Part2: Linux Server, Backup, Boot, LVM, Virtualization

Franz Schäfer

Linux LV, WU Wien

November 16, 2018

Table of contents

netcat - the swiss army knife

```
# apt-get install netcat
# netcat -h
# nc -h
# echo bla bla| netcat -l -p 7777
telnet localhost 7777
```

netcat - like tools

Some netcat like tools in debian:

netcat-traditional	TCP/IP swiss army knife
netcat-openbsd	TCP/IP swiss army knife
ncat	part of the nmap package
netsed	network packet-altering stream editor
socat	multipurpose relay for bidirectional data transfer
emcast	multicast toolkit

inetd - the superdaemon

config in /etc/inetd.conf

```
pop-3 stream tcp nowait root
      /usr/sbin/tcpd /usr/sbin/in.pop3d
4567  stream tcp nowait  nobody
      /usr/sbin/tcpd /bin/nc  -t 192.168.1.1 80
```

Re-read the configuration file:

```
# killall -HUP inetd
```

tcp wrapper

Built in firewall library for applications

- /etc/hosts.allow
- /etc/hosts.deny

Example:

```
ALL:127.0.0.1,192.168.0.0/255.255.255.0
man hosts_access
```

Apache Webserver

Marketshare of Apache Web Serers (Top 1M busiest Sites)

- 34% Apache
- 25% Nginx
- ...
- 10% Microsoft IIS

(Netcraft Survey September 2018)

Access a Webserver via Commandline

```
# telnet www.wu.ac.at 80
GET /
# telnet www.wu.ac.at 80
 HEAD / HTTP/1.1

HTTP/1.1 400 Bad Request
Date: Wed, 05 Dec 2007 12:38:35 GMT
Server: Apache
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

Install Apache on Debian/Ubuntu

```
# apt-get install apache
# apachectl configtest
# apachectl restart
# /etc/init.d/apache2 restart
```

Apache Configuration

```
/etc/apache2/httpd.conf
/etc/apache/httpd.conf
/etc/apache2/apache2.conf

some options can be configured in .htaccess

Most options in otherfiles that are included like this

Include /etc/apache2/mods-enabled/*.load
Include /etc/apache2/mods-enabled/*.conf
```

Scopes of the Apache Config

- Sitewide

```
Listen 80
DocumentRoot /var/meinwww/
```

- Virtual Host
- Directory
- Location

A simple HTML Document

create a file named index.html

```
<html>
<h2>test &uuml;berschrift</h2>
test <b>fett</b>
</html>
```

Per default: index.html will be served if you go to a URL that points to a directory.

Example: VirtualHost

Can be based on Name or IP

```
<VirtualHost 123.23.17.9:80>
ServerAdmin webmaster@meinserver.at
DocumentRoot /var/www/meinserver/
ServerName www.meinserver.at
ServerAlias meinserver.at
php_flag register_globals 0
RedirectPermanent /w/ http://wuw.at/bla/
</VirtualHost>
```

Example: Configuration for a Directory

```
<Directory /var/www/scripts/>

 AllowOverride AuthConfig
 Options +ExecCGI -Indexes
 Addhandler cgi-script .cgi
</Directory>
ScriptAlias /cgi/ /var/www/scripts/
```

Securing a Webserver via HTTPS

```
<VirtualHost 12.34.56.78:443>
SSLEngine on
SSLCertificateFile /etc/cert/mein.crt
SSLCertificateKeyFile /etc/cert/mein.key
ServerName meinserver.at
DocumentRoot /var/www-secure
</VirtualHost>
```

VirtualHosts with SSL should have different IPs But will now also work via SNI.
You need to generate your keys with e.g. openssl

Generate your keys with openssl

```
# openssl req -new -nodes \
   -newkey rsa:1024 -keyout mein.key \
   -out mein.csr

# openssl x509 -req -in mein.csr \
   -signkey mein.key -out mein.crt \
   -days 365
```

Check it via:

Get your keys certified

- letsencrypt.org
- Verisign, Thawte , & Co... $$
- aconet - free for .ac.at
- Peer2Peer: cacert.org

A simple CGI Script

```
#!/bin/bash
echo Content-type: text/plain
echo
echo my process id
id
echo date and time
date
```

A simple PHP Script

```
<HTML>
<?
  for($i=1; $i<20 ; $i++) {
    echo $i," squared is ",$i*$i,"<br>";
  }
?>
</HTML>
```

Security for Web Scripts

- Update Often
- Update Regularily
- Off the shell scripts and packages - keep track of new versions

SQL Injection anc Cross Site Scripting

```
$res=mysql_query(
 'SELECT * FROM bla WHERE id="' . $_GET['id'] . '"'
);

echo "Your id is ",$_GET['id'];
```

Top PHP Secruity Mistakes

- Use Unfiltered Input (and Include File, Build SQL Query, etc
- Unfilter Output XSS

Mailserver Basics

- Store and Forward via port 25 (SMTP)
- Per default mail end up in an mbox file in /var/spool/mail/
- Per default mail end up in an mbox file in /var/spool/mail/
- Later: Download mails via pop3
- Then: Manage mailbox on server via IMAP

Overview Mailserver

MTA

sendmail	Old but good
exim	small, simple, GPL
postfix	the contender
qmail	exotic

IMAP

cyrus	stable, powerful
courier	simpler for small sites
uw-imapd	standard mbox format

Email via Telnet

```
host -t mx wu.ac.at
telnet mx1.wu.ac.at 25
220 mx1.wu.ac.at ESMT..
helo .
mail from: fs@mond.at
rcpt to: mond@wu-wien.ac.at
data
bla
.
quit
```

Commandline Email - mutt

```
# echo test | \
   mutt -s test xx@mond.at

# mutt -f  \
  imaps://h7788999@sslmail.wu-wien.ac.at
```

mysql commands

```
# mysqladmin -uroot -p create bladb
# mysqldump -uroot -p bladb >other.dump
# cat other.dump | mysql -uroot -p blidb
# echo "select * from blatable;" \
  | mysql -uroot -p  bladb

# mysql -uroot -p  bladb

  CREATE USER 'anna'@'localhost';
  SET PASSWORD FOR 'anna'@'localhost' = PASSWORD('geheim');
  GRANT SELECT ON bladb.* TO 'anna'@'localhost' ;
```

Samba - Fileserver for Windows

in the file /etc/samba/smb.conf

```
[musik]
comment = meine mp3sammlung als share
writable = no
locking = no
path = /extraplatte/mp3/
public = yes
hosts allow = 192.186.0.0/255.255.0.0
```

NFS - the Unix Network Filesystem

in the file /etc/exports

```
/home/      gss/krb5i(rw,sync,fsid=0,no_subtree_check)
/data/      10.11.12.13(rw,no_subtree_check)
```

kvm virtualization

```
# wget \
http://distro.ibiblio.org/tinycorelinux/5.x/x86/release/CorePlus-current.is

# qemu-img create -f qcow2 tinycore.qcow2 2G
# kvm -hda tinycore.qcow2 -cdrom CorePlus-current.iso -boot d -m 200
```

Other Server Applications

asterisk IP telephony (sip, h323, isdn, ...)

nagios, icinga monitoring

small services dhcp, dnc, ntp, tftp ...

X11 terminal server

kerberos, ldap, radius authentication and directory

kvm, xen, qemu, LXC virtualization

irc, jabber chat

... ...

Backup: The Problem

- Metadata (Permissions, Timestamps, Symlinks, Hardlinks, Device Files)
- Multiple versions of Files
- Sparse Files
- Easy restore in case of an emergency
- In some cases: encryption

tar - the classic

```
# tar cfpz dip.tgz  diplomarbeit/
# tar tfvz dip.tgz
# cd / ; mount /bla
# tar one-filesystem -c -p -f - . \
| (cd /bla/ && tar xfvp - )
# tar ... | ssh nachbarserver \
"cd /bla/ && tar xfvp - "
# nc -l -p 7777 | tar xfvlp -
# tar ... | nc otherhost 7777
```

tar - incremental backup

- option `--newer` oder `-N` or
- option `--files-from` oder or `-T` and create the list of files with an other program. e.g `find`

```
# find . -ctime -2 >backup.list
# tar -c -T backup.list -z -f backup.tgt
```

encrypting with gpg

```
# tar ... | gpg -c > backup.tgz.gpg|
```

rsync

```
# mount /backup || exit 1
# cd /
# rsync -Hxa  --delete . /backup
# umount /backup
```

rsync: also works remote over ssh or dedicated rsync server

duplicity - encrypted backup

```
# apt-get install duplicity
# duplicity /home/anna/ \ file:///ext/duplicity/
# duplicity /home/anna \
 scp://karl@woanders.at:/bla/
```

Do not forget your passphrase

backends: local, ssh/scp, rsync, ftp

```
# apt-get install duplicity
```

Enterprise Backup

Enterprise features

- central backup for many hosts
- database of backuped files
- managing tape library

Enterprise Backup

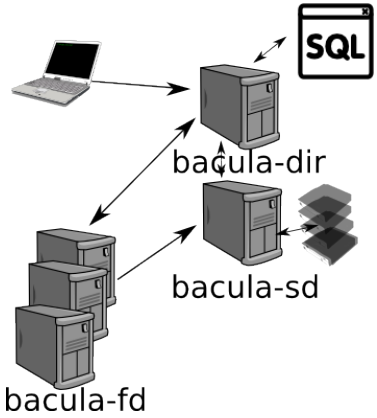
Enterprise features

- central backup for many hosts
- database of backuped files
- managing tape library

Enterprise Backup Systems

- bacula ..... free
- tivoli .... commercial (IBM)

Bacula Overview



bacula-dir **director** controls all other daemons

bacula-fd **file daemon** – daemon that runs on each client

bacula-sd **storage daemon** – actual backup (tape handling, etc, ...)

cron

backup periodically and automatically. e.g.: via cron.

- cron start processes at certain configurable times. E.g.: each sunday night at 3:17
- system wide configuration is at:
  `/etc/crontab`
- each user can have his/her own crontab:
  `crontab -e`

example crontab

```
53 3 * * * /root/meinbackup.sh >> /var/log/backup.log 2>&1
13 07 * * 0   /root/sonntagmorgens.sh
01,21,41 * * * * /root/3malprostunde
17 */3 * * 1-4   /root/8mal_mo-do.sh
# 0:17 3:17 6:17 .. each monday till thursday
30 7 1-24 12 * /root/advent.sh
```

important `PATH=...` in the system wide crontab there is an additional column that specifies the user.

Backup - Summing up

- keep a backup off-site
- keep older versions
- test a restore once in a while

Backup Hardware

Cheap Solution - 2 External USB Disks

- e.g. change weekly
- keep one off-site
- costs €90/3TB
- keep a backup off-site
- keep older versions
- test a restore once in a while

Enterprise Backup Hardware

Enterprise - Tape Library

- tape drive starting €1200
- tape media (LTO8) €120/12TB
- tape library starting at €5000

Possible Setup

- rsync to different server in different bilding
- keep old version via btrf snapshots
- additional regular full backup to tapes

Cloning your System with tar

```
# cd /
# tar --one-filesystem -c -p -f - .\
 | netcat otherhost 7777
```

- easy with above tar or with `dd if=/dev/sda`
- do not forget to change hostname
- ip address
- ssh-key
- persistent devices `/etc/udev/rules.d/` (MAC address)

Software RAID

RAID is **not** a Backup

Software RAID

RAID is **not** a Backup

Advantages of Software Raid

compared to hardware raid

- no vendor dependent tools for setup, repair and monitoring
- on a partition by partition basis
- over disks on different controller cards
- not much overhead for RAID1
- with todays CPU speed: even higher RAID levels in software possible

mdadm

```
# mdadm create /dev/md0 \
  -l1 -n2 /dev/sdb7 /dev/sdc7
# mkfs.ext3 /dev/md0
```

After a disk failure:

```
# mdadm --manage /dev/md0 \
 --add /dev/sde5
```

/proc/mdstat

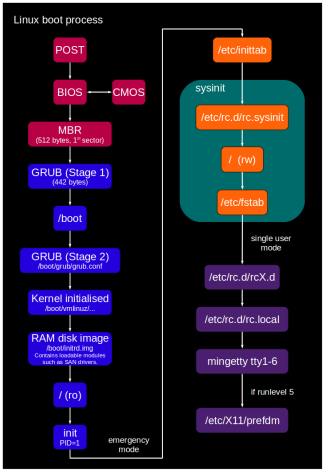
```
# cat /proc/mdstat
Personalities : [raid1]
 md2 : active raid1 sdc3[1] sdb3[0]
           146480576 blocks [2/2] [UU]
 md3 : active raid1 sdc4[1] sdb4[0]
           159252224 blocks [2/2] [UU]
```

config file (important for **monitoring**!)

```
/etc/mdadm/mdadm.conf
```

Booting Linux

- kernel auf floppy
- lilo
- grub
- loadlin
- syslinux, isolinux, pxelinux
- tftp, nfsroot,
- ...



From: `http://ilostmynotes.blogspot.co.at/2011/01/linux-boot-process-digram.html` (unkown copyright)

The Linux Boot Process

- BIOS starts GRUB from MBR
- GRUB loads kernel and inittrd
- starts kernel with initial RAM-disk
- initrd loads kernel modules and mounts /
- with pivot_root changes to / and init loaded

SysV init vs systemd

- systemd: dependency handling - faster boot times
- systemd: better monitoring of running services
- sysvinit: runlevel vs. systemd targets
- systemsd: cgroups, keeps track of services, builtin logic
- sysvinit simpler

systemd cheatsheet

```
# systemctl list-units
# systemctl restart someservice
# systemclt status someserive
# journalctl
```

config files in /usr/lib/systemd/ /etc/systemd /lib/systemd
start stops scripts in /etc/init.d/ still provided by distributions

How to set a new root password

On the GRUB prompt `ctrl-E` and then use `init=/bin/bash` kernel parameter.

```
# mount -o remount -rw -n /
# passwd
# mount -o remount -r -n /
# sync
# reboot
```

Use Live CD to reset root password

Use rescue or live CD (knoppix, grml, ...) or similar

```
# sudo bash ; su -
# mkdir /bla
# mount -t ext3 /dev/sda7 /bla
# chroot /bla /bin/bash
# passwd
# exit ; umount /bla
```

Also use it for

- repair boot sector
- emergency backup, restore
- fix hanging boot scripts, ...

GRUB Setup

- initial install: `grub-install /dev/sda`
- directory `/boot/grub`
- debian settings `/etc/default/grub`
- `update-grub` to update configuration

Sample from the generated `/boot/grub/grub.cfg`

```
menuentry 'Debian GNU/Linux, with Linux 2.6.26-1-686' {
        set root='(mduuid/9c28e79e20828a716cd5a85366beffb2)'
        linux   /vmlinuz-2.6.26-1-686
        root=UUID=d69d1d3a-4f49-447d-8cd4-b3b56884458c ro sdf=0 vga=791
        initrd  /initrd.img-2.6.26-1-686
}
```

example serial console

in /etc/default/grub

```
GRUB_CMDLINE_LINUX="console=tty0 console=ttyS1,115200n8"
GRUB_TERMINAL=serial   # for both use: =’serial console’
GRUB_SERIAL_COMMAND="serial --speed=115200 --unit=1 --word=8
  --parity=no --stop=1"
```

in /etc/inittab

```
T1:23:respawn:/sbin/getty -L ttyS1 115200 vt100
```

Logical Volume Management - LVM

- Flexibilty with Disk Layout
- Snapshots
- based on device-mapper

Terminology:

- PV (physcial volume)
- LV (logical volume)
- VG (volume group)

LVM Example

```
# pvcreate /dev/sda2
# pvcreate /dev/md1
# vgcreate meinvg /dev/hda2\ /dev/sda2 /dev/md1
# lvcreate -L20G -nmp3lv meinvg
# mkfs.ext3 /dev/meinvg/mp3lv
# pvscan
# lvdisplay
```

LVM advanced

Snapshots:

```
# lvcreate -L22G -s -n dbbackup \
  /dev/meinvg/datenbank
```

Remove Disk:

```
# pvmove /dev/sda2
# vgreduce meingv /dev/sda2
```

Resize:

```
# lvresize -L +30G /dev/meinvg/mp3lv
# resize2fs ...
```

newer LVM features

- lvmthin - Thin Volumes with cheap snapshots
- lvmcache - Use a fast SSD as cache for slower rotating disks
- raid directly done by LVM/device mapper instead of md driver

Definition of Virtualization

virtualization

In computing, **virtualization** is the creation of a virtual (rather than actual) version of something, such as a hardware platform, operating system (OS), storage device, or network resources.

While a **physical computer** in the classical sense is clearly a complete and actual machine, both subjectively (from the user's point of view) and objectively (from the hardware system administrator's point of view), a **virtual machine** is subjectively a complete machine (or very close), but objectively merely a set of files and running programs on an actual, physical machine (which the user need not necessarily be aware of).

Types of Virtualization

- CPU **Emulation** - e.g.: VICE (C64 Emulator), QEMU
- Hardware **Virtualization** - e.g.: on native CPU (e.g. KVM, VMware, UML)
- **Containers** (Shared Kernel) - e.g: chroot, BSD Jails, OpenVZ, Linux-VServer (Operating system-level virtualization), LXC (Docker ...)

Cool Feature: using QEMU to work with chroots of different CPU types.

Type1 vs. Type2 Hypervisor

Type 1 (or native)

runs directly on hardware - e.g. Xen, KVM, VMware ESX/ESXi, Hyper-V

Type2 (hosted)

runs under OS:- e.g. VMware Workstation and VirtualBox, all CPU emulators

Emulation

Microsoft Virtual PC

x86 to x86, powerpc to x86
only runs on windows and osx, limited number of guest OS

QEMU

Host CPU: x86, x86-64, IA-64, PowerPC, SPARC 32/64, ARM, S/390, MIPS
Target CPU: x86, x86-64, ARM, CRIS, LM32, MicroBlaze, MIPS, SPARC 32/64, PowerPC, M68k, Alpha, S/390, Unicore32, SH4, xtensa

KVM, Xen, VMware compared

Xen

Dedicated small Hypervisor, Most Hardware Emulation with support from Dom0

KVM

Fullblown Linux Kernel as Hypervisor, needs assitance from the CPU (only newer CPUs), uses the QEMU Framwork for handling of virtual machines and hardware

VMware-Server

ESX as Hypervisor, Dom0 Linux just for maintenance, Most Hardware drivers integrated in ESX.

Paravirtualization

The guest operating system is aware of the fact that it is running virtualized and helps the host system in its tasks.
e.g.: adding device drivers that are optimized for virtualization.

Why are we using virtualization anyways?

- Efficient hardware utilization: many small servers do not need 100% of CPU all the time
- Administrative isolation
- Secruity barriers
- Redundancy, protection against hardware failure

more topics

- KVM/QEMU hands on
- libvirt handson
- pacemaker
- simulation of networking
- shared storage
- disk formats