

Debian Server Installation

Evil

m.stroh@softhome.net

7. April 2011

Inhaltsverzeichnis

| | | |
|----------|---------------------------------------------------------------------|-----------|
| 1 | Vorwort | 3 |
| 2 | Fernwartung | 3 |
| 2.1 | SSH (Remote Zugriff) | 3 |
| 2.2 | Webmin (Remote Administration per Web-Interface) | 3 |
| 3 | Fileserver/NAS | 3 |
| 3.1 | Samba (Windows Fileserver) | 3 |
| 3.2 | Proftp (FTP-Server) | 4 |
| 3.3 | MySecureShell (SFTP-Server) | 5 |
| 3.4 | ntfs-3g (NTFS Dateisystem Support) | 6 |
| 3.4.1 | Installation | 6 |
| 3.4.2 | Update | 6 |
| 3.4.3 | Beispiel Partitionierung und mounten | 7 |
| 3.5 | usbmount (Automatisches Einbinden von USB Speichermedien) | 7 |
| 3.6 | Rsync (Datei Synchronisierung) | 8 |
| 4 | Mail Transfer/Web-Server | 8 |
| 4.1 | nullmailer - MTA (Mail Transfer Agent) | 8 |
| 4.2 | lighttp - Web-Server | 10 |
| 5 | Download-Server | 10 |
| 5.1 | MLDonkey (P2P Client) | 10 |
| 5.1.1 | Installation | 10 |
| 5.1.2 | Update | 12 |
| 5.2 | Dauny (Remote Download-Server) | 12 |
| 5.2.1 | Erstinstallation | 12 |
| 5.2.2 | Update | 13 |
| 5.2.3 | Install clive (Debian Squeeze) | 13 |
| 5.2.4 | youtube-dl (Debian Squeeze und Lenny) | 14 |

| | | |
|----------|--------------------------------------------|-----------|
| 5.3 | Pyload (Remote Download Manager) | 14 |
| 6 | Sicherheit | 17 |
| 6.1 | fail2ban (Log-Analysator) | 17 |
| 6.2 | apticron | 18 |
| 7 | Backup | 18 |
| 7.1 | Vorwort | 18 |
| 7.2 | Synchronisierung | 19 |
| 7.2.1 | Homepage | 20 |
| 7.2.2 | Webstorage | 20 |
| 7.3 | Datenquellen | 20 |
| 7.3.1 | Subversion | 20 |
| 7.3.2 | Verzeichnis/Datei | 22 |
| 8 | Monitoring | 25 |
| 8.1 | collectd - Performance-Analyse | 25 |
| 9 | Entwicklung | 27 |
| 9.1 | Latex | 27 |
| 9.1.1 | Installation | 27 |
| 9.1.2 | Auto-Build | 27 |

1 Vorwort

Diese Dokumentation ist eine Erweiterung bzw. der zweite Teil der Dokumentationen 'Debian Server Installation auf einem Thin Client'. Es wird deshalb von einem bestehenden Debian Lenny oder Squeeze System ausgegangen. Das Dokument beschreibt die Installation von verschiedenen Serverdiensten und Programmen um aus einem Thin Client einen vollwertigen NAS zu machen. Dabei werden die Bereiche Fernwartung, Fileserver, Download-Server und Backup-Server abgedeckt.

2 Fernwartung

2.1 SSH (Remote Zugriff)

```
apt-get install ssh
```

```
/etc/ssh/sshd_config [-rw-r--r-- root root]
```

```
#Port 22
Port 1022
# Authentication:
LoginGraceTime 20
PermitRootLogin no
#AllowUsers user1 user2
PermitEmptyPasswords no
```

2.2 Webmin (Remote Administration per Web-Interface)

```
wget http://prdownloads.sourceforge.net/webadmin/webmin_1.530_all.deb
dpkg -i webmin_1.530_all.deb
apt-get -f install
```

```
URL: https://<IPAdress>:10000/
```

3 Fileserver/NAS

3.1 Samba (Windows Fileserver)

```
apt-get install samba smbfs
Arbeitsgruppen-/Domain-Name? Arbeitsgruppe
Soll smb.conf so abgeändert werden, dass per DHCP angebotene WINS-Einstellungen verwendet werden? Nein
```

```
/etc/samba/smb.conf [-rw-r--r-- root root]
```

```

workgroup = Arbeitsgruppe
security = user
socket options = TCP_NODELAY SO_RCVBUF=16384 SO_SNDBUF=16384 SO_KEEPALIVE IPTOS_LOWDELAY

[homes]
  browseable = yes
  read only = no

[data]
  comment = Datenverzeichnis
  path = /data
  guest ok = yes
  writable = yes
  browsable = yes
  read only = no

[print$]
  browseable = no

adduser <Benutzer>
smbpasswd -a <Benutzer>
/etc/init.d/samba restart

```

3.2 Proftpd (FTP-Server)

```

mkdir /data/ftp
apt-get install proftpd
Proftpd starten? Servermodus

/etc/proftpd/proftpd.conf [-rw-r--r-- root root]

# AuthOrder mod_auth_pam.c* mod_auth_unix.c
AuthOrder mod_auth_file.c
AuthUserFile /etc/proftpd/proftpd.passwd

TransferLog /var/log/proftpd/xferlog
SystemLog /var/log/proftpd/proftpd.log
#defaultroot /data/ftp
defaultroot /data/

#TransferRate STOR 200 user ftpuser # max. KBytes/Sekunde for Upload
#TransferRate RETR 35 user ftpuser # max. KBytes/Sekunde for Download
#MaxClientsPerUser 1 # Es ist nicht mehr als 1 Client pro Nutzer erlaubt
#MaxClientsPerHost 1 # Es ist nur ein Client pro Host erlaubt
#MaxHostsPerUser 3 # Es sind nur 3 Hosts per User erlaubt

```

```

adduser --disabled-login ftpuser
export FTPUSERID='id -u ftpuser' echo $FTPUSERID
ftpasswd --passwd --name ftpuser --home /data/ftp --shell /bin/sh --uid $FTPUSERID --file /etc/proftpd/proftpd.passwd
Password: „PASS“
/etc/init.d/proftpd restart

```

```

addgroup ftp
adduser <Benutzer> ftp
chgrp ftp /data/ftp
chmod g+w /data/ftp

```

3.3 MySecureShell (SFTP-Server)

```
/etc/apt/sources.list [-rw-r--r-- root root]
```

```

# only i386 supported
deb http://mysecureshell.free.fr/repository/index.php/debian testing main

```

```

gpg --keyserver hkp://pool.sks-keyservers.net --recv-keys E328F22B; gpg --export E328F22B
| sudo apt-key add -
apt-get install mysecureshell

```

Neuen Benutzer anlegen:

```
useradd -s /bin/MySecureShell <Benutzernamen>
```

Bestehenden Benutzer modifizieren:

```
usermod -s /bin/MySecureShell <Benutzernamen>
```

Für die Beispielkonfiguration wurde der Benutzername sftp-user verwendet.

```
/etc/ssh/sftp_config [-rw-r--r-- root root]
```

```

## MySecureShell Configuration File ##
#Default rules for everybody
<Default>
    GlobalDownload      42k      #total speed download for all clients
                                # o -> bytes  k -> kilo bytes  m -> mega bytes
    GlobalUpload        0       #total speed download for all clients (0 for unlimited)
    Download            0       #limit speed download for each connection
    Upload              0       #unlimit speed upload for each connection
    StayAtHome          true      #limit client to his home
    VirtualChroot       true      #fake a chroot to the home account
    LimitConnection     10      #max connection for the server sftp
    LimitConnectionByUser 2       #max connection for the account
    LimitConnectionByIP 2       #max connection by ip for the account
#    Home                /home/$USER #override home of the user but if you want you can use
                                # environment variable (ie: Home /home/$USER)
    Home                /data
    IdleTimeOut         5m      #(in second) deconnect client is idle too long time
    ResolveIP           true     #resolve ip to dns

```

```

# IgnoreHidden true #treat all hidden files as if they don't exist
# DirFakeUser true #Hide real file/directory owner (just change displayed permissions)
# DirFakeGroup true #Hide real file/directory group (just change displayed permissions)
# DirFakeMode 0400 #Hide real file/directory rights (just change displayed permissions)
# HideFiles "(lost|+found|public_html)$" #Hide file/directory which match
# HideNoAccess true #Hide file/directory which user has no access
# MaxOpenFilesForUser 20 #limit user to open x files on same time
# MaxWriteFilesForUser 10 #limit user to x upload on same time
# MaxReadFilesForUser 10 #limit user to x download on same time
# DefaultRights 0640 0750 #Set default rights for new file and new directory
# MinimumRights 0400 0700 #Set minimum rights for files and dirs

# PathDenyFilter "~\." #deny upload of directory/file which match this extended POSIX regex

# ShowLinksAsLinks false #show links as their destinations
# ConnectionMaxLife 1d #limits connection lifetime to 1 day

# Charset "ISO-8859-15" #set charset of computer
# GMTTime +1 #set GMT Time (change if necessary)
</Default>

#Rules only for group ftp
<Group sftp-user>
  Download 32k
  LogFile /var/log/sftp-server_sftp-user.log #Change logfile
# ExpireDate "2007-02-28 18:31:01"
</Group>

```

3.4 ntfs-3g (NTFS Dateisystem Support)

3.4.1 Installation

```

apt-get install ntfsprogs ntfs-3g
modprobe fuse

```

```

/etc/modules [-rw-r--r-- root root]

```

```

fuse

```

3.4.2 Update

```

wget http://tuxera.com/opensource/ntfs-3g-2010.10.2.tgz
unp ntfs-3g-2010.10.2.tgz cd ntfs-3g-2010.10.2 apt-get install gcc make
./configure
make
mv /usr/bin/ntfs-3g /usr/bin/ntfs-3g.old
mv /usr/bin/ntfs-3g.probe /usr/bin/ntfs-3g.probe.old
cp src/.libs/ntfs-3g /usr/bin/
cp src/.libs/ntfs-3g.probe /usr/bin
cp -av libntfs-3g/.libs/libntfs-3g.so* /lib/
ldconfig
echo "ntfs-3g hold" | dpkg --set-selections

```

3.4.3 Beispiel Partitionierung und mounten

Formatierung Partition: fdisk /dev/sdb

n

p

1

<Enter>

<Enter>

t

7

p

w

mkntfs -v -f -L NTFS /dev/sdb1

fdisk -l /dev/sda

mkdir /ntfs-data

mount -t ntfs-3g /dev/sdb1 /ntfs-data

/etc/fstab [-rw-r--r-- root root]

```
LABEL=NTFS /ntfs-data ntfs-3g defaults,noatime,noexec 0 0
```

mount -a

3.5 usbmount (Automatisches Einbinden von USB Speichermedien)

apt-get install usbmount

wget <http://ftp.evilk99.ev.funpic.de/linux/ThinClient/Server/usbmount-ntfs3g.patch>

usbmount-ntfs3g.patch [-rw-r--r-- root root]

```
--- /usr/share/usbmount/usbmount.old 2009-06-01 11:11:02.000000000 +0200
+++ /usr/share/usbmount/usbmount      2009-06-01 11:17:35.000000000 +0200
@@ -45,6 +45,7 @@
 MOUNTOPTIONS=""
 FS_MOUNTOPTIONS=""
 VERBOSE="no"
+USE_NTFS3G="no"

# Read configuration file.
if test -r /etc/usbmount/usbmount.conf; then
@@ -119,7 +120,11 @@

        # Mount the filesystem.
log info "executing command: mount -t$fstype ${options:+-o$options} $DEVNAME $mountpoint"
- mount "-t$fstype" "${options:+-o$options}" "$DEVNAME" "$mountpoint"
+ if test $fstype == "ntfs" && expr "$USE_NTFS3G" : "[yY]" > /dev/null; then
+     mount "-tntfs-3g" "${options:+-o$options}" "$DEVNAME" "$mountpoint"
+ else
+     mount "-t$fstype" "${options:+-o$options}" "$DEVNAME" "$mountpoint"
+ fi

# Determine vendor and model.
```

```

                vendor=
@@ -166,6 +171,11 @@

    elif test "$1" = remove; then

+   # if we utilize ntfs-3g, we need to add the fuseblk fstype to allow proper unmounting
+   if [ 'expr "$USE_NTFS3G" : "[yY]"' -eq "1" ]; then
+       FILESYSTEMS="$FILESYSTEMS fuseblk"
+   fi
+
    # A block or partition device has been removed.
    # Test if it is mounted.
    while read device mountpoint fstype remainder; do

cp /usr/share/usbmount/usbmount /usr/share/usbmount/usbmount.old
patch /usr/share/usbmount/usbmount usbmount-ntfs3g.patch

/etc/usbmount/usbmount.conf [-rw-r--r-- root root]

FILESYSTEMS="ext2 ext3 vfat ntfs"
USE_NTFS3G="yes"
MOUNTOPTIONS="sync,noexec,nodev,noatime"
# rw on vfat, r on ntfs
FS_MOUNTOPTIONS="-fstype=vfat,gid=floppy,dmask=0000,fmask=0111,utf8 -fstype=ntfs\
,gid=0,uid=0,nls=utf8,dmask=0000,fmask=0111"

/etc/samba/smb.conf [-rw-r--r-- root root]

[USB]
comment =USB Geräte
path = /media
guest ok = yes
writeable = yes
browseable = yes
read only = no

/etc/init.d/samba restart

```

3.6 Rsync (Datei Synchronisierung)

```
apt-get install rsync
```

4 Mail Transfer/Web-Server

4.1 nullmailer - MTA (Mail Transfer Agent)

Bei Debian kommt als Standard MTA exim4 zu Einsatz.

Will man aber nur Mail versenden können ohne weitere Server-Funktionen, so bietet sich der MTA „nullmailer“ an. Als Beispiel soll hier eine Einrichtung mit einem Yahooemail Account (username@ymail.com) dienen. Die „nullmailer“ Version hat allerdings so seine Probleme mit den Einschränkungen bzw. Bedingungen von Yahooemail. Darum kommt

hier eine von mir modifizierte Version zum Einsatz.

```
Lenny: wget http://evilx99.ev.funpic.de/linux/nullmailer/nullmailer_1.04-2.0_i386.deb
Squeeze: wget http://evilx99.ev.funpic.de/linux/nullmailer/squeeze/nullmailer_1.04-2.0_i386.deb
dpkg -i nullmailer_1.04-2.0_i386.deb
```

Mail-Name für Ihr System: MyMailName

Smarthosts: smtp.mail.yahoo.de smtp --port=25 --user=username@ymail.com --pass=password

```
/etc/mailname [-rw-r--r-- root root]
```

```
MyMailName
```

```
/etc/nullmailer/defaulthost [-rw-r--r-- root root]
```

```
homeserver
```

```
/etc/nullmailer/defaultdomain [-rw-r--r-- root root]
```

```
at
```

```
/etc/nullmailer/remotes [-rw-r--r-- root root]
```

```
smtp.mail.yahoo.de smtp --port=25 --user=username@ymail.com --pass=password
```

```
/etc/nullmailer/nullmailer.conf [-rw-r--r-- root root]
```

```
NULLMAILER_USER=username
```

```
NULLMAILER_HOST=ymail.com
```

```
#NULLMAILER_NAME=
```

```
#NULLMAILER_SUSER=
```

```
#NULLMAILER_SHOST=
```

```
#NULLMAILER_QUEUE=
```

```
#
```

```
#Options
```

```
#USE_NAME_ADDRESS_STYLE=1
```

```
#IGNORE_HEADER_FIELD_FROM=1
```

```
#IGNORE_HEADER_FIELD_MID=1
```

```
#IGNORE_HEADER_FIELD_RPATH=1
```

```
#HEADER_ADD_TO=1
```

```
MAIL_HEADER_REMOVE_FROM=1
```

```
/etc/aliases [-rw-r--r-- root root]
```

```
# /etc/aliases
```

```
root: username@ymail.com
```

```
root@homeserver.at: username@ymail.com
```

```
user1: username@ymail.com
```

```
user1@homeserver.at: username@ymail.com
```

Test Mail:

```
echo „This is a test mail“ | sendmail -F AbsenderName -f username@ymail.com username@ymail.com
```

Diagnose:

```
/var/spool/nullmailer/
```

4.2 lighttpd - Web-Server

```
apt-get install lighttpd php5-cgi
```

```
/etc/lighttpd/lighttpd.conf [-rwxr-xr-x root staff]
```

```
server.modules = (
    "mod_access",
    "mod_alias",
    "mod_compress",
    "mod_redirect",
    "mod_fastcgi",
#    "mod_rewrite",
)

fastcgi.server = ( ".php" => ((
"bin-path" => "/usr/bin/php-cgi",
"socket" => "/tmp/php-fastcgi.sock",
    "bin-environment" => (
        "PHP_FCGI_CHILDREN" => "0",
"PHP_FCGI_MAX_REQUESTS" => "1000"
    ),
),
    )))
```

```
/etc/init.d/lighttpd restart
```

5 Download-Server

5.1 MLDonkey (P2P Client)

5.1.1 Installation

```
apt-get install mldonkey-server
```

MLDonkey beim Hochfahren starten? Ja

```
/etc/init.d/mldonkey-server stop
cp /var/lib/mldonkey/downloads.ini /var/lib/mldonkey/downloads.old
```

```
/var/lib/mldonkey/downloads.ini [-rw-r--r-- root root]
```

```
allowed_ips = [
    "192.168.0.0-192.168.0.255";
    "127.0.0.1";]
```

```
max_hard_upload_rate = 10
```

```
temp_directory="/data/incoming/temp"
```

```
shared_directories = [
    {
        dirname = shared
        networks = []
        strategy = all_files
        priority = 0
    };
    {
        dirname = "/data/incoming/files"
        networks = []
        strategy = incoming_files
        priority = 0
    };
    {
        dirname = "/data/incoming/directories"
        networks = []
        strategy = incoming_directories
        priority = 0
    };
];]
```

```
max_displayed_results = 200
```

```
mkdir /data /data/incoming
cd /data/incoming
mkdir temp files directories
chown -R mldonkey:mldonkey /data/incoming
```

Hinzufügen von lokalen Benutzern zur Gruppe mldonkey:

```
adduser <Benutzer> mldonkey
chmod a+w /data/incoming/files/ /data/incoming/directories/
```

```
/etc/init.d/mldonkey-server start
```

5.1.2 Update

```
/etc/init.d/mldonkey-server stop
wget http://surfnet.dl.sourceforge.net/sourceforge/mldonkey/mldonkey-2.9.7.static.i386-Linux_glibc-2.3.6.tar.bz2
unp mldonkey-2.9.7.static.i386-Linux_glibc-2.3.6.tar.bz2
rm /usr/bin/mlnet.old
mv /usr/bin/mlnet /usr/bin/mlnet.old
cp mldonkey-distrib-2.9.7/mlnet /usr/bin/
rm -r mldonkey-distrib-2.9.7/
/etc/init.d/mldonkey-server start

echo "mldonkey-server hold" | dpkg --set-selections
```

5.2 Dauny (Remote Download-Server)

5.2.1 Erstinstallation

```
apt-get install wget ctorrent spidermonkey-bin
mkdir /data/incoming/dauny
```

```
wget http://evilx99.ev.funpic.de/dauny/dauny/dauny_2.0.7_i386.deb
dpkg --install dauny_2.0.7_i386.deb
```

oder

```
wget http://evilx99.ev.funpic.de/dauny/dauny/dauny-2.0.5beta-i386.tar.gz
tar xzvf dauny-2.0.5beta-i386.tar.gz
cd dauny
./install-debian.sh
```

```
Vollständiger Name []:
Raumnummer []:
Telefon geschäftlich []:
Telefon privat []:
Sonstiges []:
Is the information correct? [Y/n] y
```

```
user name: <Name>
enter password: <Pass>
confirm password: <Pass>
Downloadlimit [MB] (0 ... unlimited): <Enter>
user homedir: /data/incoming/dauny
User <Name> added to userdatafile
```

```
add another user ? [Y/N] n
```

```
Squeeze(dependency-based booting): inserv dauny
```

```
/etc/dauny/dauny.ini [-rw-r--r-- root root]
```

```
AccessPermission=775
```

```
/etc/init.d/dauny start
```

```
adduser <Benutzer> dauny
```

```
chmod g+w /data/incoming/dauny/
```

5.2.2 Update

```
/etc/init.d/dauny stop
```

```
wget http://evilx99.ev.funpic.de/dauny/dauny/dauny-2.0.23beta-i386.tar.gz
```

```
tar xzvf dauny-2.0.23beta-i386.tar.gz
```

```
cd dauny
```

```
mv /usr/local/bin/dauny /usr/local/bin/dauny.old
```

```
cp dauny /usr/local/bin
```

```
/etc/init.d/dauny start
```

5.2.3 Install clive (Debian Squeeze)

```
/etc/init.d/dauny stop
```

```
apt-get install clive
```

```
/etc/dauny/dauny.ini [-rw-r--r-- root root]
```

```
#YoutubeProgram
```

```
# 0=disable, 1=clive, 2=cclive, 3=youtube-dl, 9=default program
```

```
YoutubeProgram=1
```

```
#
```

```
CLiveExe=clive
```

```
# mp4 480p Format
```

```
#CLiveParameter=--format=mp4
```

```
# max. Format
```

```
CLiveParameter=--format=best
```

```
/etc/init.d/dauny start
```

5.2.4 youtube-dl (Debian Squeeze und Lenny)

```
/etc/init.d/dauny stop
```

```
wget http://ftp.at.debian.org/debian/pool/main/y/youtube-dl/youtube-dl_2011.01.30-2_all.deb  
dpkg -i youtube-dl_2011.01.30-2_all.deb
```

```
/etc/dauny/dauny.ini [-rw-r--r-- root root]
```

```
#YoutubeProgram  
# 0=disable, 1=cclive, 2=cclive, 3=youtube-dl, 9=default program  
YoutubeProgram=9  
#  
# Set executeable filename for youtube-dl  
Youtube-dlExe=youtube-dl
```

```
/etc/init.d/dauny start
```

5.3 Payload (Remote Download Manager)

```
apt-get install python python-crypto python-pycurl python-jinja2 python-imaging python-  
django tesseract-ocr tesseract-ocr-eng spidermonkey-bin openssl unzip unrar
```

```
wget http://get.payload.org/get/src/ -O payload.zip
```

```
unzip payload.zip
```

```
mv payload /usr/share/
```

```
rm payload.zip
```

```
ln -s /usr/share/payload/pyLoadCore.py /usr/bin/pyLoadCore
```

```
ln -s /usr/share/payload/pyLoadCli.py /usr/bin/pyLoadCli
```

```
adduser payload --disabled-login
```

```
Vollständiger Name []:
```

```
Raumnummer []:
```

```
Telefon geschäftlich []:
```

```
Telefon privat []:
```

```
Sonstiges []:
```

```
Sind die Informationen korrekt? [J/n] j
```

```
mkdir /var/lib/payload/ /var/log/payload /data /data/incoming /data/incoming/payload
```

```
chown payload:payload /var/lib/payload/ /var/log/payload /data/incoming/payload
```

```
echo /var/lib/payload > /usr/share/payload/module/config/configdir
```

```
/usr/share/payload/module/config/default.conf [-rw-r--r-- root root]
```

```
log - "Log":
```

```
    bool file_log : "File Log" = True
```

```
folder log_folder : "Folder" = /var/log/pyload

permission - "Permissions":
  bool change_user : "Change user of running process" = True
  str user : "Username" = payload

/usr/bin/pyLoadCore

This is your first start, running configuration assistent now.
Choose your Language / Wähle deine Sprache ([en], de, it, pl, fr, cs, es): de
Wenn du bereit für den System-Check bist, drücke enter. <enter>
```

```
## System Check ##
Python Version: OK
pycurl: OK
sqlite3: OK
```

```
pycrypto: OK
OpenSSL: fehlt
```

```
py-imaging: OK
tesseract: OK
```

```
PyQt4: fehlt
```

```
django: OK
JS engine: OK
```

```
System-Check beendet, drücke Enter um deinen Status Bericht zu sehen. <enter>
Mit Setup fortfahren? ([y]/n): y
Config Pfad ändern? (y/[n]): n
Erstelle Grundeinstellungen? ([y]/n): y
Benutzername [User]: payload
Passwort: payload
Passwort (nochmal): payload
Sprache ([en], de, it, pl, es, cs, fr): de
Download Ordner [Downloads]: /data/incoming/pyload
Maximale parallele Downloads [3]: 3
Benutze Reconnect? (y/[n]): n
Konfiguriere Webinterface? ([y]/n): y
Benutzername [User]: payload
Passwort: payload
Passwort (nochmal): payload
Aktiviere Webinterface? ([y]/n): y
```

Adresse [0.0.0.0]: <enter>
Port [8000]: <enter>
Drücke Enter zum Beenden und starte pyLoad neu <enter>

```
chown -R payload:payload /var/lib/payload/
```

```
/etc/init.d/payload [-rwxr-xr-x root root]
```

```
#!/bin/sh
```

```
#
```

```
# Written by Evil <m.stroh@softhome.net>
```

```
#
```

```
### BEGIN INIT INFO
```

```
# Provides:          payload
```

```
# Required-Start:    $network $local_fs
```

```
# Required-Stop:     $network $local_fs
```

```
# Default-Start:     2 3 4 5
```

```
# Default-Stop:      0 1 6
```

```
# Short-Description: Server for different kinds downloads.
```

```
# Description:       Server for different kinds downloads.
```

```
### END INIT INFO
```

```
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
```

```
DAEMON=/usr/bin/pyLoadCore
```

```
NAME=payload
```

```
DESC="remote download server"
```

```
test -f $DAEMON || exit 0
```

```
set -e
```

```
case "$1" in
```

```
  start)
```

```
    echo -n "Starting $DESC: "
```

```
    start-stop-daemon --start --background --pidfile /var/run/$NAME.pid -o -m --exec $DAEMON
```

```
    echo "$NAME."
```

```
    ;;
```

```
  stop)
```

```
    echo -n "Stopping $DESC: "
```

```
    start-stop-daemon --stop -o --pidfile /var/run/$NAME.pid
```

```
    echo "$NAME."
```

```
    ;;
```

```
  restart)
```

```
    echo -n "Restarting $DESC: "
```



```
$0 stop
sleep 1
$0 start
;;
*)
echo "Usage: /etc/init.d/$NAME {start|stop|restart}" >&2
exit 1
;;
esac
```

```
exit 0
```

```
chmod +x /etc/init.d/pyload
```

```
Lenny: update-rc.d pyload start 98 2 3 4 5 . stop 02 0 1 6 .
```

```
Squeeze(dependency-based booting): insserv pyload
```

```
/etc/init.d/pyload start
```

6 Sicherheit

6.1 fail2ban (Log-Analysator)

Wichtiger Hinweis: <http://www.ossec.net/main/attacking-log-analysis-tools>

```
apt-get install fail2ban
```

```
/etc/fail2ban/jail.conf [-rw-r--r-- root root]
```

```
ignoreip = 127.0.0.1
```

```
bantime = 600
```

```
maxretry = 3
```

```
[ssh]
```

```
enabled = true
```

```
port = ssh
```

```
filter = sshd
```

```
logpath = /var/log/auth.log
```

```
maxretry = 3
```

```
[proftpd]
```

```
enabled = true
```

```
port = ftp,ftp-data,ftps,ftps-data
```

```
filter = proftpd
```

```
logpath = /var/log/proftpd/proftpd.log
maxretry = 3
```

```
/etc/init.d/fail2ban restart
```

```
fail2ban-client status
fail2ban-client status proftpd
```

6.2 apticron

```
apt-get install apticron
```

```
/etc/apticron/apticron.conf [-rw-r--r-- root root]
```

```
#EMAIL="root"
EMAIL="user@ymail.com"
SYSTEM="Server Name"
```

7 Backup

7.1 Vorwort

Backup ist ein wichtiges Thema im Softwarebereich. Ein Backup Konzept sollte dabei immer für den „worst case“ ausgelegt sein. Zur Datensicherung wird im professionellen Bereich ein Bandlaufwerk (z.B. mit 800 GB) eingesetzt. Im privaten Bereich kommen aus Kostengründen andere Speichermedien in Frage. Meist werden DVD- oder RAM-Disks verwendet. Nachteil ist die begrenzte Speicherkapazität und, dass der Backupmechanismus nicht automatisierbar ist. Im Idealfall sollte man sich nicht selbst um die regelmäßigen Backups kümmern müssen, sie sollten automatisch erstellt werden. Deshalb schlage ich ein Backup auf einer Festplatte vor. Dies mag zwar nicht das sicherste Speichermedium sein. Aber wenn die Daten auf zwei Festplatten gespeichert sind (Arbeitssystem und Backupssystem) ist die Datensicherheit bereits sehr hoch. Das „worst case“ Szenario ist aber wenn mein Haus abbrennt. Deshalb sollte ein Backup idealerweise örtlich getrennt sein. Daher schlage ich vor sehr wichtige Daten im Internet zu speichern. Natürlich ist bei dem Speichermedium die Kapazität begrenzt aber auch vor allem die upload Geschwindigkeit. Deshalb sollte man nur die wirklich wichtige Daten auf diese Weise sichern. Ein Vorteil der Methode ist, dass die Daten komplett automatisiert synchronisiert werden können. Vielfach wird das Verschlüsseln von Daten eine Notwendigkeit sein um private Daten vor Spionage zu schützen. Die folgende Beschreibung soll eine Möglichkeit geben, das beschriebene Backupzenario zu implementieren. Dabei verwende ich ein 2 GB gratis Webstorage von MyDrive (<http://www.mydrive.ch/>). Ein lokale Verzeichnis auf einer Festplatte wird per WebDAV mit dem Webstorage synchronisiert.

Folgende Sicherungen werden beschrieben:

- Sicherung einer Homepage auf eine Festplatte.
- Sicherung eines Subversion Repositories auf einem Webservice.
- Sicherung eines Verzeichnisses oder Datei auf einem Webservice.

Zum Komprimieren und Verschlüsseln der Daten wird 7-Zip verwendet. Bei Systemen mit wenig Arbeitsspeicher sollte unbedingt der Parameter „-mx3“ verwendet werden (Komprimierungsstufe).

7.2 Synchronisierung

```
mkdir /data
mkdir /data/bkup
cd /data/bkup
mkdir scripts tmp web homepage log
```

```
apt-get install sitecopy
mkdir /var/lib/sitecopy
chmod 700 /var/lib/sitecopy
touch /etc/sitecopy
chmod 600 /etc/sitecopy
```

```
/etc/sitecopy [-rw----- root root]
```

```
site Backup
server webdav.mydrive.ch
protocol webdav
remote /Backup
local /data/bkup/web
username USERNAME
password PASSWORD
```

```
site Homepage
server ftp.name.ev.funpic.de
protocol ftp
remote ~/
local /data/bkup/homepage/
username USERNAME
password PASSWORD
symlinks follow
```

Initialisierung (Welche Dateien liegen am Web):

```
sitecopy -rcfile=/etc/sitecopy -storepath=/var/lib/sitecopy -f Backup
sitecopy -rcfile=/etc/sitecopy -storepath=/var/lib/sitecopy -f Homepage
```

7.2.1 Homepage

```
/data/bkup/scripts/make_homepage_backup [-rwxr--r-- root root]
```

```
#!/bin/bash
```

```
#
```

```
# Ablauf:
```

```
# 1. Inhalte aktualisieren
```

```
# 2. Dateien lokal sichern
```

```
#
```

```
sitecopy --rcfile=/etc/sitecopy --storepath=/var/lib/sitecopy -f Homepage \  
> /data/bkup/log/homepage.log 2>/dev/null
```

```
sitecopy --rcfile=/etc/sitecopy --storepath=/var/lib/sitecopy -s Homepage \  
>> /data/bkup/log/homepage.log 2>/dev/null
```

```
/etc/crontab [-rwxr--r-- root root]
```

```
# Every day at 4:10 Backup the homepage down to local disk
```

```
10 4 * * 0 root /data/bkup/scripts/make_homepage_backup > /dev/null 2>&1
```

7.2.2 Webstorage

```
/etc/crontab [-rwxr--r-- root root]
```

```
# Every day at 5:20 the local files will be synchronized with the web storage
```

```
20 5 * * * root sitecopy --rcfile=/etc/sitecopy --storepath=/var/lib/sitecopy\  
-u Backup > /dev/null 2>&1
```

```
/etc/backup_password [-rw----- root root]
```

```
PASSWORD
```

7.3 Datenquellen

7.3.1 Subversion

```
/data/bkup/scripts/make_svn_backup [-rwxr--r-- root root]
```

```
#!/bin/bash
```

```
#
```

```
# Parameter 1: svn repository path, for example /var/lib/svn
```

```
# Parameter 2: file name for archive, for example svn_repository-Rev
```

```
# Parameter 3: email address for notification
```

```

#
function Exit1
{
echo
echo -n $0: $ERROR!
echo
        exit
}

if ! svnlook youngest $1 > /dev/null ; then
ERROR="svn repository invalid"
Exit1
fi
if [ -z $2 ]; then
ERROR="backup filename missing"
Exit1
fi
if [ ! -f /etc/backup_password ]; then
ERROR="backup password file '/etc/backup_password' missing"
Exit1
fi
PASSWORD='cat /etc/backup_password'
TITLE="Backup subversion repository '$1'"
REVISION='svnlook youngest $1'
FILE=$2-$REVISION.dump
DEST_FILE=/data/bkup/web/$FILE.7z
INFO_FILE=/data/bkup/web/$FILE.txt
TIME_FILE=/data/bkup/tmp/$FILE.time
DATE='date +%d.%m.%G'
PC='uname -n'
COMP_PROG='7z | grep 7-Zip'

if [ -f $DEST_FILE ]
then
echo "Web backup '$TITLE' ($DEST_FILE) is up to date!"
else
echo "Creating '$TITLE' to $DEST_FILE"

/usr/bin/time -f %e -o $TIME_FILE -- svnadmin dump $1 2>/dev/null | \
7z a -t7z -mx3 -si$FILE -p$PASSWORD $DEST_FILE > /dev/null
echo ""
HASH1='md5sum $DEST_FILE | cut -f 1 -d " "'
TIME='cat $TIME_FILE'

```

```

rm $TIME_FILE

SIZE='du -m $DEST_FILE | cut -f 1'
echo "PC: $PC" > $INFO_FILE
echo "Backup: $TITLE" >> $INFO_FILE
echo "Date: $DATE" >> $INFO_FILE
echo "File: $FILE" >> $INFO_FILE
echo "Program: $COMP_PROG" >> $INFO_FILE
echo "Encrypted: yes" >> $INFO_FILE
echo "Size: $SIZE MB" >> $INFO_FILE
echo "Hash: $HASH1" >> $INFO_FILE
echo "Backup Time: $TIME s " >> $INFO_FILE
if [ ! -z $3 ]; then
    cat $INFO_FILE | mail -s "Backup: $TITLE" $3
fi
cat $INFO_FILE
fi

/etc/crontab [-rwxr--r-- root root]

# Every day at 3:00 svn repository backup will be created
0 3 * * * root /data/bkup/scripts/make_svn_backup /var/lib/svn/ \
svn_latex_repository-Rev username@ymail.com > /dev/null 2>&1

```

7.3.2 Verzeichnis/Datei

```

/data/bkup/scripts/make_folder_backup [-rwxr--r-- root root]

#!/bin/bash
#
# Parameter 1: folder or file to backup
# Parameter 2: file name for archive
# Parameter 3: email address for notification
#
function Exit1
{
    echo
    echo -n $0: $ERROR!
    echo
    exit
}
if [ -z $1 ]; then
    ERROR="filename or directory to backup missing"
    Exit1
fi

```

```

if [ -z $2 ]; then
    ERROR="backup filename missing"
    Exit1
fi
if [ ! -f /etc/backup_password ]; then
    ERROR="backup password file '/etc/backup_password' missing"
    Exit1
fi

PASSWORD='cat /etc/backup_password'
TITLE="Backup of '$1'"
# Destination file
FILE="$2.7z"
TEMP_FILE=/data/bkup/tmp/$FILE
DEST_FILE=/data/bkup/web/$FILE
INFO_FILE=/data/bkup/web/$FILE.txt
MD5_FILE=/data/bkup/web/$FILE.md5
TIME_FILE=/data/bkup/tmp/$FILE.time
MD5_TEMP_FILE=/data/bkup/tmp/$FILE.md5
BACKUP_FOLDER=$1
PARAMETER=""
if [ -d "$1" ]; then
    echo Backup folder \"$1\"
    BACKUP_FOLDER="$1/*"
    PARAMETER="-r"
    echo creating md5 file
    find "$1" ! -type d -print0 | xargs -0 md5sum > $MD5_TEMP_FILE
else
    if [ -f "$1" ]; then
        echo Backup file \"$1\"
        BACKUP_FOLDER="$1"
        md5sum "$1" > $MD5_TEMP_FILE
    else
        ERROR="backup file not found"
        Exit1
    fi
fi
if [ -f $INFO_FILE ]; then
    HASH1='md5sum $MD5_TEMP_FILE | cut -f 1 -d " "'
    HASH2='cat $INFO_FILE | grep Hash | cut -f 2 -d " "'
    echo Hash1=$HASH1
    echo Hash2=$HASH2
else
    echo md5 file '$MD5_FILE' destination not found!

```

```

    HASH1='md5sum $MD5_TEMP_FILE | cut -f 1 -d " "'
    HASH2="0"
fi
if [ $HASH1 = $HASH2 ]; then
    echo
    echo "Web backup is up to date!"
    rm $MD5_TEMP_FILE
    exit
fi

DATE='date +%d.%m.%G'
PC='uname -n'
COMP_PROG='7z | grep 7-Zip'
#Compress files
/usr/bin/time -f %e -o $TIME_FILE -- 7z a -t7z -mx3 -p$PASSWORD \
$PARAMETER $TEMP_FILE "$BACKUP_FOLDER"

TIME='cat $TIME_FILE'
rm $TIME_FILE

SIZE='du -m $TEMP_FILE | cut -f 1'
echo "PC: $PC" > $INFO_FILE
echo "Backup: $TITLE" >> $INFO_FILE
echo "Date: $DATE" >> $INFO_FILE
echo "File: $FILE" >> $INFO_FILE
echo "Program: $COMP_PROG" >> $INFO_FILE
echo "Encrypted: yes" >> $INFO_FILE
echo "Size: $SIZE MB" >> $INFO_FILE
echo "Hash: $HASH1" >> $INFO_FILE
echo "Backup Time: $TIME s " >> $INFO_FILE
mv $TEMP_FILE $DEST_FILE
rm $MD5_TEMP_FILE
if [ ! -z $3 ]; then
    cat $INFO_FILE | mail -s "Backup: $TITLE" $3
fi
cat $INFO_FILE

/etc/crontab [-rwxr--r-- root root]

# Every day at 3:30 a local folder backup will be created
30 3 * * * root /data/bkup/scripts/make_folder_backup \
/home/evil/Eigene\ Dateien/ Evil-EigendeDateien XXX@XXX.XX > /dev/null 2>&1

```


8 Monitoring

8.1 collectd - Performance-Analyse

apt-get install collectd rrdtool

```
/etc/collectd/collectd.conf [-rw-r--r-- root root]
```

```
# Config file for collectd(1).
#
# Some plugins need additional configuration and are disabled by default.
# Please read collectd.conf(5) for details.
#
# You should also read /usr/share/doc/collectd/README.Debian.plugins before
# enabling any more plugins.

#Hostname "localhost"
FQDNLookup false
BaseDir "/var/lib/collectd"
PluginDir "/usr/lib/collectd"
#TypesDB "/usr/lib/collectd/types.db" "/etc/collectd/my_types.db"
Interval 10
#ReadThreads 5

#LoadPlugin logfile
LoadPlugin syslog

#<Plugin logfile>
#     LogLevel "info"
#     File STDOUT
#     Timestamp true
#</Plugin>

<Plugin syslog>
    LogLevel info
</Plugin>
LoadPlugin cpu
LoadPlugin df
LoadPlugin disk
#LoadPlugin entropy
LoadPlugin exec
LoadPlugin interface
LoadPlugin irq
LoadPlugin load
LoadPlugin memory
```

```

LoadPlugin processes
LoadPlugin rrdtool
LoadPlugin swap
LoadPlugin users

<Plugin df>
    Device "/dev/sda2"
    Device "/dev/sda4"
    Device "/dev/shm"
    IgnoreSelected false
</Plugin>

<Plugin disk>
    Disk "sda"
    Disk "/sda[24]/"
    IgnoreSelected false
</Plugin>

<Plugin interface>
    Interface "eth0"
    IgnoreSelected false
</Plugin>

<Plugin irq>
    Irq 7
    Irq 8
    Irq 9
    IgnoreSelected true
</Plugin>

<Plugin exec>
    Exec "nobody:dialout" "/usr/local/bin/roomtemp"
</Plugin>

<Plugin rrdtool>
    DataDir "/var/lib/collectd/rrd"
    CacheTimeout 120
    CacheFlush 3600

#
# The following settings are rather advanced
# and should usually not be touched:
#     StepSize 10
#     HeartBeat 20

```

```

#      RRARows 1200
#      RRATimespan 158112000
#      XFF 0.1
</Plugin>

Include "/etc/collectd/thresholds.conf"

mkdir /var/www/collectd
cd /var/www/collectd/
wget http://pommi.nethuis.nl/storage/software/cgp/cgp-0.3.tgz
tar xzvf cgp-0.3.tgz
mv cgp-0.3 cgp

```

9 Entwicklung

9.1 Latex

9.1.1 Installation

```
apt-get install texlive texlive-lang-german
```

Kompilierung: pdflatex latex.tex

9.1.2 Auto-Build

Sinnvoll ist es Latex Projekte in einem Subversion Repository zu verwalten. Ziel soll es nun sein täglich zu überprüfen ob sich das Latex Repository geändert hat. Wenn es verändert wurde, so sollen die gesamten Latex Dateien ausgecheckt werden und alle Projekte druckkompiliert werden. Zum Erkennen der Latexdateien die Übersetzt werden sollen dient eine tcp-Datei (TeXnicCenter Projekt). In der Datei ist die Zeile „MainFile=Dokument.tex“ enthalten, diese Zeile dient als Identifizierung.

eb_svn_up-to-date überprüft ob ein lokales Repository „checkout“ Verzeichnis aktuell ist.

eb_process_files sucht nach tcp Daten und übergibt die gefundenen an das Script eb_compile_tcp.

eb_compile_tcp übersetzt eine tcp Daten mit pdflatex drei mal, ein erfolgreich erstellte PDF-Datei werden in das Verzeichnis „pdf“ kopiert.

```

/usr/local/bin/eb_svn_up-to-date [-rwxr-xr-x root root]

#!/bin/bash
#
#
# Parameter 1: svn repository path, for example /var/lib/svn/latex
# Parameter 2: local repository checkout path /usr/src/latex
#
# Script checks if a local checkout repository is uptodate with the
# subversion repository

```

```

#
# use example: eb_svn_up-to-date /var/lib/svn/latex /usr/src/latex || svn update /usr/src/latex/
#
if [ -z $1 ] ; then
    echo "svn repository missing"
    exit
fi
if ! svnlook youngest $1 > /dev/null ; then
    echo "svn repository invalid"
    exit
fi
if [ -z $2 ] ; then
    echo "local repository checkout path missing"
    Exit1
fi
REPOSITORY='svnlook youngest $1'
LOCAL_FILE='svn info $2 | grep Revision | cut -f 2 -d " "'
if [ $REPOSITORY -eq $LOCAL_FILE ] ; then
    echo up-to-date
    exit 0
else
    echo not up-to-date
    exit 1
fi

/usr/local/bin/eb_process_files [-rwxr-xr-x root root]

#/bin/bash
#
# written by evil 16.1.2010
#
# Parameter 1: folder
# Parameter 2: extention
# Parameter 3: email address for notification
#
# script search in specified folder (Parameter 1) for given extention (Parameter 2)
# for every file that matches the extention it will call a script that
# works with the file.
# The called worker script expect some parameters:
#   Parameter 1: filename (without directory)
#   Parameter 2: path to file
#   Parameter 3: email address for notification
#
if [ -z $1 ] ; then
    echo error: path not defined!
    exit
fi
if [ -z $2 ] ; then
    echo error: extention not defined!
    exit
fi
FIND_PATH=$1
TIME_FILE=/dev/shm/$2.time
cd $FIND_PATH
echo -n processing *.$2 in "$FIND_PATH"\
if [ $2 = "tcp" ] ; then
    echo as latex project
    /usr/bin/time -f %e -o $TIME_FILE -- find -name *.tcp -exec eb_compile_tcp "{}" $FIND_PATH $3 \;
else
    echo extention not supported!
fi
if [ -f $TIME_FILE ] ; then
    TIME='cat $TIME_FILE'

```

```

    echo "Processing of *.$2 required $TIME s "
    rm $TIME_FILE
fi

/usr/local/bin/eb_compile_tcp [-rwxr-xr-x root root]

#/bin/bash
#
# written by evil 16.1.2010
#
# Parameter 1: tcp file to extract (without directory)
# Parameter 2: path to tcp file
# Parameter 3: email address for notification
#
# Script extract main tex-file from tcp (Latex Project) file
# and compile it with pdflatex tree times:w
#
if [ -z "$1" ]; then
    echo tcp filename missing
    exit
fi
if [ -z "$2" ]; then
    echo path missing
    exit
fi
#echo -n change to directory
#pwd
cd "$2"
if [ ! -d pdf ]; then
    echo create directory pdf
    mkdir pdf
fi
#echo -n change to directory
#pwd
TEXFILE='cat "$1" | grep MainFile= | cut -d = -f 2'
echo -n "compiling $1 [$TEXFILE] ... "
TEXPATH='dirname "$1"'
#basename
cd "$TEXPATH"
for counter in 1 2 3 ; do
    pdflatex -halt-on-error -interaction=nonstopmode "$TEXFILE" > output.log 2>&1
    PDFFILE='cat output.log | grep "Output written on \" | cut -d \" -f 2'
    # The pdf-filename is with " or without!
    if [ -z "$PDFFILE" ] ; then
        PDFFILE='cat output.log | grep "Output written on " | cut -d " " -f 4'
    fi
    if [ -z "$PDFFILE" ] ; then
        echo error - no pdf-file created
        if [ ! -z $3 ]; then
            cat output.log | mail -s "Latex: $1/$TEXFILE" $3
        fi
        exit
    else
        echo -n "($counter/3) "
        if [ $counter -eq 3 ] ; then
            echo success
            cp "$PDFFILE" "$2/pdf"
        fi
    fi
fi
done

/etc/crontab [-rw-r--r-- root root]

```

```
0 2 * * * root eb_svn_up-to-date /var/lib/svn/latex /usr/src/latex >/dev/null 2>&1 || \  
    ( svn update /usr/src/latex/ > /dev/null 2>&1 && \  
    /usr/local/bin/eb_process_files /usr/src/latex/ tcp >/dev/null 2>&1)
```