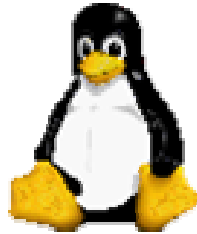


LINUX



WEITERNETSERVER

Die Hinweise in diesem Skriptum beziehen sich auf die Installation und Konfiguration der SuSE Distribution 8.1



<u>1</u>	<u>ZEITPLAN:</u>	4
<u>2</u>	<u>TIPPS & TRICKS</u>	5
2.1	X-WINDOWS:	5
2.2	ONLINE-UPDATE:	6
2.3	GRAFISCHER ZUGRIFF AUF RECHNER, DIE SSH-DIENSTE ANBIETEN:	6
2.4	CHKROOTKIT	6
<u>3</u>	<u>WEB-SERVER: (APACHE)</u>	7
3.1	GRUNDKONFIGURATION:	7
3.2	WEBSEITEN FÜR BENUTZER:	8
3.3	ZUGRIFF AUF BESTIMMTE WEBSEITEN MITTELS AUTHENTIFIZIERUNG:	8
3.3.1	LDAP:.....	9
3.3.2	MYSQL:	9
3.4	INTRANET	10
3.5	VIRTUELLE SERVER:.....	10
3.6	LOG-FILES	11
<u>4</u>	<u>FTP-SERVER:</u>	11
4.1	BIS SUSE 7.3:	11
4.2	AB SUSE 8.X	12
4.3	FTP-DIENSTE AM NOVELLSERVER:	14
<u>5</u>	<u>PROXY-SERVER:</u>	17
5.1	KASKADE DER PROXYSERVER:	18
5.2	FILTERN VON WEBSEITEN:	18
5.2.1	FILTERLÖSUNG DIREKT ÜBER SQUID:.....	18
5.2.2	FILTERN ÜBER EIN EXTERNES PROGRAMM (SQUIDGUARD):.....	19
5.3	TRANSPARENTER PROXY.....	21
<u>6</u>	<u>MAIL-SERVER:</u>	21
6.1	SMTP : (SENDMAIL – PORT 25)	21
6.1.1	DETAILS SMTP:	23
6.2	POP3: (PORT: 110)	24
6.3	IMAP: (PORT 143).....	24
6.4	VIRENSCANNER:	25
6.5	NACHTRAG ZU POP3 UND IMAP:.....	26



7	<u>CRON:</u>	26
8	<u>BENUTZERPLATZBESCHRÄNKUNG - QUOTAS</u>	26
9	<u>NIS UND NFS:</u>	27
10	<u>SQL-SERVER – MYSQL</u>	28
10.1	INSTALLATION UND KONFIGURATION	28
10.2	ZUGRIFF VON WINDOWS-WORKSTATIONS	28
10.2.1	EINFACHERE VERWALTUNG MIT PHPMYADMIN:	29
10.3	ZUGRIFFE AUS WEB-DOKUMENTEN	30
10.4	EIN BEISPIEL MIT PHP:	31



1 ZEITPLAN:

Mo. 7.4.2003	
09.30 – 10.15	Einrichten der Seminarkonfiguration, Tipps&Tricks
10.30 – 12.00	Webserver – Grundkonfiguration, FTP-Server, Benutzer für Wartung der Homepage einrichten
14.00 – 15.30	Erweiterte Konfigurationsmöglichkeiten: Gesicherte Bereiche, Benutzerhomepages
15.45 – 17.15	Virtuelle Server, LogFiles
Di. 8.4.2003	
08.45 – 10.15	Konfiguration des Proxyservers SQUID , Transparenter Proxy,
10.30 – 12.00	Filtern von Webseiten mit einfachen Textdateien bzw. mit Squidguard
14.00 – 15.30	Mailserver: Konfiguration von Sendmail, Aliases, Spamming, Konfiguration des POP Servers
15.45 – 17.15	Details zu SMTP und POP3
Mi. 9.4.2003	
08.45 – 10.15	CRON-Jobs, QUOTAS, MYSQL
10.30 – 12.00	NIS / NFS bzw. SAMBA
14.00 – 15.30	SAMBA



2 TIPPS & TRICKS

2.1 X-WINDOWS:

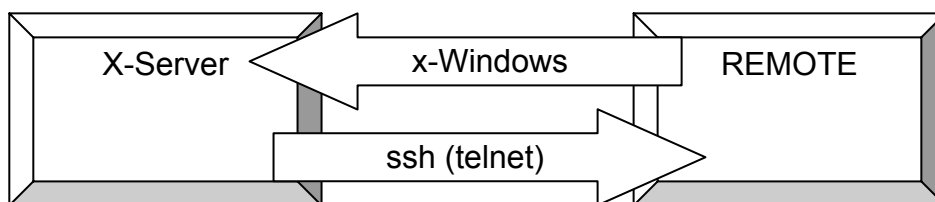
Bei einem laufenden X-Server (grafischer Oberfläche) ist es auch möglich anderen Rechnern zu gestatten X-Window-Programme auf dem (fremden) Rechner zu starten. Dies kann z.B. verwendet werden um mittels telnet oder ssh einen fremden Rechner zu übernehmen, ein Programm für die grafische Oberfläche zu starten und die Ausgabe dieses Programmes (wie kwrite oder kfmclient (KONQUERER)) auf den eigenen Rechner umzuleiten.

HINWEIS: Der Novellserver bietet die Möglichkeit (konfigurierbar über inetcfg an der Serverkonsole) einen Remotezugriff einzurichten. Neben der RCONSOLE, die nur das IPX-Protokoll verwendet kann auch eine X-Konsole eingerichtet werden. Damit ist von außerhalb eine telnet-Verbindung zum Server möglich, die Rückverbindung vom Server zum eigenen Rechner startet dann ein Fenster auf der X-Window-Oberfläche, von dem aus die Serverkonsole dann administrierbar wird. Da die Novellserver im allgemeinen jedoch nicht direkt im Internet sichtbar sind. Wäre für die Einrichtung einer solchen Verbindung noch eine spezielle Anpassung der Firewallregeln am Linuxrechner (wird im Teil 3 behandelt) notwendig.

ERGÄNZUNG: Eine einfachere Fernwartung des Novellserver erhält man jedoch, wenn man ein vt100 kompatibles Terminal verwendet. Dazu kann man unter Linux mit dem Befehl „export TERM=vt100“ in diesen Modus umsteigen. (Kontrolle mit „echo \$TERM“). Danach kann man mit einer Telnet-Verbindung zum Novellserver die Serverkonsole auf einen Linuxrechner umleiten (z.B.: Linux mit ssh übernehmen und dann vom Linuxrechner mit telnet die Serverkonsole übernehmen). Wenn ein direkter Zugriff auf den Novellserver möglich ist kann man z.B. auch mit dem Telnet von Windows 2000 die Konsole übernehmen:

- telnet (ohne Zielangabe) startet die telnet-Sitzung
- set TERM vt100 schaltet in den VT100 Modus
- open 10.x.y.z verbindet zum Server
- quit schließt Telnet

Siehe Grafik:



BEFEHLE:

Starten eines Programmes mit Ausgabe auf einem anderen Rechner:

```
kwrite -display 10.0.1.1:0.0 &
```

wenn vorher der Zugriff auf den Rechner 10.0.1.1 mit xhost MY_IP erlaubt wurde.

Xhost + schaltet die Zugriffskontrolle aus (alle dürfen)

xhost - schaltet die Zugriffskontrolle wieder ein

Oder export DISPLAY=10.0.1.1:0.0

```
kwrite
```

.....

Xn.hosts anlegen (steuert den Zugriff auf den Server n)



```
xlsfonts          zeigt die inst. Fonts
xfontsel          zeigt einen Probetext in der gew. Schriftart
xterm -geometry 70x35 -fn lucidasanstypewriter-bold-24 &
xterm -geometry +100+50 70x35 -fn lucidasanstypewriter-bold-24 //BS Position und Größe
```

.Xresources im jeweiligen Home Verzeichnis des Users. Dort können Sie Zeilen wie die folgenden eintragen:

```
XTerm*font: -misc-fixed-bold-r-normal--13-100-100-100-c-70-iso8859-1
XTerm*Background: bisque2
XTerm*Foreground: blue
XTerm*geometry: 90x40
```

Starten einer weiteren Oberfläche

```
/usr/X11R6/bin/X :1
BEFEHL -display localhost:1.0
bzw.: einen Fenstermanager auch auf der neuen Oberfläche starten:
export DISPLAY=localhost:1.0
/usr/X11R6/bin/windowmaker
```

Für die Verbindung zu einem X-Server wird der Port 6000+n verwendet, wobei n die Nummer des X-Servers ist. D.h. die erste X-Windows Oberfläche ist über Port 6001 erreichbar, die 2. über 6002 u.s.w.

2.2 ONLINE-UPDATE:

Manchmal treten im Zusammenhang mit dem SUSE Online Update Fehler auf. Z.B kann es bei manchen Rechnern beim Versuch die Patches zu installieren zur Fehlermeldung „Signaturüberprüfung fehlgeschlagen“ kommen und ein Installieren der Updatepakete ist mittels yast nicht möglich. Eine Abhilfe schafft hier z.B. das Konsolenscript: online_update, das man mit dem Parameter -n (keine Signaturprüfung) aufrufen kann. Ein Nebeneffekt ist, das dieser Befehl keine weiteren Benutzereingaben erfordert und daher auch zeitgesteuert aufgerufen werden kann (einbinden in den CRON-Dämon).

2.3 Grafischer Zugriff auf Rechner, die SSH-Dienste anbieten:

Mit dem Paket kio_fish wird ein Zusatzprotokoll installiert. Damit ist mit einem Browser über die URL: fish://root@www.xy.ac.at/etc ein Zugriff auf den fremden Rechner über das ssh-Protokoll (verschlüsselt) möglich. Damit können auch Dateien vom eigenen Rechner auf den Fernrechner übertragen werden (scp ... Secure Copy), bzw. Konfigurationsdateien am Fernrechner direkt bearbeitet werden.

2.4 CHKROOTKIT

In letzter Zeit häufen sich leider die Angriffe auf die Linuxrechner in den Schulen. Über diverse Sicherheitslücken werden sog. ROOT-KITS installiert, über die sich dann ein Hacker einen Zugang mit root-Rechten verschafft. Ein Tool zum Aufspüren von (bereits installierten) Root-Kits ist das Paket chkrootkit (<http://www.chkrootkit.org>) Dieses Skript (laufend updaten!!!) meldet falls bekannte Rootkits im System gefunden werden. (Für die Installation sind folgende Pakete notwendig (allenfalls nachinstallieren): make, gcc) → „make sense“ → „./chkrootkit“ (beide Befehle aus dem Verzeichnis aufrufen, indem das Paket liegt.)



3 Web-Server: (APACHE)

3.1 Grundkonfiguration:

Bei einer Standardinstallation wird der Webserver APACHE automatisch mitinstalliert und auch bereits gestartet (Einstellung über START_HTTPD – Variable in rc.config, bzw. ab SuSE 8.x ist mit dem RunlevelEditor einfach der entsprechende Startlink zu erstellen). Bei konfigurierem Netzwerk können sie von einer Workstation bereits mit einem Webbrowser unter <http://a.b.c.d> auf den Webserver des Linuxrechners zugreifen, wenn a.b.c.d stellvertretend für eine der Adressen des Linuxrechners steht. Wenn für diese Adresse bereits ein Rechner/Domainname vergeben ist und auch im entsprechenden Nameserver eingetragen wurde, kann der Zugriff auch über den Namen erfolgen.

Die Konfigurationsdatei des Servers ist /etc/httpd/httpd.conf.

Das Root-Verzeichnis des Webservers ist standardmäßig auf /usr/local/httpd eingestellt. (Ab SuSE 8.1 befinden sich www und ftp-Server unter /srv/www bzw. /srv/ftp) Hier findet man u.a. 2 wichtige Verzeichnisse:

cgi-bin:

Enthält ausführbare Dateien (*.cgi). In der Konfigurationsdatei wird dieses Verzeichnis als Script-Verzeichnis definiert, wodurch ein Ausführen von Programmen via WEB in diesem Verzeichnis erlaubt wird. Wenn man in einem anderen Verzeichnis auch das Ausführen von Scripts erlauben will, muss die Option ExecCGI für diesen Ordner eingeschaltet werden.

htdocs:

Das Wurzelverzeichnis für HTML-Dokumente. Beim Zugriff auf den Webserver wird zunächst nach der Datei index.html gesucht. Wenn Sie nun ihre eigene Homepage gestalten wollen, überspielen sie ihre Homepage (mit allen zugehörigen Dateien und Verzeichnissen) ins Verzeichnis /usr/local/httpd/htdocs, wobei die Startseite ihrer Homepage index.html heißen sollte. Benennen Sie vorher die bereits vorhandene Datei index.html z.B. in suse.html um. Damit können sie mit <http://a.b.c.d/suse.html> jederzeit auf die Suse-Dokumentation zugreifen.

Beachten Sie, dass alle Dateien ihrer Homepage mit dem Recht 644 (lesen für „others“) versehen werden und alle Verzeichnisse das Recht 755 (Ausführen/lesen für „others“) besitzen.

Es empfiehlt sich für die Wartung der Homepage einen eigenen Benutzer anzulegen, ihn als Besitzer aller Dateien unter /usr/local/httpd (bzw. /srv/www) festzulegen. Mittels ftp-Zugriff kann dieser Benutzer dann die Homepage von anderen Rechnern aus warten.

Weitere Tipps und Tricks zu den Wartungsaufgaben finden sie am PI-Webserver (<http://www.pinoe-hl.ac.at>) unter AG-Informatik → Anbindung eines Netzes ans Internet → WWW-Server Apache.

Wenn ein Ausführen von CGI-Skripts auch außerhalb des Verzeichnisses /srv/www/cgi-bin möglich sein soll, muss es in explizit erlaubt werden. (Notwendig z.B. für das WEB-Mailinterface MAILMAN)

```
<Directory "/srv/www/htdocs/mailman">  
    Options +ExecCGI +Includes  
</Directory>
```

Um die Funktionalität von PHP zu testen kann man folgenden Zähler verwenden:

```
<html>
```



```
<title> Ein Zähler </title>
<?php
    $FILENAME="counter.txt";
    $fp = fopen ($FILENAME,"r");
    if ($fp) {
        $zahl=fgets($fp,20);
        fclose($fp);
    }
    $zahl=$zahl+1;
    $fp=fopen($FILENAME,"w");
    fputs($fp,$zahl);
    echo "< h2> BESUCHER: ".$zahl."</h2>";
?>
</html>
```

3.2 Webseiten für Benutzer:

Sie können es ihren Benutzern am Linuxrechner gestatten sich eigenständig Webpages zu gestalten. Der Zugriff auf diese Seiten erfolgt mit <http://a.b.c.d/~USER>, wenn USER der Name des Benutzers ist. Dazu müssen sie in der Konfigurationsdatei den entsprechenden Teil für das Verzeichnis /home/*/public_html aktivieren, wobei sie keine so komplizierte Definition wie im Beispiel benötigen.

Es genügt z.B.:

```
<directory /home/*/public_html>
    Options Indexes
    Order allow,deny
    Allow from all
</directory>
```

Aufgaben für den jeweiligen Benutzer:

Er muß der Gruppe others das Execute-Recht für sein Homeverzeichnis einräumen (Read ist nicht notwendig) und dies ebenfalls für das anzulegende Verzeichnis public_html tun. Für die Zugriffsrechte auf die Dateien unter public_html gilt das Gleiche wie oben für die Homepage erwähnt.

3.3 Zugriff auf bestimmte Webseiten mittels Authentifizierung:

Man kann einzelne Verzeichnisse am Webserver sperren, sodass ein Zugriff nur mit gültiger Benutzerauthentifizierung möglich ist. (Siehe Dokumentation PI-Server → Web-Zugriff auf Verzeichnisse sperren)

Dazu muss eine eigene Passwortdatei angelegt werden:

z.B.: mit dem Befehl: "htpasswd -c /usr/local/httpd/allowed.users lehrer" eine Passwortdatei z. B. für den Benutzer **lehrer** anlegen. Danach erfolgt eine zweimalige Abfrage eines Passwortes. Dieses Passwort wird dann verschlüsselt in der Datei **allowed.users** abgespeichert. In dem Verzeichnis, das gesperrt werden soll (z. B. /usr/local/httpd/htdocs/**geheim**) wird mit einem Editor eine Datei mit dem Namen **.htaccess** angelegt. In diese Datei wird folgendes geschrieben:

```
AuthName "Verzeichnis geheim"
AuthType Basic
AuthUserFile /usr/local/httpd/allowed.users
require user privat
```



(oder require valid-user → Zugriff kann mit jeder gültigen Benutzerkennung erfolgen)

Bei den Standardeinstellungen des Apache-Servers werden die Einstellungen der Datei **.htaccess** ignoriert. Es muss daher für das Verzeichnis **/usr/local/httpd/htdocs/geheim** diese Möglichkeit aktiviert werden. Dazu editiert man die Datei **/etc/httpd/httpd.conf** und fügt für dieses Verzeichnis eine eigene Definition ein:

```
<Directory "/usr/local/httpd/htdocs/geheim">
  AllowOverride All
</Directory>
```

Wenn man weitere gültige Benutzerkennungen zur Passwortdatei hinzufügen will, so kann dies mit dem Befehl

“htpasswd /usr/local/httpd/allowed.users NocheinBenutzer“ (d.h. ohne -c) erfolgen.

Neben dieser einfachen Authentifizierung über Passwort und Gruppendateien sind auch Authentifizierungen gegen andere Quellen möglich von denen hier 2 erwähnt werden sollen:

3.3.1 LDAP:

(es können z.B. die NDS-Informationen des Novellservers via LDAP veröffentlicht werden)
Einbinden des notwendigen Modules in httpd.conf:

```
LoadModule ldap_auth_module /usr/lib/apache/mod_auth_ldap.so
```

```
<Directory "/usr/local/apache/htdocs/foo">
Options Indexes FollowSymLinks
AllowOverride None
order allow,deny
allow from all
AuthName "RCS Staff only"
AuthType Basic
LDAP_Server ldap.fccc.edu
LDAP_Port 389
Base_DN "o=Fox Chase Cancer Center,c=US"
UID_Attr uid
#require valid-user
require user muquit foo bar "john doe"
#require group cn=racs,ou=Groups
</Directory>
```

3.3.2 MYSQL:

Für eine Authentifizierung gegen eine MYSQL Datenbank muss das Modul mod_auth_mysql eingebunden werden.

```
AuthName Bereich
AuthType Basic
AuthMySQLHost localhost
AuthMySQLDB authdb
AuthMySQLUser myname
AuthMySQLPassword mypass
AuthMySQLUserTable users
AuthMySQLGroupTable groups
```



```
AuthMySQLNameField user
AuthMySQLPasswordField passwort
AuthMySQLGroupField group
AuthMySQLAuthoritative on
AuthMySQLCryptedPasswords on
AuthMySQLKeepAlive off
```

```
require group admin
```

3.4 INTRANET

Das folgende Beispiel zeigt ein Verzeichnis, das nur aus dem internen Netz betrachtet werden, oder von außerhalb, wenn man gültige Benutzerkennungen hat. Wenn der Parameter „Satisfy any“ auf „Satisfy all“ ändert ist ein Betrachten nur dann möglich, wenn beide Bedingungen erfüllt werden.

```
<Directory /srv/www/htdocs/intra>
  AuthType Basic
  AuthName intranet
  AuthUserFile /etc/httpd/users
  AuthGroupFile /etc/httpd/groups
  Require group customers
  Order allow,deny
  Allow from 10.0
  Satisfy any                // Standardeinstellung ist Satisfy all
</Directory>
```

3.5 Virtuelle Server:

Es ist auch möglich auf einem Rechner mehrere Webserver zu betreiben. Dazu ist es notwendig das DNS-Service entsprechend einzurichten (NickNames für den Rechner eintragen) und es Apache mitzuteilen wo die Dokumente für diesen virtuellen Server liegen. Für das angeführte Beispiel war es z.B. notwendig im DNS-Server für den Novellserver (IP 10.0.1.2) die Synonyme student.brg-wrn.ac.at, sta.brg-wrn.ac.at, al.brg-wrn.ac.at,..... einzutragen. Die Konfigurationsdatei für APACHE unter Novell befindet sich (bei installiertem Webserver) im Verzeichnis sys:\apache\conf und hat die gleiche Syntax wie unter Linux (bis auf die Verzeichnisangaben).

Ergänzung: Unter Novell wird der Server mit load sys:\apache\apache gestartet.

```
NameVirtualHost 10.0.1.2      //Unter welcher IP laufen die virt. Server
// mit der Anweisung NameVirtualHost 10.0.1.2:5045 wäre es z.B. möglich die virt.
// Server auf einem anderen Port laufen zu lassen.
```

```
<VirtualHost 10.0.1.2>
  ServerAdmin sta@brg-wrn.ac.at
  DocumentRoot sys:/apache/student
  ServerName student.brg-wrn.ac.at
  <Directory home:/schueler/*/public.www>
    AllowOverride FileInfo AuthConfig Limit
    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
    <Limit GET POST OPTIONS PROPFIND>
      Order allow,deny
      Allow from all
    </Limit>
    <LimitExcept GET POST OPTIONS PROPFIND>
```



```
        Order deny,allow
        Deny from all
    </LimitExcept>
</Directory>
</VirtualHost>

<VirtualHost 10.0.1.2>
    ServerAdmin sta@brg-wrn.ac.at
    DocumentRoot home:/lehrer/sta/public.www
    ServerName sta.brg-wrn.ac.at
</VirtualHost>

<VirtualHost 10.0.1.2>
    ServerAdmin sta@brg-wrn.ac.at
    DocumentRoot home:/lehrer/al/public.www
    ServerName al.brg-wrn.ac.at
</VirtualHost>
```

Nachtrag: Es kann sein, dass man den eigentlichen Server auch als virtuellen Server deklarieren muss um ihn neben den anderen virtuellen Servern auch zu erreichen. (Lt. Apache Doku sollte dies zwar nicht notwendig sein, die Seminarkonfiguration hat jedoch ohne diesen Eintrag nicht funktioniert.)

```
<VirtualHost 10.0.1.2>
    ServerAdmin sta@brg-wrn.ac.at
    DocumentRoot /srv/www/htdocs
    ServerName HAUPT.brg-wrn.ac.at
</VirtualHost>
```

3.6 LOG-Files

Die Zugriffe auf den Webserver, sowie eventuelle Fehlermeldungen werden in LOG-Files im Verzeichnis /var/log/httpd mitprotokolliert.

4 FTP-Server:

4.1 Bis SuSE 7.3:

Als Teil des Inetd-Dämons wird auch ein FTP-Server gestartet. (siehe entsprechende Zeile in inetd.conf). Man kann hier zwischen 3 verschiedenen Servern auswählen (sofern die entsprechenden Pakete installiert sind):

```
In.ftpd
Wu.ftpd
Proftpd
```

Verwenden Sie die Zeile in der wu.ftpd gestartet wird und kommentieren sie die beiden anderen Zeilen aus. Weitere Konfigurationsmöglichkeiten zum jeweilig verwendeten Server können sie sich über die entsprechende Manpage anzeigen lassen. Über die Datei /etc/ftphosts kann man den ftp-Zugriff einschränken (z.B. einen Benutzer auf das lokale Netzwerk beschränken)

```
/etc/ftpaccess
allow web 193.170.207.* 192.168.*
allow ftp *
allow guest *
```



Diese Einstellung erlaubt dem Webbenutzer den ftp-Zugriff nur aus der Schuldomäne und dem internen Netz, die Benutzer ftp und guest sind jedoch nicht eingeschränkt.

Zugeordnete Dateien:

ftppass; ftphosts; ftpusers

4.2 Ab SuSE 8.x

ist der Standard FTP-Server der Dienst vsftpd, der ebenfalls über inetd gestartet wird. Die Konfiguration dieses Dienstes erfolgt (etwas einfacher) über die Konfigurationsdatei /etc/vsftpd.conf. Eine genauere Beschreibung der Parameter erhält man mit „man vsftpd.conf“

```
# Example config file /etc/vsftpd.conf
#
# The default compiled in settings are very paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
#
# Allow anonymous FTP?
#anonymous_enable=YES
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
#anon_upload_enable=YES
#
# Uncomment this if you want the anonymous FTP user to be able to create
# new directories.
#anon_mkdir_write_enable=YES
#
# Activate directory messages - messages given to remote users when they
# go into a certain directory.
dirmessage_enable=YES
#
# Activate logging of uploads/downloads.
xferlog_enable=YES
#
# Make sure PORT transfer connections originate from port 20 (ftp-data).
connect_from_port_20=YES
#
# If you want, you can arrange for uploaded anonymous files to be owned by
# a different user. Note! Using "root" for uploaded files is not
# recommended!
#chown_uploads=YES
#chown_username=whoever
#
# You may override where the log file goes if you like. The default is shown
# below.
```



```
#xferlog_file=/var/log/vsftpd.log
#
# If you want, you can have your log file in standard ftpd xferlog format
#xferlog_std_format=YES
#
# You may change the default value for timing out an idle session.
#idle_session_timeout=600
#
# You may change the default value for timing out a data connection.
#data_connection_timeout=120
#
# It is recommended that you define on your system a unique user which the
# ftp server can use as a totally isolated and unprivileged user.
#nopriv_user=ftpsecure
#
# Enable this and the server will recognise asynchronous ABOR requests. Not
# recommended for security (the code is non-trivial). Not enabling it,
# however, may confuse older FTP clients.
#async_abor_enable=YES
#
# By default the server will pretend to allow ASCII mode but in fact ignore
# the request. Turn on the below options to have the server actually do ASCII
# mangling on files when in ASCII mode.
# Beware that turning on ascii_download_enable enables malicious remote parties
# to consume your I/O resources, by issuing the command "SIZE /big/file" in
# ASCII mode.
# These ASCII options are split into upload and download because you may wish
# to enable ASCII uploads (to prevent uploaded scripts etc. from breaking),
# without the DoS risk of SIZE and ASCII downloads. ASCII mangling should be
# on the client anyway..
#ascii_upload_enable=YES
#ascii_download_enable=YES
#
# You may fully customise the login banner string:
#ftpd_banner=Welcome to blah FTP service.
#
# You may specify a file of disallowed anonymous e-mail addresses. Apparently
# useful for combatting certain DoS attacks.
#deny_email_enable=YES
# (default follows)
#banned_email_file=/etc/vsftpd.banned_emails
#
# You may specify an explicit list of local users to chroot() to their home
# directory. If chroot_local_user is YES, then this list becomes a list of
# users to NOT chroot().
#chroot_list_enable=YES
# (default follows)
#chroot_list_file=/etc/vsftpd.chroot_list
#
# You may activate the "-R" option to the builtin ls. This is disabled by
# default to avoid remote users being able to cause excessive I/O on large
# sites. However, some broken FTP clients such as "ncftp" and "mirror" assume
# the presence of the "-R" option, so there is a strong case for enabling it.
#ls_recurse_enable=YES

pam_service_name=vsftpd
chroot_local_user=YES
```



4.3 FTP-Dienste am Novellserver:

Auch am Novellserver kann ein FTP-Serverdienst gestartet werden (load nwftpd). Falls dieser Dienst noch nicht installiert ist, kann er von der Netware5.1 CD mittels nwconfig nachinstalliert werden. Die Konfigurationsdateien (und auch die LOG-Files) für diesen Serverdienst befinden sich im Ordner SYS:\ETC. Die Syntax der Datei ftpserv.cfg ist ähnlich zur Konfiguration von vsftpd am Linuxrechner. Ausgehend von der Standardkonfiguration wurden bei meinem Server folgende Anpassungen vorgenommen: (HERVORHEBUNGEN)

```
#FTPSERVER Configuration File
#Format: A comment line starts with #
#   Each Configuration Parameter is in a single line of form
#   parameter=value.

#List of Configuration Parameters with their Values:

#IP address of the host on which FTP Server is being loaded. If parameter
# not specified, it binds the local host
HOST IP ADDR=10.0.1.2

#Port Number to which FTP Server should bind and listen for connection
#requests. If not specified, it binds to standard FTP port 21
FTP_PORT=21

#Maximun number of ftp sessions
MAX_FTP_SESSIONS=30

#Time duration in seconds for which the session can remain idle
#IDLE_SESSION_TIMEOUT=600

#To Allow or Deny Access to Anonymous Users
ANONYMOUS_ACCESS=NO

#Home Directory for Anonymous users
#ANONYMOUS_HOME=sys:/public

#Get email address as password for anonymous user access
#ANONYMOUS_PASSWORD_REQUIRED=YES

#Path of FTP user restrictions file
RESTRICT FILE=sys:/etc/ftprest.txt

#Server where the default home of users is present. If not specified
#it stays on the local server. Specify the server name only
DEFAULT USER HOME SERVER=EINSTEIN

#Default home directory of the user
#DEFAULT_USER_HOME=sys:/public

#Specifies whether to ignore home directory if it is on remote server
#and stay on the local server
#IGNORE_REMOTE_HOME=NO

#Specifies whether to ignore home directory and stay on default directory
#IGNORE_HOME_DIR=NO

#FTP Servers Default NameSpace
DEFAULT_NAMESPACE=LONG

#If Intruder checking not required set INTRUDER_USER_ATTEMPTS
# & INTRUDER_HOST_ATTEMPTS =0
#Number of invalid login attempts for intruder host detection
```



```
#INTRUDER_HOST_ATTEMPTS=20

#Number of invalid login attempts for intruder user detection
#INTRUDER_USER_ATTEMPTS=5

#Time interval in Minutes during which the intruder host is
#not allowed to login
#HOST_RESET_TIME=5

#Time interval in Minutes during which the intruder user is
#not allowed to login
#USER_RESET_TIME=10

#Minimum Port Number for Passive Connection
#PASSIVE_PORT_MIN=1

#Maximum Port Number for Passive Connection
#PASSIVE_PORT_MAX=65534

#Path of welcome banner file
WELCOME BANNER=sys:/etc/welcome.txt

#Message File Name
MESSAGE_FILE=sys:/etc/message.txt

#FTP Catalog object name (used for Contextless login)
#FTP_CATALOG_NAME=ftpcat

#FTP Log file creation directory
#FTP_LOG_DIR=sys:/etc

#Maximum number of messages that will be logged in
#all the log files
#NUM_LOG_MSG=32000

#Logging Level of FTP Log Files
#FTP_LOG_LEVEL=7

#FTP Daemon log file path - System log msgs
#FTPD_LOG=ftpd

#Audit log file path - general log msgs
#AUDIT_LOG=ftpaudit

#Intruder log file path - Intruder log msgs
#INTRUDER_LOG=ftpintr

#Statistics log file path - Statistics history log msgs
#STAT_LOG=ftpstat

#To allow or deny execution of quote site commands
#DISABLE_SITE_CMDS=NO

#Search list has a list of full DN names of containers
#separated by commas, from where the search should start for users.
#Maximum number of containers allowed is 25.If you do not set any search containers,
#search will start from the server's default context.
SEARCH LIST=lehrer.brg,schueler.brg
```

Weiters wurde im File FTPREST.TXT (RESTRICTIONS) folgende Anpassungen vorgenommen:



```
# This file contains the restrict file format for FTPserver.
# All comment lines should start with a '#'
# If a line continues in the next line, a '\' should be given
# in the end of the first line to indicated the continuation.
# The access rights permitted are
# DENY - Deny access to FTPserver for that client.
# READONLY - Gives Readonly Access to the Client.
# NOREMOTE - Restricts access to remote server navigation.
# GUEST - Gives only guest access to the user.
# ALLOW - Gives the user access to ftpserver.

#NOTE :
# Guest User cannot navigate to remote servers.
# Will be given acces only within his home directory and subdirectories.
# Access rights can be seperated by a comma(.).
# The Access rights are taken according to the order in which they
# appear in the restrict file.

#Key Words :
# "ADDRESS=" - Should be given to restrict a particular Node.
# IPAddress or Machine Name can be given.
# "ADDRESS_RANGE=" - Should be given to restrict a Range of Nodes based on the
# IPAddress. It applies the restriction to any node having
# the IP Address between the specified IP address Range.
# The Range is specified by two IP addresses separated by
# SPACE.
# "DOMAIN=" - Should be given to restrict a particular DOMAIN.
# "*" - Should be given for container level restrictions.
# "ACCESS=" - Mandatory for each line. Should be followed by access
# rights.
# "ALL" - Given for domain name, applies restrictions to all domains.

#Note :
# There should not be space between the word and '=' sign.

#File Format:
# Each line should have one of the Restriction level keywords
# and access keyword
# For container and User level restrictions, fully distinguished name
# should be give.
# Container/User name can be Canonical or a Full DN name.
# It should start with a period(.) for user Restriction.
# For Container level Restrictions the line should start with a '*'.
# For giving access restrictions to all domain,
# DOMAIN= ALL should be given.

# Examples:

#ADDRESS= 190.90.90.190 ACCESS= NOREMOTE, GUEST
#ADDRESS= testmachine.testdomain.com ACCESS= GUEST
#ADDRESS_RANGE= 164.99.154.1 164.99.155.255 ACCESS= GUEST
#DOMAIN= testdomain.blr.novell.com ACCESS= DENY

#*.testorgunit.testorg ACCESS= GUEST, READONLY
#*.OU=testorgunit.O=testorg ACCESS= GUEST, READONLY

#.testuser.testou.test0 ACCESS= NOREMOTE
#.CN=testuser.OU=testou.O=test0 ACCESS= ALLOW
```



```
#DOMAIN= ALL                ACCESS= NOREMOTE
*.OU=schueler.O=brg        ACCESS=GUEST
*.OU=lehrer.O=brg         ACCESS=GUEST
*.OU=server.O=brg         ACCESS=NOREMOTE
.CN=sta.OU=lehrer.O=brg   ACCESS=NOREMOTE
.CN=pfw.OU=lehrer.O=brg   ACCESS=NOREMOTE
```

Alle Benutzer, die der GUEST-Gruppe zugeordnet werden, können auf ihr Homeverzeichnis am Server zugreifen, dieses aber nicht verlassen (vergleichbar mit dem Parameter `chroot_local_user` am Linuxrechner). Die Benutzer des Containers server, bzw. 2 spezielle Lehrerbenutzer wurden von dieser Regelung ausgenommen und haben FTP-Zugriff auf alle Ordner am Novellserver, die ihnen auch bei der Arbeit im SchulLAN zur Verfügung stehen. NOREMOTE bedeutet jedoch, dass sie über den Novellserver EINSTEIN an dem sie sich mittels FTP anmelden keinen Zugriff auf eventuell weitere Novellserver (in der gl. NDS) erhalten.

5 Proxy-Server:

Ein Proxyserver (PROXY = Stellvertreter) richtet Webanfragen stellvertretend für seine Clients ins Internet. Der Vorteil dieser Lösung liegt darin, dass die Seiten aus dem Internet geholt werden und auch lokal (am Proxyserver) gespeichert werden. Ein nochmaliger Aufruf der Seite bewirkt beim Proxyserver nur eine Anfrage ins Internet, ob sich die Seite inzwischen geändert hat. Wenn dies nicht der Fall ist, wird die Seite aus dem lokalen Cache geliefert.

Der Standardproxyserver am Linuxrechner ist SQUID in der Version 2. Für den Start dieses Dienstes existiert in der Datei `rc.config` wieder eine Variable mit dem Namen `START_SQUID` (yes/no). Beim erstmaligen Start werden automatisch unter `/var/squid` die Cacheverzeichnisse angelegt.

Squid operiert standardmäßig am Port 3128 (dies kann jedoch in der Datei `/etc/squid.conf` auf den Port 8080 geändert werden; Standard bei vielen anderen Proxyserverdiensten). Es ist auch möglich Squid so zu konfigurieren, dass er auf beiden Ports lauscht.

Squid kann für Web-Abfragen (http) und für Downloadprozesse (ftp) als lokaler Zwischenspeicher verwendet werden.

Man muss nur noch bei den Clients (Netscape → Einstellungen, IE → Internetoptionen) das Verwenden eines Proxyserver einstellen. Als Proxyserver trägt man hier eine der IP-Adressen des Linuxrechners ein (am Besten die, die von der WS als erstes sichtbar wird). Der Proxyport (falls nicht geändert) ist mit 3128 gegeben. Wählen sie danach eine Internetseite, die noch nicht im lokalen Cache des Rechners ist, von mehreren Workstations, die alle den Proxy verwenden, an. Der Aufbau der ersten Seite ist noch immer so langsam wie vorher. Wenn man jedoch dieselbe Seite von einer anderen Workstation aus aufruft sollte sie deutlich schneller angezeigt werden.

Wenn sie nach der Erstinstallation Squid erstmalig verwenden wollen, werden sie möglicherweise eine Fehlermeldung in der Art „Access denied“ erhalten. Der Grund dafür ist, dass die Verwendung in der Datei `squid.conf` gesperrt ist. Editieren sie diese Datei und suchen sie nach einem Eintrag der Art: „`http_access deny all`“. Ändern sie in diesem Eintrag „deny“ auf „allow“ und sie sollten über einen verwendbaren Proxyserver verfügen.

(Eine ausführlich dokumentierte Version der Datei `Squid.conf` finden sie unter <http://www.gymmelk.ac.at/~berx/squid.html>)

Besser: (definiert welche Adressen zur Schule gehören)

```
acl brg src 193.170.207.0/255.255.255.0
acl brg src 10.0.0.0/16
.....
http_access allow brg
```



Die letzte Zeile erlaubt einen Zugriff von Rechnern aus dem Bereich brg. Eine Zeile der Form `http_access deny all` ist nicht notwendig, da falls die letzte Regel eine `allow` ist, implizit ein `deny all` danach angehängt wird.

5.1 Kaskade der PROXYSERVER:

Man kann die Speicher der Proxyserver kaskadieren. Dies bedeutet, dass wenn eine Seite nicht im lokalen Cache vorhanden ist, zunächst der übergeordnete Proxyserver angefragt wird, bevor die Information direkt aus dem Internet geholt wird. Sie sollten auf jeden Fall als übergeordneten Server den Proxy des Landesschulrates eintragen (in NÖ). Dazu suchen sie in ihrer `squid.conf` nach einem Eintrag `„cache_peer hostname type 3128 3130“`. Fügen Sie nach dieser (auskommentierten) Zeile eine weitere Zeile mit folgendem Inhalt ein:

```
„cache_peer proxy.lsr-noe.gv.at parent 8080 3130“
```

Einige Schulen haben ihren Proxyserver via Sibling für andere NÖ-Schulen freigegeben. Information wie sie auch diese Speicher einbinden können finden sie unter <http://www.pinoe-hl.ac.at/arge/ahsinf/linuxsquid.html>

5.2 Filtern von Webseiten:

5.2.1 Filterlösung direkt über Squid:

(Anleitung unter <http://www.htl-kapfenberg.ac.at/squid/index.html>)

Das von Rainer Grabher unter obiger Adresse angebotene Firewallkonzept kann ab der Version 6.2 nicht mehr verwendet werden (`ipfwadm` → `ipchains`), die Konfiguration von Squid kann jedoch (vor allem seine Sammlung verbotener Adressen) übernommen werden. Die entscheidenden Zeilen, die in die Datei übernommen werden müssen lauten:

```
acl denied_hosts dstdomain „/etc/noaccess“  
acl denied_phrases url_regex „/etc/nophrases“
```

```
http_access deny denied_hosts  
http_access deny denied_phrases
```

Die ersten beiden Zeilen definieren die Variablen `denied_hosts` und `denied_phrases`, wobei der Inhalt aus 2 Dateien geholt wird. Wenn man eine Internetsite sperren will, trägt man die Adresse in `/etc/noaccess` ein, wenn man einen Begriff sperren will (egal in welcher Adresse er auftaucht) muss man ihn in `/etc/nophrases` ergänzen.

Die letzten beiden Zeilen verbieten es, dass der Inhalt von `denied_hosts`, bzw. `denied_phrases` angezeigt wird. Hierbei ist es entscheidend, dass diese Zeilen vor der allgemeinen Freigabe `„http_access allow all“` stehen.

Wenn (!!) die Clients nun den Proxyserver verwenden, werden Webzugriffe gefiltert. Es ist jedoch noch immer möglich ohne Proxy zu surfen (Direkte Verbindung ins Internet am Client einstellen). Diese Hintertür kann man nur durch Aktivieren einiger Firewallregeln schließen. (Im Teil 3 wird dafür eine elegante Möglichkeit vorgestellt.)



5.2.2 Filtern über ein externes Programm (SQUIDGUARD):

In den letzten SuSE - Versionen wird noch ein Paket mit der Bezeichnung „SquidGuard“ mitgeliefert, das noch mehr Möglichkeiten als die soeben vorgestellte Filterlösung bietet. Dazu muss man das Paket installieren, in der Datei /etc/squid.conf die Zeile „redirect_program /usr/sbin/squidGuard“ einfügen. Damit werden alle angeforderten Webseiten zunächst an das Programm SquidGuard übergeben, in dem dann über die weitere Vorgangsweise entschieden wird. Die Konfiguration dieses Dienstes erfolgt über die entsprechende Datei im Verzeichnis /etc.

In der Datei squidguard.conf können nun zunächst Benutzergruppen und Ziele im Web (dest) definiert werden. Danach werden Zugriffsrechte (ACL's) festgelegt. Innerhalb dieser Regeln kann nun für jede Gruppe festgelegt werden, welche Zugriffe erlaubt und was verboten ist. Im Falle eines Verbotes kann mit einer Redirect - Anweisung auf eine HTML-Seite verwiesen werden, die z.B. eines Fehlermeldung ausgibt.

Eine genauere Information findet man auf der Seite: <http://www.squidguard.org> und im Verzeichnis /usr/share/doc/packages/squidgrd.

Die Datei /etc/squidguard.conf:

```
logdir /var/squidGuard/logs
dbhome /var/squidGuard/db/blacklists

src verwaltung {
    ip 10.1.0.0/24 # range 10.0.0.0 - 10.0.0.255
}

src schule {
    ip 10.0.0.0/16 # range 10.0.0.0 - 10.0.255.255
}

dest ads {
    domainlist ads/domains
    urllist ads/urls
}

dest freigabe {
    domainlist freigabe/domains
    urllist freigabe/urls
}

dest aggressive {
    domainlist aggressive/domains
    urllist aggressive/urls
}

dest av {
    domainlist audio-video/domains
    urllist audio-video/urls
}

dest drugs {
    domainlist drugs/domains
    urllist drugs/urls
}
```



```
}

dest gambling {
    domainlist gambling/domains
    urllist gambling/urls
}

dest hacking {
    domainlist hacking/domains
    urllist hacking/urls
}

dest mail {
    domainlist mail/domains
}

dest porn {
    domainlist porn/domains
    urllist porn/urls
    expressionlist porn/expressions
}

dest proxy {
    domainlist proxy/domains
    urllist proxy/urls
}

dest warez {
    domainlist warez/domains
    urllist warez/urls
}

dest violence {
    domainlist violence/domains
    urllist violence/urls
    expressionlist violence/expressions
}

acl {
    verwaltung {
        pass all
    }

    schule {
        pass freigabe !ads !aggressive !av !drugs !gambling !hacking !mail !porn !proxy !violence
    }
    !warez all
    redirect http://10.0.0.1/warnung.html
}

default {
    pass none
    redirect http://10.0.0.1/warnung2.html
}
}
```



5.3 Transparenter PROXY

Durch eine Anpassung der Firewall kann man aus dem internen Netz nach außen (auf den Port 80) gerichtete Anfragen automatisch an den lokalen Port 3128 (SQUID) umleiten.

Damit der Proxyserver Squid nun als Transparenter Proxy (die Benutzer merken gar nicht, dass sie einen Proxyserver verwenden) agiert, muss in seiner Konfigurationsdatei noch eine kleine Veränderung vorgenommen werden:

Suchen sie in der Datei SQUID.CONF nach dem Abschnitt: HTTPD-ACCELERATOR OPTIONS und belegen sie den Wert folgender Variablen:

```
httpd_accel_host    virtual
httpd_accel_port    80
httpd_accel_with_proxy  on
httpd_accel_uses_host_header  on
```

Starten sie danach Squid mit „rcsquid reload“ neu. Danach sollte jede Workstation aus dem internen Netzwerk automatisch (unabhängig von ihren lokalen Einstellungen) einen Webzugriff über den Proxy vornehmen.

6 Mail-Server:

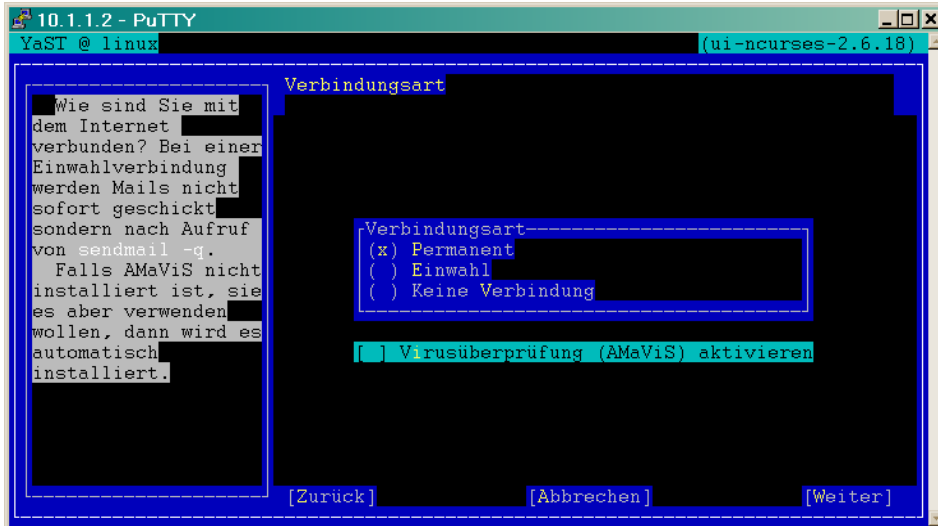
Eine Email-Adresse besteht grundsätzlich aus 2 Teilen, der Benutzerkennung und einem gültigen Internetnamen für einen Rechner. Diese beiden Teile werden durch das Zeichen @ getrennt. Wenn ihr Linuxrechner z.B. den Internetnamen mail.DOMÄNE.ac.at (wobei DOMÄNE für die Domänbezeichnung ihrer Schule steht) hat und sie auf diesem Rechner (z.B. mit YAST) einen Benutzer mit Namen Kurt anlegen, hat dieser Benutzer automatisch die (weltweit gültige) Email-Adresse kurt@mail.DOMÄNE.ac.at. Da ihr Rechner mit 3 Namen konfiguriert ist (www, mail, ftp) kann dieser Benutzer auch Mails auf kurt@www.DOMÄNE.ac.at und kurt@ftp.DOMÄNE.ac.at empfangen. Die Domänbezeichnung selbst ist eigentlich keine Adresse für einen Rechner. Damit (einfachere) Emailadressen wie kurt@DOMÄNE.ac.at möglich sind, wird ein MailExchange (MX) Eintrag für DOMÄNE.ac.at gemacht, der definiert, welcher Rechner die Domänmails erhalten soll. Diese Eintragungen sind am Nameserver durchzuführen. Der Nameserverdienst für fast alle Schulen in NÖ wird vom LSR übernommen (Ing. Wirlach).

Damit dies auch funktioniert, sind für das Programm SENDMAIL am Linuxrechner noch einige einfache Konfigurationsarbeiten notwendig. Grundsätzlich besteht ein Mailserver aus zwei verschiedenen Diensten:

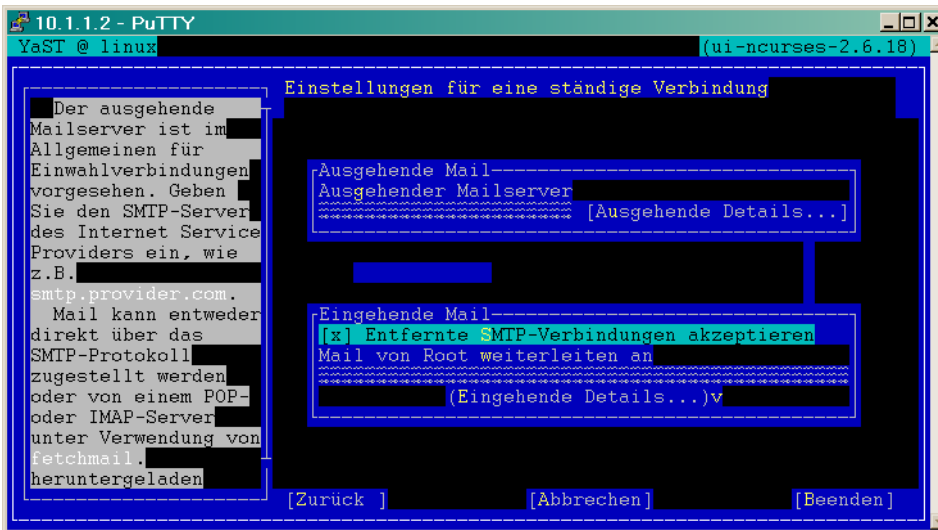
6.1 SMTP : (Sendmail – Port 25)

Verantwortlich für den Versand von Nachrichten. (Simple Mail Transport Protocol) Dieser Dienst wird überwiegend vom Programm SENDMAIL übernommen. Sendmail ist ein mächtiges Programmpaket und kann auch viele Spezialaufgaben übernehmen (siehe umfangreiche Konfigurationsdatei /etc/sendmail.cf) . Für solche Fälle sei auf das umfangreiche Werk über Sendmail (1021 Seiten; B.Costales / E. Allman) aus dem OReilly-Verlag verwiesen.

Für die Verwendung von Domainmails muss Sendmail noch besonders konfiguriert werden. Über yast → Administration des Systems → Netzwerk konfigurieren → Sendmail konfigurieren kommt man in ein Konfigurationsmenü für Sendmail. Damit der Rechner nicht nur Emails auf seinen primären Rechnernamen akzeptiert muss man im Expertenmodus eine kleine Korrektur vornehmen.

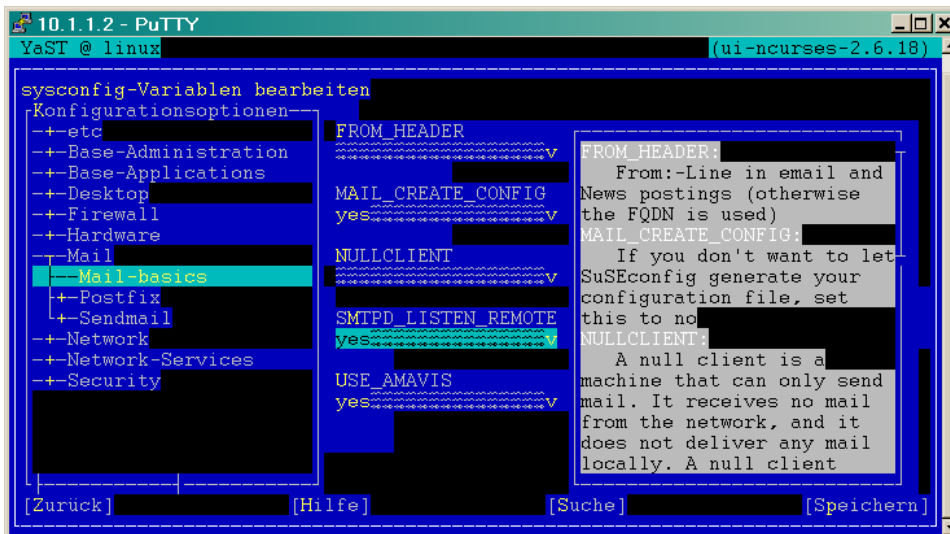


Amavis einschalten



Versenden von Emails über den Linuxrechner auch von anderen Rechnern aus erlauben

Über den Sysconfig Editor (Yast-Modul) werden noch Einträge bearbeitet, die den Mailversand betreffen:



Sendmail Einstellungen



Im Parameter sendmail_localhost ist z.B. folgendes einzutragen: mail.brg-wrn.ac.at www.brg-wrn.ac.at brg-wrn.ac.at

Wenn die Eintragungen auch am Nameserver passend vorgenommen wurden, sollte einem funktionsfähigen Betrieb als Mailserver nichts mehr im Wege stehen. Möglicherweise erhalten sie bei einem Versuch selbst Mails zu versenden die Nachricht, dass ihre Mail nicht weitergeleitet wurde. Der Grund dafür ist, dass in den neueren Versionen von Sendmail das Relay (Weiterleiten) zunächst abgeschaltet ist, um einen Missbrauch des Servers für SpammingMails zu unterbinden. Man muss in der Datei /etc/mail/access die Rechner (Netze) freischalten, die über den Linuxrechner Mails versenden dürfen:

```
/etc/mail/access:
    brg-wrn.ac.at      OK
    192.168.          OK
```

Diese Konfiguration erlaubt das Weiterleiten von Mails, wenn sie von Rechnern kommen, die der Domäne brg-wrn.ac.at oder dem B-Klasse Netzwerk 192.168.x.x angehören.

Die Textdatei muss nun in ein Datenbankformat übersetzt werden:

makmap hash /etc/mail/access.db < /etc/mail/access oder nur aufrufen von SuSEconfig

Danach muss Sendmail neu gestartet werden (rcsendmail restart). Falls der Rechner noch nicht zur Zufriedenheit funktioniert, empfiehlt es sich einmal den Rechner neu zu starten.

Wenn sie nun auf ihrem Linuxrechner einen Benutzer anlegen erzeugen sie damit automatisch einen Email-Account. Am Client tragen sie als Mailserver (POP3, SMTP) jeweils die IP-Adresse oder DNS-Namen ihres Linuxrechners ein. Als POP3-Account verwenden sie den Benutzernamen und das zugehörige Passwort.

Wenn eine Mail nun an ihren Linuxrechner gesendet wird (aus dem Internet) und Sendmail diese Nachricht als lokal zustellbar erkennt, wird die Nachricht in das Postfach des entsprechenden Benutzers eingeordnet. Das Postfach ist einfach eine Datei im Verzeichnis /var/spool/mail, an die die Nachricht angehängt wird.

Eine zu versendende Mail wird in die Mail-Warteschlange (/var/spool/mqueue) eingeordnet. Diese Warteschlange wird in periodischen Abständen (ca. 10min.) abgearbeitet.

Weitere Infos siehe: <http://www.pinoe-hl.ac.at/arge/ahsinf/linuxmail.htm>

6.1.1 DETAILS SMTP:

von der Kommandozeile (mit mail):

```
mail -s „SUBJECT“ EMPFÄNGER
DATEN
DATEN
.           {Der Punkt bedeutet Ende der Mail}
```

mittels telnet auf Port 25 (des Mailserver) (telnet RECHNER 25)

```
HELO      gast.domain      (Bezeichnung des Clientrechners)
MAIL FROM: abc@gast.domain (Absender)
RCPT TO:  cde@mailserver.domain (Empfänger)
DATA
Subject:  Kurzbezeichnung
Daten
.....
```



Daten

QUIT

6.2 POP3: (PORT: 110)

Das PostOfficeProtokoll ist verantwortlich für das Abholen der Post aus dem Postfach. Damit dies funktioniert muss am Linuxrechner ein POP3-Serverdienst laufen. Dieser Dienst (qpopper oder ipop3d aus dem Paket imap) wird als Teil des inetd gestartet. (siehe Zeile in inetd.conf)

```
# Pop et al
#
# pop2 stream tcp  nowait root  /usr/sbin/tcpd  ipop2d
# pop3 stream tcp  nowait root  /usr/sbin/tcpd  ipop3d
# pop3 stream tcp  nowait root  /usr/sbin/tcpd  /usr/sbin/popper -s
```

POP3 Sitzung:

Sobald man bei einem Pop3 Server eingeloggt ist, wird das Postfach gesperrt (Transaktionsphase)

Kommandos	Bedeutung
APOP	Verschlüsseltes Einloggen (Optional)
DELE	Markiert eine Nachricht als gelöscht.
LAST	Gibt die höchste bisher bearbeitete Nachrichtennummer zurück.
LIST	Gibt die Größe der Nachricht(en) zurück.
NOOP	No Operation, gibt einen positiven Wert zurück, falls der Server noch lebt.
PASS	Übermittelt das Passwort für USER
RSET	Setzt die Markierung aller als gelöscht markierten Nachrichten zurück.
RETR	Holt eine komplette Nachricht (Head und Body).
STAT	Ermittelt die Anzahl der vorhandenen Nachrichten und die Größe der Mailbox.
TOP	Holt den Header und die angegebenen Zeilen der Nachricht.
TOP 10 5	holt den Header und die ersten 5 Zeilen von Nachricht 10. (Optional)
UIDL	(Unique ID Listing) Fragt nach der eindeutigen Kennung der Nachricht. (Optional)
USER	Übermittelt den Usernamen für die Mailbox (maildrop)
QUIT	Beendet die Verbindung. Löscht alle als gelöscht markierten Mails.

6.3 IMAP: (PORT 143)

IMAP in inetd.conf

```
#
# Imapd - Interactive Mail Access Protocol server
# Attention: This service is very insecure
# imap stream tcp  nowait root  /usr/sbin/tcpd  imapd
#
```

Als Alternative zu POP3 können die Mails auch via IMAP betrachtet werden. Der Hauptgrund für die Einführung von IMAP liegt in der Tatsache, dass, wenn die Mails via POP3 vom Server heruntergeladen werden sie nur auf dem einen lokalen Rechner vorliegen auf dem man arbeitet. Wenn man nun von mehreren Rechnern aus arbeitet ist am aktuellen Rechner immer nur ein Teil der Mails vorhanden. Alternative: Die Mails bleiben am Server liegen. → IMAP (<http://www.imap.org>)



Key goals for IMAP include:
Be fully compatible with Internet messaging standards, e.g. MIME.
Allow message access and management from more than one computer.
Allow access without reliance on less efficient file access protocols.
Provide support for "online", "offline", and "disconnected" access modes *
Support for concurrent access to shared mailboxes
Client software needs no knowledge about the server's file store format.

6.4 **Virens scanner:**

Ab SuSE 7.2 ist eine Email-Virens scannerlösung implementiert. Dazu wird über die Variable START_AMAVIS der Virens scanner gestartet. Über die in der Datei sendmail.cf eingestellte Option wird bereits standardmäßig jede Mail zunächst von Amavis entpackt an den Virens scanner Antivir übergeben und geprüft. Nur wenn keine Viren gefunden werden wird die Mail an den Empfänger weitergeleitet. Neuere Virendefinitionen kann man sich von www.antivir.de herunterladen. Diese Definitionen müssen dann ins Verzeichnis /var/lib/AntiVir abgelegt werden.

Die infizierten Mails werden unter /var/spool/vscan/..... abgelegt.

SuSE 8.x:

Für AntiVir existiert ein Konfigurations skript: /etc/antivir.conf

```
#  
# Sample AntiVir configuration file  
#  
  
# You can receive email notifications of viruses using this  
# directive. You must specify the email address to which the  
# notification will be sent. There is no default value for  
# this directive.  
EmailTo xyz@abschule.ac.at  
  
# Virus activity may also be logged to a specified file  
# (in addition to syslog). You must specify the file. There  
# is no default value for this directive.  
LogTo /var/log/antivir.log  
  
# New engine and virus data files can be automatically  
# updated via the internet. Please use only one of the  
# following options as the last one will be taken. You  
# can choose to have the automatic updates be every 2 hours  
# or once a day. If neither directive is given, the  
# automatic internet updater will be disabled.  
# Note: Internet updates can also be done manually using  
# the --update parameter with the command line  
# scanner.  
#AutoUpdateEvery2Hours  
#AutoUpdateDaily  
  
# If automatic updates are done daily, you can specify  
# at what time of day the updates should be done.  
#AutoUpdateTime 4:23
```

Die letzten Optionen bieten die Möglichkeit ein automatisches Aktualisieren der Virendefinitionen vorzunehmen. Eine andere Möglichkeit bietet der Befehl „antivir - - update“ der als CRON Job aufgerufen werden kann. Damit wird dieser Befehl zu einem bestimmten Zeitpunkt automatisch ausgeführt.



6.5 NACHTRAG zu POP3 und IMAP:

In beiden Protokollen werden die Benutzerdaten unverschlüsselt über das Netz übertragen. Diese Sicherheitslücke könnte man mit einigem Zusatzaufwand schließen, da beide Server die Möglichkeit bieten auf mit Verschlüsselung zu arbeiten (APOP, SSL,...) Hinweise dazu findet man in den Dokumentationen der Pakete qpopper oder imap (im Verzeichnis /usr/share/doc/packages)

7 CRON:

Zur Steuerung dieser automatischen Abläufe dient die Datei /etc/crontab:

```
SHELL=/bin/sh
PATH=/usr/bin:/usr/sbin:/sbin:/bin:/usr/lib/news/bin
MAILTO=root
#
# check scripts in cron.hourly, cron.daily, cron.weekly, and cron.monthly
#
-*/15 * * * * root test -x /usr/lib/cron/run-crons && /usr/lib/cron/run-crons >/dev/null 2>&1
59 * * * * root rm -f /var/spool/cron/lastrun/cron.hourly
14 0 * * * root rm -f /var/spool/cron/lastrun/cron.daily
29 0 * * 6 root rm -f /var/spool/cron/lastrun/cron.weekly
44 0 1 * * root rm -f /var/spool/cron/lastrun/cron.monthly
```

Die Syntax dieser Datei kann mit "man 5 crontab" nachgelesen werden.

AUSZUG:

The time and date fields are:

field	allowed values
----	-----
minute	0-59
hour	0-23
day of month	1-31
month	1-12 (or names, see below)
day of week	0-7 (0 or 7 is Sun, or use names)

EXAMPLE CRON FILE

```
# use /bin/sh to run commands, no matter what /etc/passwd says
SHELL=/bin/sh
# mail any output to `paul', no matter whose crontab this is
MAILTO=paul
#
# run five minutes after midnight, every day
5 0 * * * $HOME/bin/daily.job >> $HOME/tmp/out 2>&1
# run at 2:15pm on the first of every month -- output mailed to paul
15 14 1 * * $HOME/bin/monthly
# run at 10 pm on weekdays, annoy Joe
0 22 * * 1-5 mail -s "It's 10pm" joe%Joe,%%Where are your kids?%
23 0-23/2 * * * echo "run 23 minutes after midn, 2am, 4am ..., everyday"
5 4 * * sun echo "run at 5 after 4 every sunday"
```

8 Benutzerplatzbeschränkung - QUOTAS

Bei einer größeren Anzahl von Systembenutzern ist es notwendig, dass der, den Benutzern zur Verfügung stehende Plattenplatz beschränkt wird. Dies kann mit den sog. QUOTAS erfolgen. Wenn sie das Paket QUOTA (Serie ap1) installiert haben, müssen die Beschränkungen noch konfiguriert und aktiviert werden. Dazu legen sie im Wurzelverzeichnis des Dateisystems, auf dem



sie Beschränkungen verwenden wollen die Datei "aquota.user" an. Für die Standardinstallation in NÖ ist dies eigentlich nur für die Root-Partition sinnvoll und kann z.B. mit dem Befehl "touch /aquota.user" durchgeführt werden (als root-Benutzer). Weisen sie dieser Datei mit "chmod 600 aquota.user" die passenden Filerechte zu.

Editieren sie nun die Datei "/etc/fstab". In dieser Datei finden sie eine Zeile, in der die Optionen für das Mounten der Root-Partition stehen: (Quotas sind nur in z.B:

```
/dev/hda2 / ext3 defaults, usrquota .....
```

Fügen sie in dieser Zeile die Option usrquota hinzu. Diese Option aktiviert die Quotas auf der entsprechenden Partition.

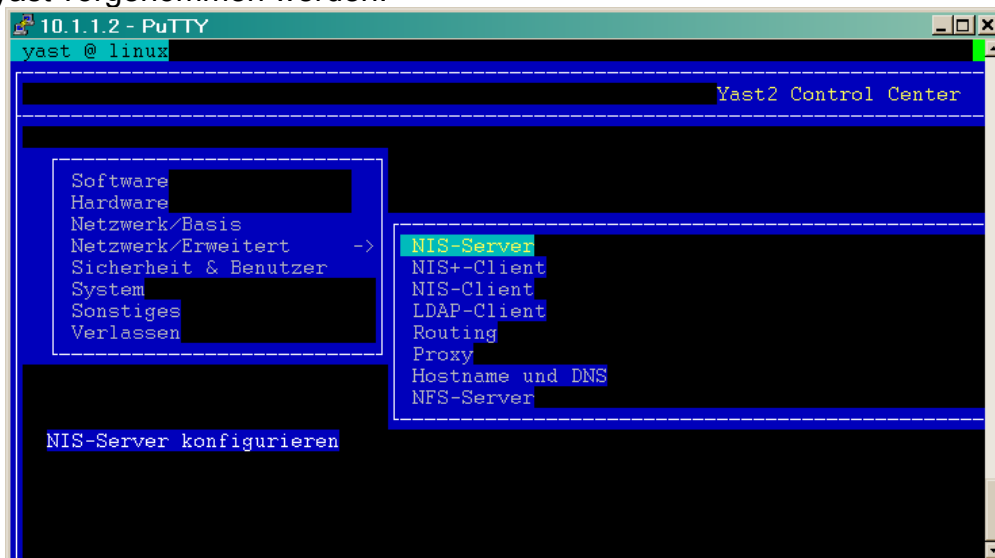
Speichern sie die Änderungen ab und editieren sie danach die Datei "/etc/rc.config". Setzen sie in dieser Datei den Parameter START_QUOTA auf yes. Jetzt müssen die bereits vorhandenen Dateien ihren Besitzern zugeordnet werden. Die erste Initialisierung wird mit dem Befehl **quotacheck -acuvqm** durchgeführt, danach starten sie den Rechner neu.

Nach dem neuerlichen Start des Rechners werden die Quotas aktiviert. Mit dem Befehl "quota *Benutzername*" können die Quotas eines Benutzers eingesehen werden. Zum Einstellen von Beschränkungen kann der Befehl "edquota *Benutzername*" verwendet werden. Sie können im Editor (vi) nun für den jeweiligen Benutzer ein Softlimit (darf einige Zeit überschritten werden) und ein Hardlimit einstellen. (Im vi speichert man mit ":w" ; Verlassen: ":q"). Für eine größere Anzahl verwendet man einen Beispielbenutzer und macht die anderen äquivalent zu diesem. (kann mit edquota gemacht werden)

Hilfe erhalten sie über die entsprechende Manpage. Eine einfache Überprüfung und Einstellung der Quotas kann auch über das WEBMIN – Interface erfolgen.

9 NIS und NFS:

NIS (Network Information System auch yellow pages genannt) bietet die Möglichkeit Benutzerinformationen übers Netz zu verteilen und NFS (Network File System) die Möglichkeit Dateifreigaben für das Netz festzulegen. Die Konfiguration der Serverdienste können relativ einfach über yast vorgenommen werden:



NIS/NFS



In einer Arbeitsgruppe wird ein Rechner als Server definiert, während die anderen Rechner als NFS und NIS –Client agieren. Zu beachten ist, dass die Homeverzeichnisse der Netzwerkbenutzer auch auf einem NFS-Share liegen und damit von jedem Arbeitsplatz aus erreichbar sind. Über diese beiden Dienste können somit Linuxnetzwerke aufgebaut werden. Wenn man Windowsrechner einbinden will empfiehlt sich die Verwendung von SAMBA.

10 SQL-SERVER – MYSQL

10.1 Installation und Konfiguration

Die Pakete zu MYSQL finden sie in der Serie „*base*“ der SuSE Distribution. Installieren sie alle Pakete die mit der Bezeichnung *mysql*... beginnen (6). Danach finden sie in *rc.config* eine Variable *START_MYSQL* (yes/no). Damit wird MYSQL gestartet und ist bereits einsatzfähig. Die Datenbanken und die Steuerung der Zugriffsrechte (befinden sich in einer Datenbank mit Namen „*mysql*“) finden sie im Verzeichnis */var/mysql*.

Sie können nun von der Linuxkonsole aus mit „*mysql*“ einen SQL-Client starten, von dem aus sie mit SQL-Befehlen auf den Server zugreifen können.

Eine Einführung in die Verwendung von SQL finden sie z.B. in „SQL - Der Schlüssel zu relationalen Datenbanken“ von G. Kuhlmann und F. Müllmerstadt

10.2 Zugriff von Windows-Workstations

Der Zugriff von Windowsrechnern auf den SQL-Server erfolgt über einen ODBC-Treiber, den man aus dem Internet unter <http://www.mysql.com/download/myodbc.html> herunterladen kann. Nach der Installation steht ein unter Systemsteuerung → ODBC-Datenquellen ein neuer Treiber (MySQL) zur Verfügung.

Um auch die Rechteverwaltung über z.B. ACCESS durchführen zu können ist am Linuxrechner noch eine kleine Vorarbeit notwendig:

Starten sie von der Konsole mit „*mysql*“ den SQL-Client. Mit „*use mysql*“ greifen sie auf die Systemdatenbank zu und mit „*show tables;*“ können sie sich die darin enthaltenen Tabellen anzeigen lassen. Den Inhalt der jeweiligen Tabelle können sie mit dem SQL-Befehl „*select * from xyz;*“

Für die Zugriffsrechte grundlegend sind die Tabellen „*user*“ und „*db*“. In der *user*-Tabelle werden die einzelnen SQL-Benutzer (nicht gleich Unix-Benutzer) definiert und ihre grundlegenden Einstellungen festgelegt. In der *db*-Tabelle werden dann zusätzliche Rechte für die jeweilige Datenbank eingestellt.

In der Standardinstallation ist ein Benutzer *root* (ohne PW, <> Unix-Benutzer *root*) angelegt, der alle Tabellen verändern darf (allerdings nur vom lokalen Rechner aus).

Mit dem Befehl „*update user set Host = „%“ where Host = „RECHNERNAME“*“ wird der Platzhalter % (Alles) in die Hostspalte eingetragen. Damit wird der Zugriff auf alle Rechner erweitert.

Wenn sie nun eine entsprechende Datenquelle definieren können sie z.B. von Access aus auf die Datenbank *mysql* am Server zugreifen und so komfortabler ihre Zugriffsrechte einstellen.

Aus Sicherheitsgründen sollten sie auf jeden Fall für den *root*-(SQL)-Benutzer ein Passwort vergeben. Dies können sie vom *mysql*-Client des Linuxrechners aus mit folgendem Befehl durchführen:

```
Update user Set Password=Password('MYPW') Where User="root"
```

Dieser Befehl aktualisiert die Spalte *Password* in allen Datensätzen des Benutzers „*root*“. (Das Passwort können sie nicht direkt in die *user*-Tabelle eingeben, da *mysql* mit verschlüsselten Passwörtern arbeitet!!)

Mit „*flush privileges*“ werden die veränderten Dateien vom SQL-Server neu gelesen.



Unter Access können sie in einer vorhandenen Datenbank mittels „Datei → Externe Daten → Tabellen verknüpfen → ODBC-Datenquellen“ eine Verknüpfung zu den Datenbanken des SQL-Servers erstellen.

Anlegen eines Gastbenutzers, der Leserechte für eine bestimmte Datenbank (dbtest) besitzt:

Tragen sie in der user-Tabelle einen Datensatz für „gast“ ein, wobei in der Hostspalte der Platzhalter % verwendet wird und alle Rechte auf „N“ gesetzt werden. In der db-Tabelle schließlich geben sie dem Benutzer „gast“ die „Select“ Rechte für die Datenbank dbtest.

10.2.1 Einfachere Verwaltung mit phpMyAdmin:

Das Tool phpMyAdmin bietet die Möglichkeit, den SQL-Server via WEB zu verwalten. Die notwendige Grundkonfiguration wird im Verzeichnis /srv/www/htdocs/phpMyAdmin in der Datei config.inc.php durchgeführt:

Nach der Standardinstallation gibt es einen MYSQL Benutzer root ohne PW. Damit kann die Datenbank bereits über PHP voll administriert werden. Aus Sicherheitsgründen sollte entweder der Zugriff auf dieses Verzeichnis nur autorisierten Benutzern erlaubt werden, oder es ist ein spezieller SQL Benutzer anzulegen, der nur eingeschränkte Rechte besitzt:

z.B: User „phplogin“ anlegen:

Im phpMyAdmin Hauptmenü über den Link user den Benutzer phplogin anlegen (für den Rechner „localhost“). In diesem Formular, darf dem Benutzer noch kein Recht verliehen werden. (Diese Rechte würden sich auf alle Datenbanken beziehen) Diesem Benutzer verleiht man die SELECT Rechte in der Datenbank mysql auf folgende Tabellen:

- user
- db
- tables_priv

In der Datei config.inc.php wird dieser Benutzer (mit PW) als „controluser“ eingetragen. Als Authentifizierungsmethode wählt man anstelle config die Methode „http“ oder „cookie“. Den Root-Benutzer in der Zeile darunter muss man entfernen. Der Sinn dieser Methode ist, dass das Paket phpMyAdmin für alle verfügbar gemacht werden kann und das Skript mit nur geringen Rechten sich am SQL Server anmeldet. Über das Loginfenster (bei Methode http) meldet man sich dann mit einer privilegierten Benutzerkennung an und kann die Datenbanken auf die man Zugriffsrechte hat dann verwalten.

Auszug aus config.inc.php:

```

$cfg['Servers'][$i]['host']           = 'localhost'; // MySQL hostname
$cfg['Servers'][$i]['port']          = ''; //MySQL port - leave blank for default port
$cfg['Servers'][$i]['socket']        = ''; // Path to the socket - leave blank for default socket
$cfg['Servers'][$i]['connect_type']  = 'tcp';//How to connect to MySQL server ('tcp' or 'socket')
$cfg['Servers'][$i]['controluser']   = 'phplogin'; // MySQL control user settings
                                        // (this user must have read-only
$cfg['Servers'][$i]['controlpass']    = 'schule'; // access to the "mysql/user"
                                        // and "mysql/db" tables)
$cfg['Servers'][$i]['auth_type']      = 'http'; // Authentication method (config,
                                        // http or cookie based)?
$cfg['Servers'][$i]['user']           = 'root'; // MySQL user
$cfg['Servers'][$i]['password']       = ''; // MySQL password (only needed
                                        // with 'config' auth_type)
    
```



10.3 Zugriffe aus Web-Dokumenten

Um von Webdokumenten auf SQL - Datenbanken zuzugreifen bieten sich 2 Möglichkeiten an: **Verwendung von PHP** (siehe FDB – Projekt, bzw. MYSQL-Dokumentation von Herwig REIDLINGER unter <http://www.pinoe-hl.ac.at/material>), oder die **Einbindung von PERL-SKRIPTS**. Die Details dazu würden alleine ein Seminar füllen, aus dem Grund beschränke ich mich auf ein einfaches Demonstrationsbeispiel, das auf die zuvor erstellte SQL-Datenbank DBTEST zugreift:

Erstellen sie sich ein HTML-Formular mit einer Eingabezeile und einem Startknopf:

```
<html>
<title> Suchformular </title>
<body>
  <form METHOD="post" ACTION="http://localhost/cgi-bin/suche.pl">
    <input type="text" name="suchname">
    <br>
    <input type="submit" value='SUCHEN'>
  </form>
</body>
</html>
```

Danach erstellen sie im Verzeichnis /usr/local/httpd/cgi-bin das Perlskript suche.pl:

```
#!/usr/bin/perl
sub header
{
print "content-type: text/html \n\n";
}
sub form_parse {
  read (STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
  @pairs = split(/&/, $buffer);
  foreach $pair (@pairs)
  {
    ($name, $value) = split(/=/, $pair);
    $value =~ tr/+// ;
    $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
    $FORM{$name} .= "," if ($FORM{$name});
    $FORM{$name} .= $value;
  } # End of foreach
} # End of sub

use DBI;
&header;
&form_parse;
$searchname = $FORM{"suchname"};
$searchname = "F%";
$sel_string = "select * from namen where Name like ".$searchname."";
$dbh = DBI->connect("DBI:mysql:dbtest:192.168.1.254","gast");
$sth = $dbh->prepare($sel_string);
$rv = $sth->execute;

while (($name)=$sth->fetchrow_array) {
  print $name."<br>";
};
print "</body></html>";
$rc = $dbh->disconnect;
```



10.4 Ein Beispiel mit PHP:

Die Formularedatei: (z.B.: formular.html)

```
<html>
<body>
  <form METHOD="post" action="http://localhost/suche_form.php">
    <input type="text" name="suchname">
    <br>
    <input type="submit" value="SUCHEN">
  </form>
</body>
</html>
```

und das zugehörige PHP-Skript (suche_form.php)

```
<html>
<head>
  <title> SUCHE in MYSQL-DATENBANK </title>
</head>
<?php
  $snam=$HTTP_POST_VARS["suchname"];
  mysql_connect('127.0.0.1:3306','gast','');
  $id = mysql_db_query('data','select * from schueler where name like ".$snam.%");
  while ($zeile = mysql_fetch_object($id)) {
    echo '<h4> NAME:      '.$zeile->name.'</h4><br>';
    echo '<h5> VORNAME:   '.$zeile->vorname.'</h5><br>';
    echo '<h6> EMAIL:     '.$zeile->email.'</h6><br><br>';
  }
  mysql_close();
?>
</html>
```