# Linux Unterstützung in Schulungssystemen

Marek Walther



## Inhaltsverzeichnis



# Situation an Schulen

In den meisten Fällen ist die Ausstattung von Schulungssystemen nicht homogen, das bedeutet, dass die Arbeitsplatzsysteme einzeln installiert und eingerichtet werden müssen. Die Gründe liegen hierfür oft bei einer uneinheitlichen Hardware-, oder Softwareausstattung, aber auch Lizenzgründe können hier eine Rolle spielen. Hardware und Software haben oft schon einige Jahre auf dem Buckel, und als Betriebssystem kommt oft Windows 98 zum Zuge. Die Systeme werden meistens von ehrenamtlichen Hilfskräften oder Fachlehrern betreut, die bei etwas Glück hierfür ein paar Freistunden bekommen. Bei der Nutzung der Systeme von anderen Fachlehreren fährt meistens die Angst mit etwas kaputt zu machen, oder diese in einem nicht funktionsfähigem Zustand vorzufinden. Fachlehrer anderer Bereiche trauen sich meistens gar nicht die Rechner überhaupt in ihren Unterricht mit einzubauen.

PC-Systeme an Schulen sind zumeist schlecht gesichert, hier wird oft versucht diesen Mangel mit der Einführung von Restriktionen auszubügeln, was die Situation noch verschlimmert. Auch der längere Ausfall eines Arbeitsplatzes muss hier gelegentlich einmal hingenommen werden. Netzwerke dienen, sofern voranden, nur dem allgemeinem Zugang zum Internet, und Email Konten für den Unterricht werden bei Freemailern eingerichtet. Die Situation ist damit für die meisten Schulen schwierig, und die System nur für einen beschränktes Einsatzgebiet ausgestattet.

# Netzwerkunterstützung bei der Administration

## Gründe für das Sichern von Arbeitsplätzten

Das Aufsetzten eines Systemes ist meistens mit einem hohen Zeitaufwand verbunden, Einspielen von Service-Packs und die individuelle Einrichtung des Systems nehmen viel Zeit in Anspruch. Glücklich kann hier sein wer, aus Hardware und Lizenssicht, in der Lage ist ein aufgesetztes System auf andere Arbeitsplätze zu duplizieren

(klonen). Deshalb ist es aus ökonomischer Sicht sinnvoll, fertige Systeme zu sicher und diese bei Bedarf wieder auf die Rechner zurückgespielen, um ein jungfreuliches System wie nach einer Installation zu erhalten.

Wenn es die Lizenzpolitik erlaut, ist es möglich eine rekursive Installationsherachie aufzubauen. Hierbei wird eine Basisinstallation erstellt, die alle notwendigen Hardwaretreiber und Grundeinstellungen enthält. Für unterschiedliche Hardwareausstattungen werden Hardwareprofile angelegt, und alle notwendigen Scripte und Datenbanken werden integriert. Diese Installation dient jetzt als Basis für unterschiedliche Installationen, und kann danach mittels klonen der Installation an die Rechner verteilt werden. Mit dieser Technik ist eine Wartung der Installationen auch wesentlich einfacher durchzuführen, Updates und SP´s können zentral von der Administrative auf die Installation eingespielt und danach auf alle betreffenden Systeme verteilt werden.

# Probleme unterschiedlicher Filesystem

DOS und W9x Systeme arbeiten mit einem FAT Dateisystem, welches bekannt und auch gut dokumentiert ist. Aus diesem Grunde gibt es für dieses Filesystem auch die meiste Unterstützung, allerdings hat es auch einige gravierende Nachteile die eine Sicherung erst notwendig machen. FAT unterstützt keine Dateirechte, aus diesem Grund kann jeder der auf dem System arbeitet dieses auch verändern und beschädigen. Hierbei kann es sich um eine mutwillige, aber auch um eine versehentliche Beschädigung handeln. NT, W2000 und XP verfügen über ein NTFS Filesystem, wobei hier die Versionen und verfügbaren Funktionen des Dateisystemes variieren. Dieses Filesystem wird vom Hersteller Microsoft mehr oder weniger geheim gehalten. Die Folge ist, dass es kaum Tools zur Bearbeitung und sinnvollen Sicherung solcher Systeme gibt, da derzeitig nur MS Betriebssysteme dieses Dateisystem sicher bearbeiten können. Systeme der NT-Schiene besitzen intern eine System-ID, diese muss auf allen System unterschiedlich sein, was durch ein Klonen nicht zu ist. Zur Lösung des Problemes hat Microsoft ein Tool zur Verfügung gestellt, mit dem es möglich ist die SID auf den Arbeitsplatzrechnern zu ändern Nach dem clonen ist es notwendig jeden W2000 und XP Arbeitsplatz noch einmal von Hand anzufassen, um den Rechner in eine eventuell bestehende Domain einzugliedern oder weitere Netzwerk-Einstellungen durchzuführen. Man kann ruhig sagen, dass das Aufsetzten eines W2000 oder XP Systems wesentlich aufwendiger ist als ein W9x System. Allerdings darf man auch nicht vergessen, das Aufgrund der Nutzerrechte dieses wesentlich länger hält. Eine Empfehlung einiger Klone-Tool Hersteller, sein System mit FAT32 zu installieren, um die Funktionen der Software nutzen zu können, sollte man deshalb kritisch betrachten. Filesysteme unter Linux sind dank Open-Source offen, und bringen von Hause aus genügend Tools zur Sicherung und Wartung mit. Linux kennt ebenfalls Nutzerrechte, und steht in diesem Punkt sicherheitstechnisch nicht hinter dem Microsoft Betriebssystemen zurück. Es ist aufgrund seiner offenen Struktur leichter zu pflegen, und lässt sich durch Scripte am besten automatisieren.

# Unterschiedliche Sicherungsmethoden und Arten

Aufgrund der unterschiedlichen Eigenschaften der vorliegenden Dateisysteme können auch unterschiedliche Sicherungsmethoden verwendet werden. Hier unterscheide ich zwischen Dateisystemen die auf Dateibasis gesichert werden können, und Systemen die geometrieabhängig gesichert werden müssen. Bei einer Sicherung auf Dateibasis wird keine Information des darunter liegenden Dateisystemes mitgesichert, sondern nur die reinen Dateinformationen. Kandidaten für eine derartige Sicherung sind alle Unix/Linux Dateisysteme, und das DOS und Windows basierende FAT16. Der Vorteil ist, dass beim Recovern auch die Größe des Laufwerkes geändert werden kann. Das Bedeutet aber, dass das Laufwerk neu Formatiert, und ggf. auch noch ein Bootloader neu installiert werden muss. Hier bleibt für FAT16 nur die DOS Bootdiskette, und der Befehl sys c: um einen Loader auf die Partition zu installieren. Unter Linux wird meistens der Linux Loader LILO eingesetzt, dieser kann auch von dem Gast Linuxsystem, mit dem das System recovert wurde, wieder initialisiert werden.

+++

Das Sichern/Recovern eines Linuxsystems auf Dateibasis sieht analog aus, allerdings werden hier andere Filesysteme auf den Partitionen verwendet. FAT32 ist ein aufgebohrtes FAT16 System, die Unterschiede liegen hier in der verwaltbaren Größe, und in der Möglichkeit lange Dateinamen zu verwenden. Die langen Dateinamen können sich hierbei als Problem erweisen, da alle Dateien mit ihrem langem Dateinamen gesichert werden, ist die Wahl des kurzen Namens beim Recovern willkürlich. Das wird zu Problemen bei Programmen führen, die auf die kurzen Namen angewiesen sind. NTFS Dateisystem können derzeitig unter Linux nur als Partitionsimage gesichert,

und recovert werden. Diese Möglichkeit besteht aber auch mit allen anderen Systemen, hierbei kann keine Veränderung der Laufwerksgrösse durchgeführt werden. Ein Recovern ist auch nur möglich, wenn die Ziel-Partition mindestens genau so gross ist wie die Quell-Partition. Images von Partitionen lassen sich mit 'dd' erstellen, oder zurückspielen. 'dd' kopiert die gesamte Partition in die entsprechende Datei, hierbei werden auch nicht verwendete Blöcke mit kopiert. Aus diesem Grund sind die Images immer relativ groß, und auch im komprimiertem Zustand umständlich zu händeln. Die eben aufgeführten Methoden sind für viele relativ schwierige Komandozeilenbefehle, und setzen aucheiniges an Kenntnissen über die verwendeten Techniken voraus. Einfacher lässt sich die Sicherung über ein Tool lösen, welches über ein Benutzerinterface bedient wird. Ein entsprechendes Tool ist auf der aktuellen Knoppix CD verfügbar, es heißt partimage , und kann aus der Komandozeile über Dialog bedient werden. Es unterstützt die Linux Filesysteme ReiserFS, ext2, ext3 und die Windowsdateisystem FAT16 und FAT32. NTFS wird hier als ;experimental support; geführt, dieses wird sich warscheinlich solange nicht ändern, wie Microsoft die technischen Details des Filesystemes unter Verschluss hält. Partimage speichert nur die benutzten Blöcke einer Partition, kann aber derzeitig noch nicht zur Änderung von Partitionsgrössen verwendet werden. Die Daten werden komprimiert, und auf Wunsch kann die Sicherung auch auf mehrere Volumen mit einer festlegbaren Größe gesichert werden.

Ich habe hier jetzt mehrere Sicherungsmethoden aufgezeigt, diese lassen sich auf mindestens 2 Arten anwenden. Zum einen besteht die Möglichkeit auf dem System lokal eine Partition anzulegen, in dem die Sicherung abgelegt wird. Auf diese Weise kann eine schnelle Rücksicherung mit einem Gastsystem wie zum Beispiel Knoppix erfolgen, und es werden keine Netzwerkresourcen belegt. Da diese Partition immer im direktem Zugriff des Benutzers steht, besteht hier auch immer die Gefahr einer Beschädigung oder Löschung der Daten. Ein einfaches Spielen mit fdisk unter DOS reicht dafür schon aus, auch sind die Daten nicht vor einem defekt der Festplatte sicher. Eine andere Möglichkeit besteht darin, die Sicherungsdaten auf eine Server auszulagern. Dadurch sind diese vor Beschädigungen relativ sicher, und können ggf. einer Datensicherung zugeführt werden. Wer die ersten beiden Methoden tar und dd bevorzugt, kann mit dem Gastsystem ein Netzlaufwerk mounten. Als Server könnten hier NFS oder ein mit Samba aufgebauter SMB Server dienen. Partimage bringt hierfür seinen eigenen Server mit, dieser sollte auf allen aktuellen Distributionen verfügbar sein. Der Dienst kann dann mit auf dem Fileserver eingerichtet werden, auf diesen kann der Partimage Client zugreifen, und die Images lesen und schreiben. Auf diese Weise stehen die Images Systemweit zur Verfügung, allerdings dauert das Aufspielen einer Sicherung dadurch länger, und es werden erhebliche Netzresourcen belegt.

# Fileserver für administrative Aufgaben

Jeder der Computersysteme verwaltet kennt das Problem, wenn zum Beispiel eine Hardwarekomponente von Rechner A zu Rechner B wechselt, oder mal eben ältere Treiber für eine Grafikkarte benötigt werden. Jeder Administrator ist ein Jäger und Sammler, denn auch der älteste Treiber oder die ungewöhlichste Boot-Diskette kann einem schon mal das Wochenende retten. Eine Möglichkeit besteht darin alles auf CD zu brennen und akribisch zu beschriften, eine andere Treiber, Tools und Software systemweit über einen Fileserver zur Verfügung zu haben. Der Server kann seinen Dienst hier über SMB mit Samba, über eine NFS-Freigabe oder über einen FTP Zugriff zur Verfügung stellen. Der Zugriff sollte über ein Passwort geschützt sein, um sicherzustellen das keine Daten lizenzwidrig entfläuchen. Mit einem gut gewartetem Server für Treiber ist es ein leichtes mal eben eine neue Hardwarekomponente zu installieren, oder ältere Treiber aufzuspielen.

# Automatisches verteilen von Updates

Das neue Semester hat begonnen, und man hat es gerade so geschafft alle Systeme einzurichten. Da kommt am Nachmittags der Dozent X vorbei, und merkt an, das ihm auf dem Rechner das Tool Y fehlt, ohne das er nächste Woche natürlich keinen Unterricht machen kann. Schöner Mist, die Arbeit der letzten 3 Tage zum Teufel, und da man ja nur ab Samstag Mittag die Rechner stillegen kann ist das Wochende auch noch versaut.

Wer vorgebaut hat ist in solchen Situationen fein raus, viele Updates lassen sich auch über ein Netzwerk erledigen und sind so im laufendem Betrieb möglich. Ein Server stellt hier die Updates zur Verfügung, der Client prüft diese beim Hochfahren ab, und spielt sie ein wenn es notwendig ist. Linux-Clients sind hier durch ihr offenes System sehr stark im Vorteil, die leistungsfähigen Scriptfähigkeiten erweitern hier die Möglichkeiten. Bei Windowssystem ist das ganze schon wesentlich schwieriger, häufig muss hier das Update im zwei-pass Verfahren durchgeführt werden. Das bedeutet, es sind mindestens 2 Systemstarts notwendig um das Update einzuspielen.

Der Ablauf des Updates ist bei allen System gleich, allerdings unterscheidet sich der Aufruf und die Abarbeitung je nach System. Zum Beginn muss ein Netzlaufwerk eingebunden werden, und von diesem eine Indexliste aller zur Verfügung stehenden Updates geladen werden. Anhand der Indexliste kann der Client ermitteln welche Pakete er benötigt, und welche er verwerfen kann da er diese schon installiert hat. Danach können die Pakete geladen, installiert, ein Updatestatus gespeichert, das Netzlaufwerk getrennt und ein Protokoll geschrieben werden. Auf diese Weise können Einzeldateien oder Programmpakete ausgetauscht werden. Handelt es sich bei dem Client nicht um ein W9x System, ist auf ein korrektes setzen der Benutzerrechte zu achten.

Für Einzeldateien und Pakete kann in der Indexliste eine Prüfsumme mit übergeben werden, der Client kann so, ohne die Datei zu laden, prüfen ob diese nicht schon aktuell ist. Da jetzt nicht jedesmal beim Starten 20 Clients 30 Pakete laden, die sie danach eh wieder verwerfen, werden hier Bandbreite auf dem Netzwerk und Resourcen beim Server eingespart. Bei einigen Systemen ist es nicht möglich Dateien auszutauschen die gerade verwendet werden, so kann z.B. die ’system.dat’ einer Windows 98 Installation nicht im laufendem Betrieb ausgetausch werden. Solche Dateien müssen dann in zwei Durchgängen ausgetauscht werden. Pass eins, die Dateien werden normal geladen, zwischengespeichert und das System wird neu gestartet. Pass zwei, zu einem sehr frühem Zeitpunkt (autoexec.bat) werden die lokal gespeicherten Dateien eingespielt. Hierbei ist darauf zu achten, dass keine Endlosschleife entsteht, und das System nur noch am booten ist. Der Update-Vorgang lässt sich bei W9x Clients mit in das netlogon Script einbinden, so ist sichergestellt das eine bestehende Serververbindung vorhanden ist. Zum Schreiben komplexer Scripte und Funktionen sind die Batch-Funktionen von Microsoft leider nicht geeignet. Für diesen Zweck bietet sich GNU/AWK für DOS an, damit stehen neben den allgemeinen Erleichterungen auch Feinheiten wie reguläre Ausdrücke und assoziative Arrays zur Verfügung.

Bei Linux-Systemen kann der Prozess mit in den Init-Ablauf eingebunden werden, direkt nach dem initialisieren des Netzwerkes ist ein Zugriff auf Netzwerkresourcen möglich. Unter Linux stehen auch komplette Paketformate wie .rpm und .dep zur Verfügung, diese bieten schon von Hause aus eine weitreichende Unterstützung für Updates. Pakete im .dep Format können von Hause aus über Netzwerk auf den neusten Stand gebracht werden, auch können hier Server für Updates festgelegt, oder eigene Server genutzt werden. Gerade die letzte Option ist für eine lokale Administration wichtig, mit ihr kann der Administrator Updates lokal für seine Systeme zusammenstellen und diese so unter Kontrolle halten.

# Vorteile beim Unterricht durch Vernetzung

## Zentrale Unterrichtsarchive

Die im Unterricht benötigten Materialien und Vorlagen können zentral auf einem Fileserver verwaltet und bereitgehalten werden, sie stehen dann durch die Vernetzung im gesamten Unterrichtsraum zur Verfügung. Dadurch kann Arbeitszeit gespart werden, da eine individuelle Verteilung an die Teilnehmer nicht mehr notwendig ist. Auch der bei einer lokalen Vorhaltung der Archive, auf den Clients, benötigte Speicherplatz kann eingespart werden, und die Pflege der Archive kann Zentral erfolgen.

## Verteilen von Unterrichtsmaterial

Die "eigenen Dateien" der Benutzer können auf ein Serververzeichniss ausgelagert werden. Auf dem Server liegen diese sicher, und man muss sich beim Wiederherstellen einer Installation keine Gedanken über diese Daten machen. Bei anonymen Benutzern können diese Verzeichnisse dem Dozenten zugänglich gemacht werden, auf diese Weise kann der Dozent individuell eingreifen, und dem Teilnehmer eine Hilfestellung geben. Durch die zugänglichen Verzeichnisse ist es auch möglich individuelles Material an die Teilnehmer zu verteilen oder Ergebnisse einzusammeln.

## Gemeinsames Nutzen von Hardwareressourcen

Hardware ist teuer, und Geld meistens knapp. Ein Netzwerk ermöglicht es die Hardwarerressourcen in einem Unterrichtsraum gemeinsam einzusetzen, oft ist es nur so möglich überhaupt an die Anschaffung eines Gerätes zu denken. Das klassische Beispiel ist hier der Drucker, dieser kann an einen Arbeitsplatz angeschlossen und von den anderen Benutzern genutzt werden. Sollte es allerdings unterschiedliche Drucker geben, ist es notwendig die unterschiedlichen Treiber zu installieren, und bei einem Wechsel des Gerätes an allen Arbeitsplätzten die Treiber

zu tauschen. Hier besteht wieder die Möglichkeit sich mit einem Samba Druckserver die Arbeit zu erleichtern. Auf dem Server wird der Drucker lokal eingerichtet und getestet, danach wird mit Samba eine Drucker Freigabe erzeugt. Wir wollen aber die Druckverarbeitung vom Server durchführen lassen, die Daten werden im Postscript Format empfangen und für den Drucker lokal auf dem Server weiterverarbeitet. Als Druckertreiber muss auf den Arbeitsplätzen dann ein Postscript Treiber wie der "Appel Laserwriter" installiert werden. Sollte sich dann später der Drucker ändern, braucht er nur noch lokal auf dem Server angepasst zu werden. Es ist aber zu beachten, das die Umsetzung von Postscript mit Hilfe des Ghostscipt Paketes erfolgt, dieses benötigt dafür aber mehr Zeit und auch mehr Speicher als auf den Arbeitsplätzen nötig wäre. Als Drucker sollte im Schulungsbereich nicht auch billige Tintenstrahldrucker zurückgegriffen werden, da die Verbrauchskosten und der Wegwerfanteil im Schulbetrieb zu hoch sind. Sinnvoll wären Laserdrucker mit gekapseltem Papiervorrat, diese sind günstig im Verbrauch und Robust in der Handhabung.

Wenn man denn schon einen Rechner für die Druckverarbeitung im Unterrichtsraum stehen hat, kann dieser auch gleich für die Nutzung des Scanners eingesetzt werden. Der Scanner wird wie der Drucker auf dem Server lokal installiert, und über einen Dienst allen anderen Nutzern zur Verfügung gestellt. Auf den Arbeitsplätzen muss dafür ein Client installiert werden, über den der Scanner angesprochen werden kann. Da immer nur eine Person zur Zeit einen Scannvorgang durchführen kann, ist hier etwas Disziplin bei den Teilnehmern notwendig. Auf dem Server eignet sich hierfür SANE, der Standard beim Thema scannen unter Linux. Auf der Site des Projektes kann man erkennen das SANE eine grosse Anzahl an Geräten unterstützt. Vor dem Kauf eines neuen Scanners, sollte man hier einmal einen Blick draufwerfen. Der Scannvorgang wird in 2 Bereich unterteilt. Über das Backend wird der Zugriff auf die Hardware durchgeführt, es bildet den Treiber für das verwendetet Gerät. Zusätzlich gibt es auch noch Meta-Backends, über diese ist es zum Beispiel möglich einen einen Netzwerkscanner anzusprechen. Das Frontend bildet die Schnittstelle zum Benutzer, dieses wird aber auf dem Server meisens nicht benötigt, außer man möchte von hier direkt scannen können. Das Herzstück für den Netzwerkeinsatz ist aber der Dienst Saned, der den Scanner nach außen zur Verfügung stellt. Dieser Dienst, der für einen lokalen Geräteeinsatz nicht benötigt wird, wird über den Inetd Demon geladen, und muss deshalb auch in der Datei /etc/inetd.conf wie folgt eingerichtet werden.

```
sane stream tcp nowait saned.saned /usr/......../saned saned
```

Wie in der Konfigurationszeile zu sehen ist, sollte für den Betrieb ein Benutzer (saned) und eine Gruppe (saned) zur Nutzung eingericht werden. Die Gruppe (saned) sollte auch einen vollen Zugriff auf das verwendete Geräte im Geräteverzeichniss /dev bekommen, um dieses Gerät nutzen zu können. Ebenfalls ist in zu erkennen, das wir den Port des Dienstes über einen Alias ansprechen, dieser muss eventuell in der Datei /etc/services nachgetragen werden.

```
sane     6566/tcp   saned #; SANE network scanner deamon
```

Jetzt können wir daran gehen das Backend von SANE zu konfigurieren, die Dateien dafür befinden sich meistens unter /etc/sane.d oder /usr/local/etc/saned.d. Die Dateien dll.conf und net.conf sind Meta-Backends, über dll.conf wird festgelegt mit welchen Backends SANE arbeiten soll. In net.conf können Netzwerkscanner angegeben werden, die eingebunden werden sollen. Alle anderen Dateien sind zur Konfiguration der Backends verschiedener Geräte vorgesehen, hier schlagen Sie für die Konfiguration am besten auf der SANE Site nach. Falls es notwendig ist, kann der Zugriff auf das Gerät auch noch durch einen Benutzer- und Passwortschutz gesichert werden. Benutzer, Passwörter und erlaubte Backends werden hierbei in die Datei saned.user mit folgendem Schema eingetragen:

```
user:password:backend
```

Eine weitere Datei die den externen Zugriff auf den Dienst regelt ist die Datei saned.conf. Hier können die Hosts mit Namen oder IP-Adresse eingetragen werden, von denen ein Zugriff erlaubt ist. Es ist aber darauf zu achten, dass auch ein Reverse-Lookup der Hostnamen möglich ist, da sonst der Zugriff verweigert wird. Bei DNS Problemen kann das Resolving lokal erfolgen, hierfür müssen die die Hosts in der Datei /etc/hosts eingetragen

werden. Wenn keine Zugriffsbeschränkung auf Netzbasis gewüscht ist, ist in der Datei saned.conf nur eine Zeile mit einem "+" einzutragen. Als letztes kann geprüft werden welche Scanner zur Verfügung stehen.

```
$ scanimage -L
```

Die Einrichtung des Netzwerkscanners auf Arbeitsplatzrechner mit einem Linux Betriebssystem funktioniert analog, hier kann aber die Konfiguration des saned Dienstes ausgelassen werden, da der Scanner ja nicht noch einmal freigegeben werden soll. Als Backend ist das net.conf Backend zu konfigurieren, die Verfügbarkeit von Scannern kann dann wieder mit scanimage geprüft werden. Bei Problemen sollte man auch mal einen Blick auf die Log Dateien des Servers werfen.

Für Windows stehen verschiedene Clients zur Verfügung, hierbei sind auch erste Clients mit TWAIN Schnittstelle, die ein gewohntes arbeiten mit dem Gerät ermöglichen, verfügbar. Zu Empfehlen wäre der XSane Client, hier müssen die eingescannten Objekte zwar noch als Datei gespeichert werden, aber er ist einfach zu bedienen und ist sehr ausgereift. Die Installation von XSane ist sehr einfach, das Archiv wird nach c:\ entpackt, hierbei wird ein Unterverzeichniss sane angelegt in dem sich das gesamte Paket befindet. Konfiguriert wird das ganze über die Datei c:\sane\etc\sane.d\; net.conf, hier ist die IP Adresse des Servers anzugeben über den der Scanner verfügbar ist. Nach dem Aufruf von c:\sane\Xsane.exe wird der Scanner gesucht, und das GUI baut sich auf.

# Interne Netzwerkdienste

An vielen Schulen werden im Netzwerk keine eigenen Dienste angeboten, das Netwerk dient häufig nur dazu die Rechner über einen Instant-Router mit dem Internet zu verbinden. Es gibt aber Dienste, die bei der Administration Unterstützung bieten oder die benötigte Bandbreite nach außen reduzieren.

## DHCP und DNS die unsichtbaren Helfer

DHCP steht für "Dynamic Host Configuration Protokol" und kann helfen die Netzwerkkonfiguration in den Unterrichtsräumen zu vereinfachen. Hierbei wartet ein Dienst im Netzwerk darauf, dass sich ein neuer Client im Netzwerk meldet und nach Konfigurationsdaten fragt. Daraufhin wird dem Client ein Datensatz mit Konfigurationsinformationen zugestellt, die er zum Einrichten seines Netzwerkes verwenden soll. Es können folgende Informationen enthalten sein:

- Eine freie IP-Nummer die der Client verwenden soll.
- Die Subnetmask die in dem Subnet gültig ist in dem sich der Client befindet.
- Die Adresse eines gültigen Nameservers.
- Die Adresse des gültigen Gateways an den der Client Pakete senden soll, die er nicht direkt ausliefern kann.
- Die Adresse eines gültigen WINS Servers der als gültiger Masterbrowser für Windows Clients dient

Alle diese Informationen werden dem Client dynamisch mitgeteilt, dadurch wird das statisches Eintragen der Werte auf den Arbeitsplätzen vermieden. Sollten sich diese Werte einmal ändern, muss so nicht mehr jeder Arbeitsplatz von Hand umconfiguriert werden. Die Änderungen werden einmalig auf dem Server durchgeführt und stehen den Clients beim nächstem Start zur Verfügung. Es gibt hier vielfältige Möglichkeiten den Dienst zu konfigurieren, ich möchte hier den einfachsten Weg für eine Konfiguration mit einer dynamischen IP-Range beschreiben. Der Dienst ist für alle Distributionen verfügbar und wird meistens mitgeliefert. Wenn er auf einem Server installiert ist, ist die Datei /etc/dhcp.conf für die Konfiguration zuständig.

```
# /etc/dhcpd.conf
# Allg. Konfigurationsbeispiel für eine einfache Arbeitsumgebung
#
# Def. globaler Werte
default-lease-time 10800; # Die Standard-Adreßzuweisung soll 3 Stunden
                   # (10800 Sekunden) betragen
max-lease-time    86400; # Maximal gilt die Zuweisung für 1 Tag
                   # (86400 Sekunden)
get-lease-hostname false; # Der Server soll die Clients nicht mit
                   # einem Hostnamen versorgen
# Def. globaler Optionen
# Festlegen von Netzwerkmaske und Broadcast die global
# verwendet werden soll (Wenn nicht später anders angegeben)
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.101.255
# Festlegen der Domain
option domain "meineschule.local"
# Festlegen der gültigen DNS-Server im Netzwerk
option domain-name-servers 192.168.101.16, 192.168.101.17
#
#-------------------------------------------------------------
# Festlegung der IP Adressen im Arbeitsnetzwerk
# 000, 255  -> NC
# 001-015   -> Gateways, Router, Switches, Hubs
# 016-031   -> Server mit Diensten, File, Print, Time, .....
# 032-063   -> NC
# 064-127   -> IP-Vergabe von Hand
# 128-254   -> IP-Vergabe per DHCP
#-------------------------------------------------------------
#
subnet 192.168.101.0 netmask 255.255.255.0 {
  option routers 192.168.101.5;
  option broadcast-address 192.168.101.255;
  range 192.168.101.128 192.168.101.254;
}
```

Bei diesem Beispiel verwaltet der DHCP-Dienst die Hostadressen 128-254 in einem vorhandenem Subnet. Die Clients können eine Adresse für max. 3 Stunden anforden, nach Ablauf dieser Zeit muss die Adresse erneut abgefragt und bestätigt werden. Der Server vergibt eine Adresse maximal für 24 Stunden an einen Client. Die Adressen aller weiterer Dienste und Optionen wie DNS-Server, Netzwerkmasken, Broadcastadressen und Domain Name sind hier vermerkt und werden den Clients mitgeteilt. Die meisten Optionen wurden global angegeben, da sie sich im Verlauf der gesamten Konfiguration nicht ändern. Abweichende Einstellungen können innerhalb der Konfigurationsblöcke geändert werden.

Ein weiteres Hilfsmittel ist die Einrichtung eines Domain Name Service zur Auflösung von Hostnamen zu IP-Adressen. Meistens werden Mailsserver bei Outlook oder der Webproxy mit einer IP eingetragen, ändert sich diese aus irgend einem Grund, muss man seine Turnschuhe einpacken und alle Arbeitsplätze aufsuchen. Außerdem sind Namen wie pop3.meineschule.local, smtp.meineschule.local oder proxy.meineschule.local wesentlich aussagekräftiger als reine IP-Adressen, und helfen mit ein System skalierbar zu halten. Alle Anfragen zwecks einer Namensauflösung gehen an den internen DNS-Server, dieser versucht den Namen aufzulösen, oder die Antwort über eine Anfrage bei einem externem DNS-Server zu beschaffen. Für alle Antworten ist eine Zeit definiert wie lange diese gültig ist, der lokale DNS merkt sich die Antwort über diesen Zeitraum und gibt sie bei

einer gleichen Anfrage wieder heraus. Dadurch werden Resourcen und Bandbreite von Providern und externen Dienstanbietern geschont, und auch die eigenen Nutzer haben davon Vorteile. Vor dem Einrichten muss eine DNS-Domain gewählt werden, hierbei ist darauf zu achten, dass die verwendete Toplevel-Domain (.local) keine offizielle Toplevel-Domain ist. Die Verwendung einer offiziellen Toplevel-Domain könnte zu Unstimmigkeiten bei der Namensauflösung führen, und weitreichende Maßnahmen bei der Einrichtung anderer Dienste notwendig machen.

Die Einrichtung des DNS-Servers BIND erfolgt in mehreren Dateien, zum Einem die Konfigurationsdatei /etc/named.conf, zum Anderen in den Zonen- und Datenbankdateien mit den DNS Informationen.

```
#/etc/named.conf
# Einfaches Beispiel für eine BIND8 konfiguration
options {
      directory "/var/named";
      check-names master warn;
      pid-file "/var/run/named.pid";
      datasize default;
      coresize default;
      files unlimited;
      multiple-cname no;

# Der Server soll nicht selber externe Anfragen auflösen;
# sondern einen externen DNS dafür verwenden.
      forward only
      forwarders {
              212.185.254.170;
              212.185.253.70;
              };
#[.....]; ACL und Logging wurde ausgelassen
}
zone "." IN {
       type hint;
       file "root.hint";
};
zone "localhost" IN {
       type master;
       file "localhost.zone";
       check-names fail;
       allow-update { none;};
};
zone "0.0.127.in-addr.arpa" IN {
       type master;
       file "127.0.0.zone";
       check-names fail;
       allow-updates {none;};
};
zone "meineschule.local" IN {
       type master;
       file "meineschule.local.zone"
       check-names fail;
       allow-updates {none;};
};
zone "100.168.192.in-addr.arpa" IN {
       type master;
       file "meineschule.local.rev"
       check-name fail;
       allow-updates {none;};
};
```

Eine wichtige Einstellung ist hier die Angabe wo sich die Zonen-Dateien befinden (/var/named), und die Option forward only. Durch diese Option wird BIND angewiesen, für Namen die er nicht kennt, einen externen Nameserver als Forwarder zu benutzen. Diese sind in der Liste forwarder{} einzutragen, am besten nimmt

man hier 2 Nameserver seines Providers und 2 offizielle .DE Server. Ohne diese Option würde BIND selber versuchen den Namen aufzulösen, er würde dabei bei den Root-Servern anfangen und sich bis zum zuständigem Domainenserver hocharbeiten. Dieses stellt eine unnötige Verzögerung da, da die Server vom Provider einen größeren Cache und eine besser Netzanbindung besitzt, als unser lokaler Nameserver. Als nächsten kommt die Konfigurationen der Zonen. Die Zone "." ist die Root-Zone, sie zeigt auf eine Datei in der die 15 Root-Server verzeichnet sind. Diese Server wären wichtig wenn unser Nameserver selber eine Auflösung der externen Namen durchführen würde, aber auch wenn wir sie nicht benötigen, sollte man diese Datei gelegentlich updaten. Die Zone "localhost" ist eine Auflösung des loopback Interfaces, diese Dateien sollten in der Regel schon existieren und müssen ggf. nur angepasst werden. Bei der Zone "0.0.127.in-addr.arpa" handelt es sich um ein Reverse-Lookup für die Zone "localhost". Reverse-Loockup bedeutet, dass auf eine IP-Adresse ein Hostname gesucht wird. Viele Dienste führen ein Reverse-Lookup des Clients durch, der auf diesen zugreifen möchte. Fällt die Anfrage negativ oder unerwartet aus, verweigert der Dienst seine Arbeit. Bei einem sauber konfigurierten Netzwerk sollte deshalb immer ein Reverse-Lookup möglich sein. Die beiden letzten Zonen stellen unsere eigene Zone für die Domain "meineschule.local" und das dazugehörige Reverse-Lookup da. Der jeweiligen Zone wird mittels "file" mitgeteilt, in welche Datei sich die Datenbank für diese Zone befindet. Alle Dateien liegen realtiv zum Datenpfad "directory". Die Datei für die eigene Zone wurde zur besseren Wartbarkeit in mehrere Dateien aufgeteilt.

```
;/var/named/meineschule.local.zone
@  in SOA dns1.meineschule.local admin.meineschule.local (
          10110   ; Seriennummer der Datei, bei Änderung erhöhen.
          43200   ; Zeit bis zum nächsten Zonen refresh, für
                 ; einen sekundären DNS
          3600    ; Zeit für einen erneuten Versuch eines
                 ; fehlgeschlagenen Zonenrefreshs.
          360000  ; So lange darf ein sekundärer Nameserver auch
                 ; ohne gelungenem Zonenrefresh die Daten
                 ; weiterferwenden.
          250000 ) ; TTL - Lebensdauer der Antworten
include NAMEuMAIL.meineschule.local
include DIENSTE.meineschule.local.zone
include DHCP-CLIENTS.meineschule.local.zone
```

Die Datei /var/named/meineschule.local.zone enthält am Anfang den SOA Record (Start of Authority), das @ reverenziert die Zonen-Domain, diese wird aus dem entsprechendem Zonenblock der /etc/named.conf Datei übernommen. Gefolgt von dem Hostnamen, der der Masterserver für diese Domain ist und der EMail-Adresse der verantwortlichen Person für den Server. Bei der Mailadresse wurde das "@" durch einen "." ersetzt. Die weiteren Parameter nehmen Einfluss auf das Verhalten der Domain.

Die Seriennummer sollte nach jeder Änderung der Zonen-Datei hochgezählt werden, damit ein sekundärer Nameserver diese Veränderung bemerkt. Die nächsten drei Werte sind bei der Verwendung eines sekundärenm Nameservers wichtig. Ein Sekundärer Nameserver erhält seinen Datenbestand durch einen Zonentransfer mit dem Masterserver, jeder Masterserver kann auch für eine andere Domain den sekundären Nameserver übernehmen. Der zweite Wert sagt, aus wieviel Zeit ins Sekunden zwischen den Transfers liegen soll. Sollte ein Transfer nicht geklappt haben, sagt der vierte Wert nach welchem Zeitablauf ein erneuter Versuch gestartet werden soll. Der fünfte Wert sagt aus, wie lange die Server die Daten ohne durchgeführtem Zonentransfer als gültig betrachten soll. Als letztes steht die allgemeine Lebensdauer (TTL) die jeder Antwort beigefügt wird, und dem Client sagt wie lange die Antwort Gültigkeit besitzt.

Die zugehörigen Elemente wurden per include der Datei hinzugefügt. In diesen werden die DNS-Daten in Records abgelegt, der Aufbau eines Records sieht wie folgt aus:

```
 [name][ttl] IN data
```

**Aufbau eines DNS-Records.**

- [Name]

  Name Ist der Name des Domain Objektes das angesprochen wird. Der Name wird relative zur Domaine betrachtet, und stellt einen Host oder Domainnamen da. Sollte der Name absolut betrachtet werden, muss er mit einem Punkt "." enden. Wird dieses Feld frei gelassen, wird der Record auf den letzten genannten Namen angewandt.

- [ttl]

  Steht für "Time-to-Life", und bezeichnet die Zeit die dieser Eintrag in einem Cache gehalten werden darf. Normalerweise wird dieser Wert für alle Records in der Datei über der SOA definiert.

- IN

  Definiert den Record als Internet DNS-RR Klasse. Es gibt weitere Klassen, die aber im Wildlife nicht vorkommen. type Art des Records.

- daten

  Für den Recordtyp typische Daten.

```
;/var/named/NAMEuMAIL.meineschule.local
; Festlegung von Name- und Mailserver für die Zone
            IN    NS   dns1.meineschule.local.
            IN    NS   dns2.meineschule.local.
            IN    MX   10 mail.meineschule.local.
```

In dem ersten Abschnitt legen wird die Name- und Mailserver für die Domain fest, über diese Einträge kann zum Beispiel ein Mail Transfer Agent (MTA) einen Mailserver für eine Mailauslieferung lokalisieren. Die ersten beiden Records sind Nameserver Records (NS), sie deklarieren die für diese Domain zuständigen Nameserver. Der letzte Record ist ein Mail-Exchange (MX) Record, er gibt an welche Mailserver für die Domain zuständig sind. In diesem Record wird eine Präferenzwert (10) eingetragen, mit diesem kann bei mehrern Mailservern festgelegt werden in welcher Reihenfolge die Server verwendet werden. Mit mehrern Servern kann die Mailauslieferung ausfallsicherer durchgeführt werden, je kleiner der Wert, um so besser ist der Server.

```
;/var/named/DIENSTE.meineschule.local.zone
; Festlegung von Diensten die im Netzwerk verfügbar sind.
; Alle Angaben erfolgen relativ zur Zone.
; ->; Unsere Server
merkur        IN    A    192.168.101.16
mars          IN    A    192.168.101.17
; -> Zuweisung der Dienste
; Domain Name Server
dns1          IN    CNAME    merkur
dns2          IN    CNAME    mars

; Mail und News Dienste
mail          IN    CNAME    mars
pop3          IN    CNAME    mars
imap          IN    CNAME    mars
smtp          IN    CNAME    mars
sntp          IN    CNAME    mars

; Webdienste
www           IN    CNAME    merkur
proxy         IN    CNAME    mars

; Dateidienste
ftp           IN    CNAME    merkur
smb           IN    CNAME    merkur
```

```
nfs         IN    CNAME    merkur
```

In diesem Abschnitt erfolgt die Deklaration unserer Server und Dienste. Unsere beiden Server sind Merkur und Mars, auf diesen verteilen sich alle im Netzwerk befindlichen Dienste. Die ersten beiden Records deklarieren unsere Server mit ihrer IP Adresse, es handelt sich hier um Adress Records (A). Hierbei ist zu beachten, dass die Namen keinen Punkt besitzten. Dadurch gehören sie relativ zur behandelten Zonen-Domain, und der volle Name wird mars.meineschule.local lauten. Als nächstes folgt die Deklaration der Dienste. Auch hier gibt es keine Punkte, weshalb auch diese Namen relativ zur Zonen-Domain stehen. Bei den Records handelt es sich um Aliase (CNAME), es werden einfach Verweise auf die bestehenden Server gelegt auf dem der Dienst läuft.

```
DHCP-CLIENTS.meineschule.local
; Festlegung der Namen des DHCP-Adressbereiches
; Diese Datei lässt sich einfach mit einem Skript aufbauen
client-128    IN    A      192.168.101.128
client-129    IN    A      192.168.101.129
[....];
client-254    IN    A      192.168.101.254
```

Im letzten Abschnitt werden für alle übrigen Hosts Hostnamen eingetragen. Da es sich hier um eine DHCP Vergabe handelt, werden diese einfach durchnummeriert.

Jetzt muss noch die Zone für das Reverse-Lookup der Zone "meineschule.local" eingerichtet werden. Diese lautet "100.168.192.in-addr.arpa" und befindet sich in der Datei meineschule.local.rev.

```
;/var/named/meineschule.local.rev
@  in SOA dns1.meineschule.local admin.meineschule.local (
          10110   ; Seriennummer der Datei, bei Änderung erhöhen.
          43200   ; Zeit bis zum nächsten Zonen refresh, für
                ; einen sekundären DNS
          3600    ; Zeit für einen erneuten Versuch eines
                ; fehlgeschlagenen Zonenrefreshs.
          360000  ; So lange darf ein sekundärer Nameserver auch
                ; ohne gelungenem Zonenrefresh die Daten
                ; weiterferwenden.
          250000 ) ; TTL - Lebensdauer der Antworten
            IN    NS       dns1.meineschule.local.
            IN    NS       dns2.meineschule.local.

include DIENSTE.meineschule.local.rev
include DHCP-CLIENTS.meineschule.local.rev
```

Der Aufbau entspricht der Datei meineschule.local.zone, allerdings wird diese Datei andere Recordtypen verwenden. Am Anfang der Datei steht wieder der SOA, gefolgt von den Namenserver der für die Zonen-Domain zuständig sind. Die Zonen-Domain ist bei dieser Datei "100.168.192.in-addr.arpa".

```
; /var/named/DIENSTE.meineschule.local.rev
16          IN    PTR      merkur.meineschule.local.
17          IN    PTR      mars.meineschule.local.
```

Bei Diensten müssen wir nur unsere beiden Server auflösen, die unsere Dienste beherbergen. Am Anfang eines Records steht wieder der Name, dieser wird, ohne Punkt am Ende, wieder um die Zonen-Domain ergänzt. Als Recordtyp wird ein Pointer verwendet (PTR), dieser sorgt für eine Umwandlung der Adresse zum Hostnamen. Als Daten enthalten die Records die zugehörigen Hostnamen, damit diese nicht erweitert werden wurden sie mit einem Punkt abgeschlossen.

```
;/var/named/DHCP-CLIENTS.meineschule.local.rev
128          IN     PTR      client-128.meineschule.local.
[...];
254          IN     PTR      client-254.meineschule.local.
```

Als letztes folgt noch in einer eigenen Datei die Auflösung der dynamischen Client-Adressen.

# Mail und Newsserver im eigenem Haus

...

# Proxy für Webzugriff

Ein Proxy steht bei der Auslieferung zwischen dem Client und dem Server, er vertritt hierbei gegenüber dem Server den Client. Dieses kann mehrere Vorteile haben, der Client braucht keinen direkten Zugriff auf das Internet, und der Proxy kann Seiten zwischenspeichern (cachen) und diese bei Bedarf noch einmal ausliefern. Durch den Cache-Mechanismus wird an derBandbreite nach außen gespart und der Zugriff auf Seiten im Internet beschleunigt. Zu Zeiten wo man noch mit ISDN oder Modem an das Internet angebunden war, war die Verwendung von Web-Proxys weit verbreitet. Heute, wo die Anbindung meistens mit DSL und einem Instant-Router über NAT durchgeführt wird, sieht man immer weniger Proxys an kleinen Standorten und in Schulen. Das ist eigentlich schade, da durch die Verwendung von Proxys die vorhandene Bandbreite besser und öknomiescher genutzt werden kann.

Unter Linux stehen verschiedene Proxys zur Verfügung, der Bekanntestet ist hierbei wohl der Squid (Tintenfisch). Squid ist ein ausgewachsener Tintenfisch, der Funktionsumfang ist sehr weitreichend, Syncronistationsmöglichkeiten mit anderen Proxys, Zugriffsbeschränkungen über ACL´s, diverse Möglichkeiten zum Tunen und das Einbinden von Modulen machen den Tintenfisch zu einem anspruchsvollem Gesellen. In den meisten Fällen werden die Möglichkeiten nicht voll ausgeschöpft.

Ein weiterer bekannter Proxy ist der WWWOFFLE, dieser Proxy ist im Gegensatz zu Squid in der Lage Webseiten auch offline auszuliefern. Das bedeutet, dass der Cacheinhalt auch ohne Aktivierung des Internetzuganges verfügbar ist. Mit diesem Proxy ist es auch möglich, ganze Sites automatisch in den Cache zu befördern und so z.B. den Unterrichtsstoff gezielt verfügbar zu machen. Aus den Seiten können auch gezielt animierte GIFs, javascript oder HTML Tags entfernt oder verändert werden. Trotz dieser Möglichkeiten, und weil Squid besser gepflegt wird, wird Squid wohl der Standard bei der Proxyanwendung bleiben.

Beim Einsatz von Squid sollte auf eine artgerechte Haltung des Tintenfisches geachtet werden, die alte Faustregel ein Proxy läuft schon auf einem 486´er stimmt zwar, aber er wird wenig Freude machen. Zur Verwendung von Squid in einer kleineren bis mittleren Umgebung sollte mindestens ein Pentium 200 Mhz zu verwendet, und in 128MB Hauptspeicher investiert werden. Die Cachedateie sollte nicht zu groß gewählt werden (<500MB), da Squid für jedes Objekt was es speichert einen Index erstellt. Dieser Index wird im Hauptspeicher gehalten, zusätzlich kommen noch Prozesse für die Abarbeitung der Anfragen und ggf. noch andere Programme. Sobald der Speicher knapp wird, fängt das Betriebssystem an diesen auszulagern, das Ergebnis ist dann ein Tintenfisch der den Webverkehr nach außen lahmlegt anstatt ihn zu beschleunigen. Wenn möglich sollte für den Cacheinhalt eine eigene Platte oder Partition gewählt werden, um das gesamte System nicht auszubremsen.

Ich möchte hier keine komplette Konfiguration von Squid vorstellen, die Einstellungen und Möglichkeiten würden einfach den Rahmen sprengen. Deshalb verweise ich hier auf des deutschsprachige Squid-Handbuch, es stellt wohl die umfassenste Referenz zu diesem Proxy da.

# geLinkt:

Links zu relevanten Informationen :

- Das Tool für 2000/XP heist SysPrep, und ist auf dem Installationsmedium unter Tools in der Datei DEPLOY.CAB enthalten. Oder unter : www.microsoft.com/windows2000/download/tools/sysprep/default.asp
- Die Heimatseite des Projektes Partimage : www.partimage.org
- Einsatz von Gnu/AWK unter verschiedenen PC-Betriebssystemen : www.gnu.org/manual/gawk-3.11/html_node/PC-Using.html
- Die Heimatseite des Sane Projektes : www.mostang.com/sane
- Die Heimatseite des Xsane Projektes : www.xsane.org
- Projektseite von Squid : www.squid-cache.org
- Projektseite von WWWOFFLE : www.gedanken.demon.co.uk/wwwoffle
- Handbuch für den Einsatz von Squid : www.squid-handbuch.de