows-Clients auf die Box zuzugreifen. Rufen Sie die Netzwerkumgebung auf und klicken Sie sich durch die Netzwerkhierarchie bis zur WCM-Box durch. Wenn Sie Benutzernamen und Passwort richtig vergeben haben, dann haben Sie jetzt vollen Zugriff auf das Samba-Share. Ab und zu kommt es vor, dass Windows meint Sie hätten keine Berechtigung auf die Arbeitsgruppe zuzugreifen. Wenn dies auftritt, dann verbinden Sie sich einfach über den Menüpunkt „Extras/Mit Netzlaufwerk verbinden" eines Windows Explorer-Fensters mit dem Share. Wählen Sie einen Laufwerksbuchstaben aus und geben Sie unterhalb des Laufwerksbuchstaben die IP-Adresse des Samba-Servers sowie den Freigabenamen an (\\192.168.123.1\WCM-Share).

Durch unterschiedliche Gruppenzugehörigkeiten können Sie verschiedene Zugriffsrechte auf die Shares einsetzen.

Zusätzliche Features

Wenn Sie in SWAT die Schaltfläche *ADVANCED VIEW* anklicken, dann bekommen Sie weit mehr Möglichkeiten zur Sambakonfiguration geboten. Für viele Anfänger schießen diese Einstellungen jedoch über das Ziel hinaus, Fortgeschrittene sollten sich aber mit den Einstellmöglichkeiten vertraut machen, bieten dieses Features doch viele Bequemlichkeiten. Zu jedem Konfigurationspunkt gibt es vorangestellt einen HELP-Eintrag. Wenn Sie auf diesen klicken, wird ein neues Browser-Fenster geöffnet, in dem Sie eine meist ausreichende Erklärung für den jeweiligen Punkt angezeigt bekommen. Wie Sie sicher bemerkt haben, könnte man auch Drucker die an der Linux-Box angeschlossen sind für die Windows-Clients frei geben. Dies ist allerdings ein sehr komplexes Thema, das wir in einer weiteren Folge dieser Reihe behandeln werden.

Es klappt doch nicht?

Sollte etwas nicht klappen, so prüfen Sie zunächst einmal, ob Samba überhaupt läuft. *smbclient –L localhost* fragt Sie nach einem Passwort, dass Sie bitte leer lassen. Drücken Sie einfach Enter und prüfen Sie in der Liste, ob unser Server und die aktiven Shares aufgelistet werden. Wenn das nicht der Fall ist, dann läuft der Daemon nicht. Grund dafür könnten Fehler in der Konfigurationsdatei sein, die Samba nicht starten lassen. Sie können den Syntax der Konfigurationsdatei mit dem Befehl *testparm* prüfen lassen. Stimmt in den Konfig-Dateien etwas nicht, werden die entsprechenden Fehlermeldungen ausgegeben.

Samba loggt das Geschehen rund um sich in der Datei /var/log/samba/log.smbd mit. Sie können sich diese Datei in einer separaten Shell live ansehen und somit den LogIn-Vorgang einzelner Benutzer mitverfolgen. Wechseln Sie dazu mit STRG+ALT+F2 in die nächste freie Kommandozeile, melden Sie sich dort als root an und setzen Sie den Befehl *tail –fn20 /var/log/samba/log.smbd* ab. Es werden jetzt immer die letzten 20 Zeilen der Log-Datei angezeigt.

Weiter führende Literatur dazu findet sich massenhaft im Internet. Die ersten Anlaufstellen sind dabei sicher www.samba.org und samba.sernet.de.

Vorschau

Mit diesem „Tanzkurs" haben wir die Box nun auch das Servieren von Files gelehrt. Langsam aber sicher ist die WCM-Linux-Box zum festen Bestandteil eines Netzwerkes geworden. Da es nun immer mehr Dienste auf der Box gibt und die Clients entsprechend konfiguriert werden müssen, wollen wir im nächsten Workshop eine automatische IP-Adressverteilung und Einstellungsverteilung einrichten. Zum Großteil entfällt dann das lästige Konfigurieren der Clients. ■