



Linux administration

Mikael M. Hansen

mhansen@cs.aau.dk

E2-121



TOC

- Hvem er jeg
- Mine indlæg
- Dagens program: Linux administration
 - "Teorien"
 - Værktøjer
 - Servere



Hvem er jeg

- Mikael M. Hansen
- System– og netværksadministrator på CS
- Uddannet datalog på CS
 - Afbryd mig hvis I har spørgsmål eller lign.



Mine indlæg

- 5/3: Linux administration
- 19/3: Unix/Windows integration – Samba
- 22/3: Systemadministrations
arbejdsprocesser og procedure
» Spørge time (bog og praksis)
- 26/3: Systems and Network Management



Linux administration

Formål:

At give basalt kendskab til linux systemets opbygning og konfiguration af dets basale dele. Kendskab til basale kommandoer samt konfiguration af enkelte servere



Linux administration

Hvilken linux?



Linux administration

Del 1 - "Teorien"



Boot processen – overblik (x86)

1. BIOS starter boot loader (stage 1) fra MBR
2. Boot loader (stage 1) loades ind i hukommelsen og starter stage 2 (fra /boot)
3. Boot loader (stage 2) loader kernen i hukommelsen. Kernen loader moduler og mounter / partitionen read-only
4. Kernen overlader boot processen til /sbin/init
5. /sbin/init mounter partitioner og loader services samt user-space tools.
6. Brugeren presenteres for en login prompt (evt. grafisk)



Boot processen i detaljer

- Boot loader stage 1 og 2 – er oftest Grub (Tidligere Lilo)
- Grub (stage 1) har kun til formål at finde og load stage 2 som indeholder alt logikken
- Stage 2 finder sin konfigurationsfil (/boot/grub/grub.conf) og loader den. Brugeren præsenteres for den konfigurerede menu af kerner/optioner.
- Her har brugeren mulighed for at intervenere.
 - Hvorfor skulle man det?



Boot processen i detaljer

- Efter evt. editering af optioner/kerne parametre loades kernen (/boot/vmlinuz-xxx) og initramfs image (/boot/initrd-xxx) ind i hukommelsen
- Kernen startes med de evt. angivne parametre. Den udpakker initramfs filen i (/sysroot på RAMFS og ikke /boot som der står)
 - Hvorfor bruges initrd filen?
- Kernen rydder op og /sbin/init startes. Init kører:
 - /etc/rc.d/rc.sysinit
 - /etc/inittab
 - /etc/rc.d/init.d/functions
 - /etc/rc.d/rc<runlevel>.d/*
 - /sbin/mingetty for hvert virtual console
 - /etc/X11/prefdm hvis runlevel er 5



Runlevels

- Ideen: Systemer bruges forskelligt og skal derfor også startes forskelligt.
- RedHat bruger SysV modellen andre bruger BSD modellen eller afarter af SysV.
 - <http://en.wikipedia.org/wiki/Sysvinit>
- De vigtigste er:
 - 0: Halt system
 - 1: Single-user mode
 - 3: Multiuser (text)
 - 5: Multiuser (graphics)
 - 6: Reboot



Filsystemets struktur

- RedHat bruger File System Hierarchy Structure
 - Fokuserer på filsystem som
 - Sharable vs. Unsharable
 - Static vs. variable
- Hvorfor standardisere
 - Ens struktur imellem distributioner
 - Mulighed for at dele filsystemet eller dele deraf
 - Sikkerhed



Filsystemets struktur

- Eksempler
 - /etc - Konfigurations filer.
 - /var - Datafiler
 - /media - Mountpoint for flytbare medier (usb,dvd)
 - /opt - Til store software pakker
 - /proc - Filer som interfacer til/fra kernen
 - /dev - Representerer enheder på systemet
 - /usr - Katalog/partition som kan deles. Er underindelt i kataloger til forskellige ting (f.eks. etc, bin, sbin)



Filsystemets struktur

- Nogle udvalgte:
 - /etc/sysconfig
 - Stedet for konfigurationsfiler under RedHat
 - En fil per service, OS-del. F.eks
 - /etc/sysconfig/xinetd til konfiguration af xinetd services. Filen læses af /etc/init.d/xinetd når den startes.
 - /proc
 - Stedet til at interagere med kernen. F.eks.
 - cat /proc/cpuinfo - lister CPU information
 - echo "scsi add-singe-device 1 2 3 4" > /proc/scsi/scsi – Tvinger scsi mid-layer til at opbygge device tables for SCSI kort 1, kanal 2, id 3, LUN 4



Filsystemets struktur

– /proc/x/

- Katalog under /proc som modsvarer en process. Dvs. x er pid'en af processen
- Indholdet af /proc/x/ er relevant information omkring processen. F.eks:
 - cmdline - Kommandolinien som processen er startet med
 - environ - The environment variables of the process
 - root - link to root directory of the process



PAM

- PAM – Pluggable Authentication Modules
- Fordele:
 - Fælles autentificerings framework
 - Øget fleksibilitet og kontrol
 - Udviklere behøver ikke at lave deres egen autentificering



PAM

- Hver applikation/service opretter en fil i `/etc/pam.d/` navngivet efter servicen
- Filen indeholder definitionen af hvordan servicen vil have brugeren autentificeret.
- Det gør den vha. op til 4 direktiver
 - Modul interface
 - Kontrol flag
 - Modul
 - Modul parametre
- Bemærk at disse kan "stackes"



PAM

- Mulige værdier for disse direktiver er:

- Modul interface

- auth - autentificerer
 - account - kontrollerer gyldighed af en konto
 - password - skifter og verificerer password
 - session - Konfigurer brugersessioner m.m.

- Kontrol flag

- required - Skal lykkedes. Ved fejl gives besked efter sidste moduler har brugte det pågældende interface
 - requisite - Skal lykkedes. Ved fejl gives besked straks
 - sufficient - Ignoreres ved fejl. Men ved succes og ingen required ovenfor er fejlet er brugeren autentificeret
 - optional - Resultatet ignoreres. Har kun betydning hvis modulet er den eneste som referer interfacet



PAM

- Mulige værdier (fortsat)
 - Modul
 - Navnet på modulet. F.eks:
 - pam_unix: Checker op imod unix password databasen
 - pam_limits: Sætter limits for brugeren (antal processer, prioritet, stack size m.m.)
 - Modul parametre
 - Parametre til modulet. Mulige parametre afhænger af modulet (Se man siden for modulet).



PAM

- RedHat benytter oftest /etc/pam.d/system-auth konfigurationsfilen til de forskellige services.
- Et eksempel:

```
auth required pam_securetty.so
auth required pam_unix.so shadow nullok
auth required pam_nologin.so
account required pam_unix.so
password required pam_cracklib.so retry=3
password required pam_unix.so shadow nullok use_authtok
session required pam_unix.so
```



Kernen

- Modulært opbygget
 - Består af selve kernen og et antal moduler.
 - Man kan kompilere en kerne helt statisk (ingen moduler)
 - Mere fleksibelt at loadere moduler efter behov.
- `/etc/modprobe.conf` bruges til at influere hvilke moduler der loades som default.
 - Manuel kan følgende værktøjer bruges:
 - `modprobe`: loader et modul ind i kernen
 - `lsmod`: lister hvilke moduler er loadet
 - `rmmod`: fjerner et modul
 - `modinfo`: lister information om et modul



Kernen

- Forsigtighed hvis man kompilerer selv
 - Der er ikke altid muligt at bruge en standard kerne.
- Oftest kan man nøjes med installere moduler til den kørende kerne.



Det grafiske system

- Er modulært opbygget. Komponenterne er:
 - Xserver
 - Fra Xorg eller XFree86
 - Display Manager
 - xdm, gdm, kdm, ..
 - Windows Manager
 - Gnome
 - KDE
 - WindowMaker
 - Twm
 - M.fl.



Det grafiske system

- RedHat bruger X fra Xorg
- Det er en enkelt binær (/usr/X11R6/bin/Xorg)
 - Den loader de nødvendige moduler fra /usr/X11R6/lib/modules/
 - Konfigurationsfil (/etc/X11/xorg.conf)
- DisplayManager startes (f.eks. gdm)
 - Den loader sin konfigurationsfil (/etc/X11/gdm/gdm.conf)
 - Viser en grafisk logon skærm.
 - Herfra kan man vælge sin session (WindowManager)



Linux administration

Del 2 - Værktøjerne



Brugere/Grupper

- De basale kommandoer til det er:
 - useradd, groupadd
 - Eksempel:
 - /usr/sbin/groupadd -g gid group
 - /usr/sbin/useradd -g group -u uid user
 - Har også andre option
 - De enkelte distributioner har oftest deres egne (grafiske) værktøjer til det.



RPM

- RPM Package Manager
 - Opfundet af RedHat
 - Hovedprogrammet er `/bin/rpm`
- Designet med følgende for øje:
 - Upgradability
 - Powerful querying
 - System verification
 - Pristine Sources



RPM

- 5 hovedfunktioner:
 - Installere (`rpm -i`)
 - Afinstallere (`rpm -e`)
 - Opgradere (`rpm -u`)
 - Spørge (`rpm -q`)
 - Verificere (`rpm -V`)
- Pakker er signed (GnuPG)
 - Verificer den med `rpm -K`
- RedHat har et grafisk værktøj til at installere RPM'er:
 - `system-config-packages`



Netværks konfiguration

- Formål: at få et "fungerende" system på nettet
- Bemærk:
 - Man bør fra starten beslutte om man bruger det grafiske værktøj: system-config-network eller man gør det manuelt.
 - Det grafiske værktøj er ikke smart nok til at parse konfigurationsfilerne ved opstart. **DUMT!**
 - Det grafiske værktøj kan ikke konfigurere VLAN tagging... tsk tsk!



Netværks konfiguration

- Manuelt kræver lidt viden om de filer som er involveret:
 - /etc/hosts
 - Lokal navne resolution
 - /etc/resolv.conf
 - DNS konfiguration
 - /etc/sysconfig/network
 - Basalt netværksaktivering, hostname, gateway, evt. NIS domain
 - /etc/sysconfig/network-scripts/ifcfg-interfaceX
 - Interface: eth, ppp, bond, ipsec, tr m.m.
 - X er interface nummeret. 1. ethernet NIC = eth0



Netværks konfiguration

- Eksempel: `/etc/hosts` (fra `homer.cs.aau.dk`)

```
127.0.0.1 localhost.localdomain localhost
130.225.194.13 homer.cs.aau.dk homer
130.225.194.14 nis.cs.aau.dk nis
130.225.194.33 nis1.cs.aau.dk nis1
130.225.104.55 aton.cs.aau.dk aton loghost
```



Netværks konfiguration

- Eksempel: `/etc/resolv.conf` (fra `homer.cs.aau.dk`)

```
search cs.aau.dk  
nameserver 130.225.194.2  
nameserver 130.225.195.2
```



Netværks konfiguration

- Eksempel: `/etc/sysconfig/network` (fra homer.cs.aau.dk)

```
NETWORKING=yes  
HOSTNAME=homer.cs.aau.dk  
GATEWAY=130.225.194.1  
NISDOMAIN=cs.aau.dk
```



Netværks konfiguration

- Eksempel: /etc/sysconfig/network-scripts/ifcfg-eth0
 - (fra homer.cs.aau.dk)

```
DEVICE=eth0
BOOTPROTO=static
HWADDR=00:14:4f:3f:fd:28
IPADDR=130.225.194.13
NETMASK=255.255.255.0
ONBOOT=yes
TYPE=Ethernet
```



Netværks konfiguration

- `/etc/init.d/network`
 - Start/stop netværk på maskinen
- `/sbin/{ifup|ifdown}` interface
 - Start/stop et enkelt interface
- `/sbin/ifconfig`
 - Lister summarisk information omkring interfaces

Netværks konfiguration

- Ifconfig et eksempel:

```
eth0  Link encap:Ethernet HWaddr 00:14:4F:28:2A:DC
      inet addr:130.225.194.32 Bcast:130.225.194.255 Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:5938118 errors:0 dropped:0 overruns:0 frame:0
      TX packets:5519060 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:2977139488 (2.7 GiB) TX bytes:3888696065 (3.6 GiB)
      Base address:0x9c00 Memory:fbfe0000-fc000000
```



Små nyttige værktøjer

- Kort gennemgang af nogle nyttige værktøjer som man vil få brug for på et tidspunkt.
- Ikke nogen komplet liste, men en liste af nogen som man bør kende.



Små nyttige værktøjer

- vi - /usr/bin/vim
 - Editor
 - Simpel men genial
 - Tager lidt tid at komme ind i
 - Den er godt givet ud
 - Bemærk at vi på RedHat egentligt er vim (vi-improved)
 - Bemærk også placeringen!
 - Tsk tsk!



Små nyttige værktøjer

- `/bin/ping`
 - Tester netværksforbindelse
 - Ved at sende et ICMP ECHO_REQUEST datagram
 - Rapporterer svar tider.



Små nyttige værktøjer

- `/bin/netstat`
 - Nyttig netværksinformation. F.eks.
 - Aktive forbindelser (`netstat`)
 - Protokol statistik (`netstat -s`)
 - Interface statistik (`netstat -i`)
 - Routing tabellen (`netstat -r`)

Små nyttige værktøjer

- Eksempel: netstat -i

```
# netstat -i
Kernel Interface table
```

Iface	MTU	Met	RX-OK	RX-ERR	RX-DRP	RX-OVR	TX-OK	TX-ERR	TX-DRP	TX-OVR	Flg
eth0	1500	0	474743028	0	0	0	328449131	0	0	0	BMRU
lo	16436	0	17177105	0	0	0	17177105	0	0	0	LRU

- Eksempel: netstat -r

```
# netstat -r
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	MSS	Window	irtt	Iface
130.225.194.0	*	255.255.255.0	U	0 0	0		eth0
169.254.0.0	*	255.255.0.0	U	0 0	0		eth0
default	aau-gw.cs.aau.d	0.0.0.0	UG	0 0	0		eth0

Små nyttige værktøjer

- tcpdump
 - dumper headers fra netværkspakker
 - wireshark(etherreal) er bedre hvis man kører X
- Eksempel: ping fra aton til eg

```
-sh-3.00# tcpdump dst host 130.225.194.199
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
16:55:51.358118 arp who-has eg.cs.aau.dk tell aton.cs.aau.dk
16:55:51.358319 IP aton.cs.aau.dk > eg.cs.aau.dk: icmp 64: echo request seq 0
16:55:52.357957 IP aton.cs.aau.dk > eg.cs.aau.dk: icmp 64: echo request seq 1
16:55:53.357753 IP aton.cs.aau.dk > eg.cs.aau.dk: icmp 64: echo request seq 2
16:55:54.357571 IP aton.cs.aau.dk > eg.cs.aau.dk: icmp 64: echo request seq 3
16:55:55.357366 IP aton.cs.aau.dk > eg.cs.aau.dk: icmp 64: echo request seq 4
```



Små nyttige værktøjer

- Mange flere
 - traceroute, free, iostat, nfsstat, top, ps, df, du, lspci, dmesg
- Med hver ny teknologi kommer oftest flere nye kommandoer man skal lære at kende.



Linux administration

Del 3 - Servere



Syslog

- Systemet og dets komponenter sender log beskeder via syslog
 - Konfigurationsfil specificerer hvad der skal logges hvor.
 - Syslog sørger for indhold i
 - F.eks. /var/log/messages
 - Pakker ryger på port 514/udp
 - Beskeder er kategoriseret efter facility og priority
 - Facility: auth,authpriv,cron, daemon, kern, lpr, mail, syslog, user, uucp, local0..local7
 - Priority: debug, info, notice, warning, err, crit, alert, emerg

Syslog

- Starte syslog server (som også logger for remote hosts):
 - Ret /etc/sysconfig/syslog:
 - `SYSLOGD_OPTIONS="-r "`
 - Ret /etc/syslog.conf på serveren:
 - Hvad skal gemme og hvor?

auth.info	/q/log/auth.log
mail.*	/q/log/mail.log
kern.notice	/q/log/kern.log
*.crit	/q/log/crit.log



Syslog

- Ret `/etc/syslog.conf` på klienten:

```
auth.info      @server
daemon.notice  @server
kern.notice    @server
*.crit         @server
mail.debug     @server
```

- `/etc/init.d/syslog` start (restart)
 - Både server og klient
- En bedre syslog-server: `syslog-ng`
 - Bedre filtrering.
 - Mulighed for tcp.



xinetd

- En "super-service"
 - Den giver adgang til (starter) andre services efter behov.
 - Bruges til alle de services som ikke startes før de skal bruges.
 - Funktionaliteten er:
 - Modtager en forbindelse
 - Kontrollerer TCPwrappers adgangs kontrol
 - Kender I TCPwrapper funktionaliteten?
 - Kontrollerer adgangs kontrol for den ønskede service
 - Starter servicen og given den kontrol over forbindelsen

xinetd

- Konfigurations fil: /etc/xinetd.conf
- Startes med /etc/init.d/xinetd start
- /etc/xinetd.d/ indeholder en fil per service den kan starte.
 - Et eksempel:

```
service nrpe
{
    flags          = REUSE
    socket_type    = stream
    wait          = no
    user           = cadm
    server         = /usr/priv/nrpe/nrpe
    server_args    = -c /usr/priv/nrpe/nrpe.cfg -i
    log_on_failure += USERID
    disable        = no
    only_from      = 130.225.194.19 127.0.0.1
}
```



NFS

- Network File System
 - Tillader servere at exportere et filsystem som klienter kan mounte og bruge som var det lokalt.
 - NFS findes i version 2, 3 og 4
 - Karakteristika varierer
 - Version 4 er kun TCP
 - Version 2 og 3 kan benytte både TCP og UDP
 - Version 4 understøtter Kerberos
 - Alle er RPC baseret, men de påkrævede services er forskellige

NFS

- Konfigurering af server:
 - Ret /etc/exports
 - Start server processerne
 - /etc/init.d/nfs start
 - Evt. re-eksporter /usr/sbin/exportfs -r

```
/          client1(ro,sync) server*.localdomain(rw,no_root_squash)
/q/disk_0  client2.cs.aau.dk(rw,sync,no_root_squash)
/q/disk_1  @myclients(rw,sync)
```

NFS

- På klienten:
 - `mount server:/q/disk_0 /mnt/nfs`
- Bruge evt. `showmount -e server` til at se hvad serveren eksporterer:
 - Bemærk: Ingen options er vist

```
# showmount -e lfs2
Export list for lfs2:
/          @adm
/q/pack    @adm,@nadm
/q/disk_0  @adm,@nadm
/q/pack1/adm @adm
/q/pack1/old @adm,@nadm
/q/pack1/pack @adm,@nadm
```



NFS

- Faldgrube (version 2 og 3):
 - Adgang gives til klienten, ikke brugeren
 - Begræns adgang til NFS serveren
 - Kun kendte klienter under din kontrol!
 - En seriøs fare
 - Ingen fast port der kan åbnes for som god design ellers kræver
 - Port 2049 + vilkårlige porte



SSH

- Secure Shell
 - Giver ”sikker” adgang til en terminal
 - Krypteret med pub/priv-key kryptering
 - Port Forwarding
- RedHat benytter OpenSSH
 - I /etc/ssh/ er konfigurationsfilerne
 - sshd_config for serveren
 - ssh_config for klient programmerne
 - Ssh, scp, sftp

SSH

- `sshd_config`: et CS eksempel

```
Port 22
Protocol 2
HostKey /usr/priv/openssh/etc/ssh_host_dsa_key
LoginGraceTime 300
PermitRootLogin yes
StrictModes yes
X11Forwarding yes
X11DisplayOffset 10
PrintLastLog yes
TCPKeepAlive yes
PermitUserEnvironment yes
SyslogFacility AUTH
LogLevel DEBUG
HostbasedAuthentication yes
PubkeyAuthentication yes
PasswordAuthentication yes
PermitEmptyPasswords no
UseDNS yes
Subsystem sftp /usr/priv/openssh/libexec/sftp-server
```



SSH

- `ssh_config`: et CS eksempel

```
ForwardX11 yes
ForwardX11Trusted yes
HostbasedAuthentication yes
PreferredAuthentications hostbased,publickey,password,keyboard-interactive
GlobalKnownHostsFile /usr/priv/openssh/etc/ssh_known_hosts
EnableSSHKeysign yes
```



SSH

- Binære er:
 - ssh: klient programmet som skaber en forbindelse til en server
 - scp "secure copy" skaber også en forbindelse til en server, men for at kopiere filer:
 - `scp localfile user@server:/path`
 - sftp "secure ftp" skaber også en forbindelse for at kopiere filer.
 - ftp-"like"



SSH

- `ssh-keygen`
 - Genererer keys. F.eks:
 - `ssh-keygen -t dsa -f filename`
 - Husk, ingen password ved host keys
- `ssh-keyscan hosts`
 - Samler pub keys fra hosts
- `ssh-agent`
 - Startes af brugeren og holder alle private keys til brug ved public key authentication.
 - Keys skal added med `ssh-add`



SSH

- På en RedHat maskine startes OpenSSH med
 - `/etc/init.d/sshd start`
 - Den genererer selv keys hvis de ikke allerede er der.
- SSH bør styres centralt
 - Specialt mhp. key-management.



Spørgsmål?