CALLEBAUT Gilles

SYSTEEM- EN NETWERK-BEHEER

14 januari 2016

Inhoudsopgave

Lijst van figuren

Lijst van tabellen

1 Basis Unix/Linux (Labo 2)

1.1 TTY-omgeving

Via het commando `<CTRL>-<ALT>-F1` verlaten we de grafische user interface, en komen terecht in de TTY-omgeving[1]. Via het commando `<CTRL>-<ALT>-F8` komen we terug terecht in de TTY-omgeving.

1.2 X-windows

Via het platform X Windows Systeem kunnen we in o.a. UNIX een grafische bediening gebruiken. De architectuur van X-Windows bestaat uit een (X window) server, deze beheert het scherm. De clients maken verbinding met de X-server en communiceren over protocols. De clients geven aan wat er op het scherm moet komen via de X-server. De X-server staat meestal lokaal.

Windows X Systemen zijn meer flexibel en aanpasbaar aan de noden van de gebruiker. In tegenstelling tot de X-windows systemen is de desktop environment ingebouwd bij Windows. We kunnen dus als client over een netwerk verbinden met een X-server, deze kan dan lokaal een scherm bedienen.

Microsoft Windows liet vroeger programma's rechtstreeks tekenen op het scherm, vanaf Windows Vista werd dit geregeld door "The Desktop Window Manager". Het programma schrijft naar een offscreen memory buffer (a.k.a. offscreen surface). De DWM plaatst deze surfaces naar het scherm.

1.3 Gebruik van het `man`-commando

Het `man` commando laat ons toe de handleiding van commando's of programma's op te vragen.

1.3.1 `man ls`

`ls` is het commando om bestanden of directories op te lijsten. Enkele opties voor ls zijn:

`-l`
geeft een lange weergave van alle bestanden en directories in de huidige directory. In deze lange weergave zitten bijvoorbeeld de rechten.

`-a`
geeft alle bestanden en directories in de huidige directory weer, dus ook de hidden files

`-g`
doet hetzelfde als `-l`; maar toont de eigenaars niet

`-h`
kan enkel uitgevoerd worden in combinatie met -l of -s, en zorgt ervoor dat de grootte van bestanden en directories in voor mensen leesbare vorm wordt weergegeven

[1] TTY staat voor TeleType, dit is een virtuele controlling terminal die zich gedraagt als een hardware apparaat om input naar de machine te sturen. Vroeger was een TeleType een keyboard die geconnecteerd stond met een printer.

1.3.2 `man cp`

`cp` is het commando om bestanden of directories te kopiëren. Enkele opties voor cp zijn:

`-u`
het te kopiëren bestand wordt enkel gekopiëerd als het nieuwer is dan een eventueel al aanwezig bestand met dezelfde naam

`-v`
tijdens het kopiëren wordt uitgelegd wat gedaan wordt.

`-s`
in plaats van het bestand te kopiëren wordt een symbolische link gelegd naar het bestand.

`-l`
in plaats van het bestand te kopiëren wordt een harde link gelegd naar het bestand.

1.3.3 `man rm`

`rm` is het commando om bestanden of directories te verwijderen. Enkele opties voor rm zijn:

`-f`
het verwijderen van de bestanden of directories wordt geforceerd.

`-d`
alle lege directories in de huidige directory worden verwijderd.

`-i`
voor het verwijderen van elk bestand wordt een bevestiging gevraagd.

`-r`
recursief verwijderen van een map. Hierbij wordt ook alle inhoud van submappen verwijderd.

1.4 Commando tips and tricks

1.4.1 Commando regel wissen

Via de key-combinatie `<CTRL>-<U>` kunnen de inhoud verwijderen van de huidige commando regel.

1.4.2 Opdracht onderbreken

De key-combinatie `<CTRL>-<C>` breekt het huidige proces af. `<CTRL>-<Z>` zal een proces *suspenden*.

1.4.3 Autocomplete

Via de toets `<TAB>` kunnen we commando's of argumenten aanvullen. Via het dubbel indrukken van de `<TAB>` toets krijgen we een lijst van alle mogelijkheden.

1.5 Jokers/Wildcards

Jokers/Wildcards zijn karakters dat voor alle *members* van een bepaalde klasse van karakters kan staan. Wanneer een wildcard wordt gebruikt zal het computersysteem de *members* van de klasse van de wildcard karakter vervangen.

? Enkel (single) karakter behalve een punt

* Geen of meerdere karakters behalve een punt

[ ] definieert een klasse van karakters (voor een range gebruikt men '-' en '!' om uit te sluiten)

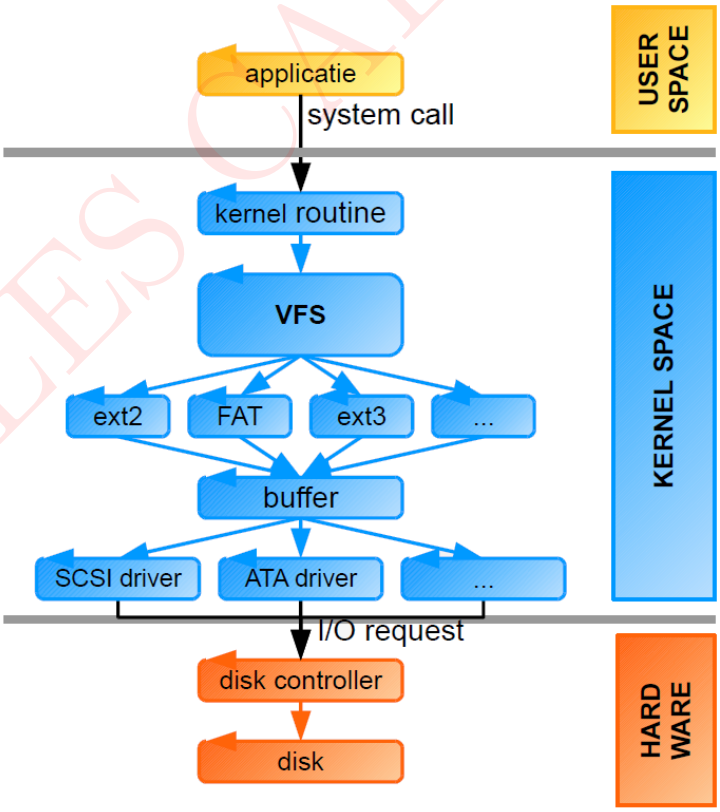
Voorbeelden:

Via het `cd` commando gaan we naar de `/etc` directory en voeren data het commando `ls *.conf` uit om alle bestanden met de extensie `.conf` weer te geven.

In onze root-folder tonen we alle bestanden uit 3 tekens met het commando `echo ???`. We kiezen het echo commando boven het `ls` commando vermits het `ls` commando ook de inhoud van de directories weergeeft.

1.6 Bestanden en directories

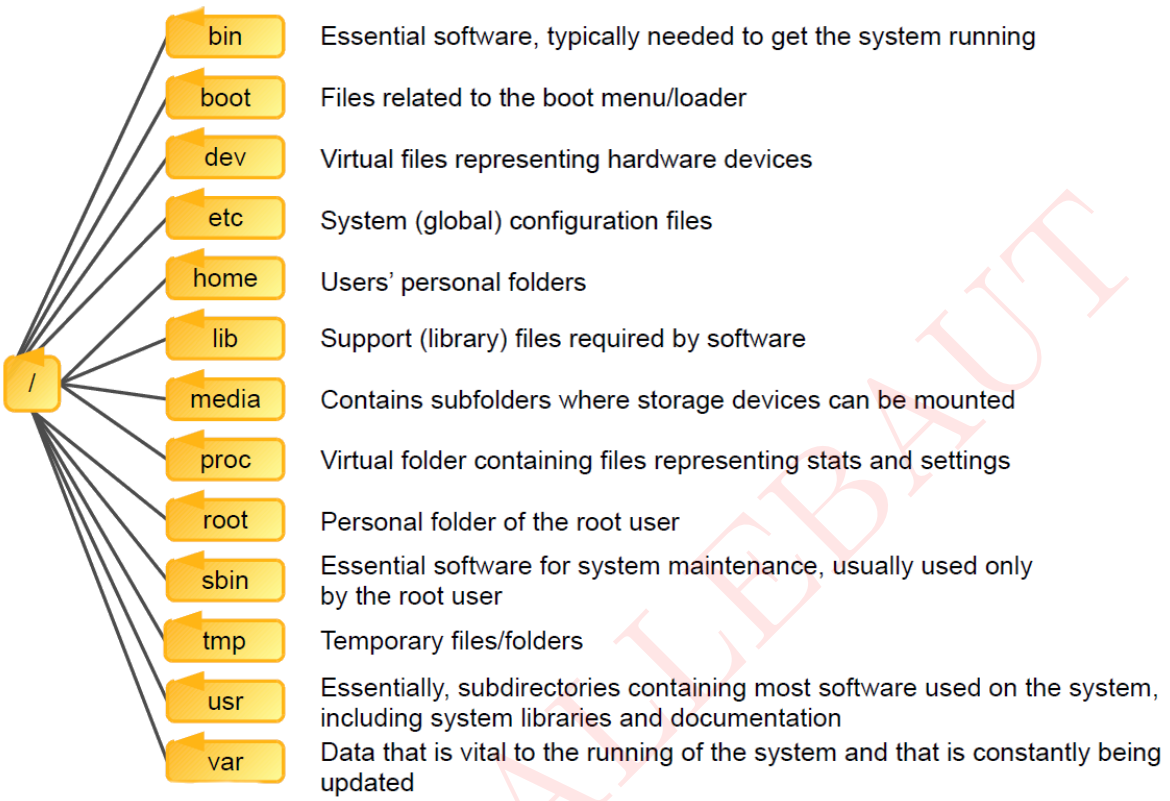
1.6.1 Virtual Filesystem (VFS)



Figuur 1: VFS

Het VFS is het filesysteem waar GNU/Linux gebruik van maakt. Dit is een boomstructuur met als wortel: root (/). In het VFS gaat men ook apparaten (CDROM, USB,..) en kernel

informatie tonen als een bestand. Apparaten staan in /dev en kernel info staat in /proc. Het doel van VFS is om ervoor te zorgen dat verschillende soorten files op 1 uniforme manier worden getoond, dan kunnen applicaties makkelijk overweg met verschillende files.



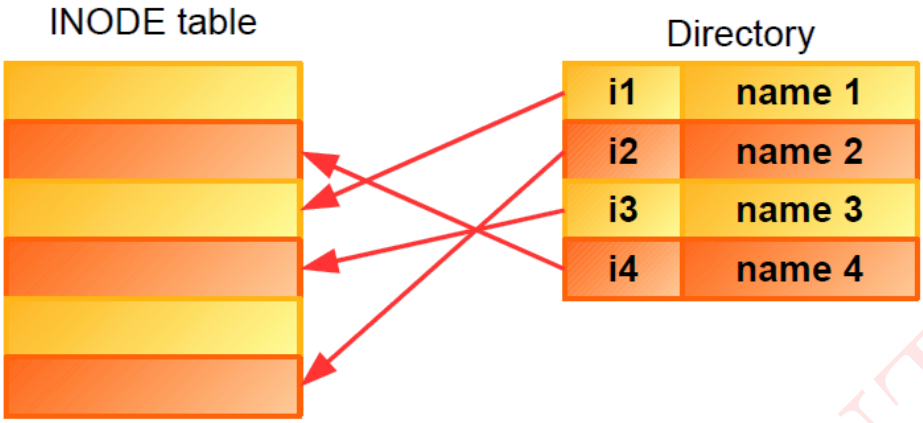
Figuur 2: VFS - belangrijke locaties

1.6.2 Inodes

Elk bestand in Linux bevat een inode. Deze inode bevat het bestandstype, toegangsrechten, eigenaars, timestamps, de grootte, en pointers naar data blokken (er zijn zowel indirecte als directe pointers naar datablokken). `ls -i` geeft de inodes weer van de inhoud van de huidige directory.

1.6.3 Directory

Een directory is een speciaal bestand die uit entries bestaat. Elke entry bevat de naam van de file/directory met de pointer naar de inode van deze file/directory.



Figuur 3: Directory

Home directory Via het `ls` commando tonen we alle bestanden in onze home directory. We kunnen naar de directory gaan door het commando `cd`. Om directory naar boven te gaan gebruiken we `cd ..` en naar beneden `cd <dir-name>`. We kunnen ook werken via het absolute pad: `cd /pad/naar/folder`. Via het commando `pwd` kunnen we het absolute pad verkrijgen van de huidige folder. We kunnen naar de home directory gaan met de commando's "`cd`"; "`cd ~`" en "`cd /home/student`" in dit geval.

1.6.4 Recursief bestanden tonen

Via het commando `ls -shR` kunnen we recursief alle bestanden tonen in 'human readable format', met de grootte erbij.

`-s, --size`
Geef de grote van elke file weer in blocks

`-h, --human-readable`
Toon de grote weer in human readable format (e.g., 1K 234M 2G)

`-R, --recursive`
Lijst alle subdirectories recursief op

1.6.5 Kopiëren en verplaatsen van/naar bestanden

We kopiëren het bestand `.bash_history` naar `testfile.txt` met het commando

```
cp .bash_history ./sysnetbeheer/labo2/testfile.txt
```

Extra info Het gebruik van `./` kan 2 betekenissen hebben:

1. Aantonen dat het gaat over een executable `./executable_file`
2. Aantonen dat het gaat over een pad dat begint bij de current directory `./PATH/to/...`

1.6.6 Recursief inhoud van een directory wissen

Wanneer we een directory en alle inhoud ervan willen verwijderen maken we gebruik van het commando `rm -r <NAAM_DIRECTORY>`.

`-i` voor het verwijderen van elk bestand wordt een bevestiging gevraagd.

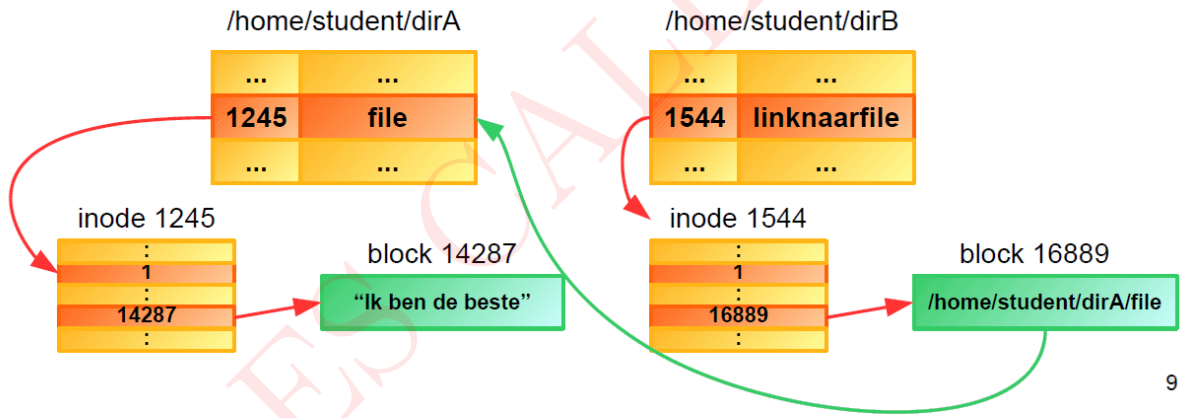
`-r` recursief verwijderen van een map. Hierbij wordt ook alle inhoud van submappen verwijderd.

1.7 Links

Links zijn verwijzingen naar andere bestanden. Naar programma's toe gedragen ze zich als gewone bestanden, in tegenstelling tot bijvoorbeeld shortcuts in Windows, die bestanden zijn die een verwijzing bevatten (m.a.w., ze zijn geen verwijzing, zoals in Unix). In Unix bestaat er nog een tweede type link, namelijk een hard link. Een hard link is in feite een bijkomende naam voor exact hetzelfde bestand.

1.7.1 Symbolische links

Symbolische links zijn bestanden die verwijzen naar padnamen (absoluut of relatief).



Figuur 4: Symbolische links

Via het commando `ln -s <src_file> <soft_copy_file>` maken we een symbolische koppeling van het eerste bestand naar het tweede.

```
ln -s ownertest.txt ot.txt
```

Output `ls -l`:

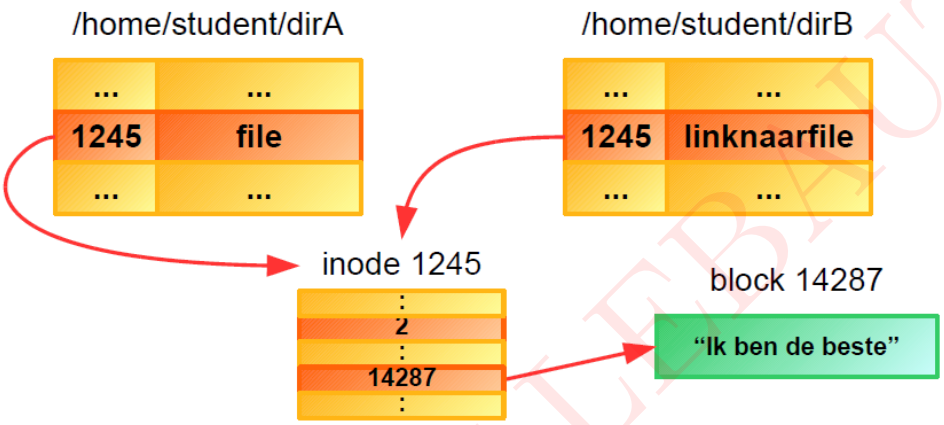
```
lrwxrwxrwx 1 student student    13 mrt  6 14:03 ot.txt ->
   ownertest.txt
```

Opmerking: Alle toegangsrechten zijn toegekend. Echter als we het bestand willen wijzigen krijgen we het volgende:
`[Fout tijdens lezen van ot.txt: Toegang geweigerd]` We zien dus dat de toegangsrechten voor het bestand (`ownertest.txt`) van toepassing zijn en niet de toegangsrechten van de link (`ot.txt`).

Als we de rechten aanpassen van `ownertest.txt` en dan de `ot.txt` file aanpassen. Zien we dat `ot.txt` een rechtstreekse link is naar `ownertest.txt` en wat we dus schrijven in de `ot.txt` file eigenlijk overeenkomt met het schrijven in de `ownertest.txt`. Vandaar de symbolische koppeling. `ot.txt` is dus een pointer naar het bestand `ownertest.txt`. De inhoud bij de inode nummer van `ot.txt` verwijst dan eigenlijk naar `ownertest.txt`.

1.7.2 Harde links

Een harde link is een entry in de directory, het verwijst rechtstreeks naar een inode van een bestand.



Figuur 5: Harde links

Wanneer we een harde koppeling maken zullen de 2 bestanden nu wel bestaan. Enkel hebben deze bestanden dezelfde inode nummer waardoor ze eigenlijk hetzelfde bestand zijn.
Harde link aanmaken:

```
ln ownertest.txt ot2.txt
```

Ze vertonen zich als 2 aparte bestanden, maar ze zijn eigenlijk één en hetzelfde bestand. Dit zagen we door het `ls -i commando`.

Inodes Zoals hierboven besproken zullen de symbolische gelinkte bestanden niet dezelfde inode hebben, en de harde gelinkte bestanden wel. We kunnen een bestandsnaam opvragen via zijn inode, dit via het volgende commando: `find . -inum <inum_number>`

1.8 Andere basisopdrachten

1.8.1 Info gebruikers opvragen

Via het commando `whoami` kunnen we de huidige gebruiksnaam opvragen. We kunnen ook info weergegeven krijgen van *system users* die recentelijk zijn ingelogd geweest op de terminal. Dit doen we via het commando `finger`.

```
Login            Name             Tty      Idle     Login    Time
student          Student          tty8     26       Feb 27   13:37
student          Student          pts/2             Feb 27   13:43
student          Student          pts/3             Feb 27   14:02
```

Extra info Een `tty` is een *regular terminal device*. Een `pts` is een *pseudo terminal slave*, een ssh connectie bijvoorbeeld.

1.8.2 More vs less

`more <FILE_NAME>` opent het bestand in de commandline, onder het ingegeven commando. `less <FILE_NAME>` zal het bestand in een leeg terminalvenster openen. In `less` is het mogelijk om terug naar boven te scrollen in het bestand, terwijl bij more alleen wordt toegelaten om voorwaarts te gaan. `less` leest ook niet heel de inputfile meteen bij het openen, hierdoor zal hij dus sneller openen.

1.8.3 Passwd

In het bestand `/etc/passwd` vinden we belangrijke informatie over alle users van het systeem. Deze informatie wordt gebruikt bij het inloggen.

1.8.4 Tail

Het commando

```
tail --lines=5 .bash_history
```

geeft de laatste 5 regels van het `.bash_history`-bestand weer. Dit bestand bevat de laatst gebruikte commando's. (opgelet: de commando's worden hier pas in opgeslagen wanneer de shell, waarin ze zijn uitgevoerd, afgesloten wordt). De optie `-f` zorgt dat het in realtime mogelijk is om de laatst uitgevoerde commando's te bekijken.

1.8.5 Kernelgrootte opvragen (Disk usage)

Via de commando-lijn hieronder komen we te weten dat de grootte van de kernel 5.6 MB is.

```
du -ah /boot/vmlinuz*
```

De opties zijn respectievelijk verantwoordelijk voor het weergeven van alle files en niet enkel directories, en dit alles in een human-readable formaat.

```
gilles@gilles-VirtualBox:~$ du -ah /boot/vmlinuz*
6,4M    /boot/vmlinuz-3.19.0-15-generic
6,4M    /boot/vmlinuz-3.19.0-18-generic
```

1.8.6 Vrije ruimte weergeven (Disk free)

```
df -kh --total
```

leert ons dat we nog 49% van onze schijf beschikbaar hebben, ofwel 24 GB.
De opties zijn respectievelijk verantwoordelijk voor het weergeven van block-sizes van 1k, en dit alles in een human-readable formaat. De `total` optie geeft ook het totaal weer.

```
Filesystem      Size  Used Avail Use% Mounted on
udev            486M     0  486M   0% /dev
tmpfs           100M  4,9M   95M   5% /run
/dev/sda1       6,8G  4,2G  2,3G  65% /
```

```
tmpfs            497M   80K  497M   1% /dev/shm
tmpfs            5,0M  4,0K  5,0M   1% /run/lock
tmpfs            497M     0  497M   0% /sys/fs/cgroup
cgmfs            100K     0  100K   0% /run/cgmanager/fs
tmpfs            100M   44K  100M   1% /run/user/1000
total            8,5G  4,2G  4,0G  52% -
```

1.8.7 Comprimeren

```
tar -zcvf workspace.tar.gz workspace
```

-z compress via gzip programma

-c create archive

-v verbose (toon progress)

-f (archive File name)

1.9 Redirection

De meeste UNIX system commando's nemen input van de terminal en zenden de resulterende output terug naar de terminal. De terminal is dus default de standaard in- en output.

1.9.1 Output redirection

De output van een commando die normaal voor de standaard output bedoelt was kan worden doorgestuurd naar een file, dit via de `>` notatie. We kunnen gebruik maken van `>>` om de output toe te voegen aan het bestand i.p.v. het te overschrijven.

Wegschrijven naar een bestand

```
ls -a > dit.txt
```

Met de optie `-a` zullen alle entries af geprint worden, ook degenen die beginnen met '.'.

Kopiëren van een bestand en de inhoud samenvoegen

```
cp       dir.txt oudelijst.txt   &&
cat -n   dir.txt oudelijst.txt   >       samengesteld.txt
```

Als `oudelijst.txt` als bestaat zal `cp` de inhoud van `oudelijst.txt` vervangen door de inhoud van `dir.txt`. Als `oudelijst.txt` nog niet bestond werd die nu aangemaakt met de inhoud van `dir.txt`. Via het commando `cat -n`, printen we de inhoud van `dir.txt` en `oudelijst.txt` (*concatenate*) uit naar de standaard output en nummeren we deze inhoud (optie `-n`). Als we deze output nu *redirecten*, kunnen we de inhoud opslaan in het bestand `samengesteld.txt`.

Uitvoer laten 'verdwijnen'

```
ls -lR > /dev/null
```

`ls -lR` print alle lijnen in het lang en recursief.

Foutmelding wegschrijven
Wanneer we de inhoud van een onbestaand bestand proberen weer te geven, krijgen we de melding: "`onbestaand: Bestand of map bestaat niet`" We kunnen deze foutmelding naar een file opslaan met het commando:

```
more onbestaand 2> resultaat.txt
```

De 2 staat hierbij voor het uitschrijven van de *error stream*.

1.9.2 Input Redirection

Net zoals de output van een commando kan worden weggeschreven naar een file, kan de input van een commando worden gebruikt van een file. Hierbij gebruiken we nu het '<' teken.

Voorbeeld: het commando:

```
sort < file_list.txt
```

We kunnen dan de input en output redirectie combineren:

```
sort < file_list.txt > sorted_file_list.txt
```

1.10 Pipes

Een krachtigere versie van I/O redirectie is het gebruik van pipes. Hierbij gaan we meerdere commando's met elkaar connecteren door *pipes*. De standaard output van het ene commando wordt gevoed als standaard input van het volgende commando.

1.10.1 Zoeken zonder commando find

We zoeken via `ls` naar alle bestanden en directories en zoeken via grep naar "`.bash_history`". we kunnen dit ook verwezenlijken door `locate .bash_history`. Of

```
ls -a | grep .bash_history
```

1.10.2 Inhoud weergeven en filteren

We zoeken in het bestand `/etc/passwd` naar de student-login met het commando:

```
more /etc/passwd | grep student
```

Een andere mogelijkheid om hetzelfde resultaat te bekomen is "`grep student /etc/passwd`" of `cat /etc/passwd | grep student`.

1.11 Tekstbestanden manipuleren via pipes en redirecties

1.11.1 Aanmaken van tekstbestanden

Vi We kunnen pas iets typen in `vi(m)`[2] wanneer we op `i` drukken. Om iets op te slaan drukken we eerst op `<ESCAPE>` zodanig dat we de input-modus verlaten en typen het commando "`:w`" in op het bestand op te slaan. We kunnen een lijn verwijderen met het commando "`:d`". We kunnen het karakter links of rechts van de cursor verwijderen met het commando "`X`", respectievelijk "`x`". '`wq test2.txt`' zorgt ervoor dat we de inhoud schrijven naar `test2.txt` en sluit `vi` af.

Afsluitmodes van vim

`:q` afsluiten

`:q!` afsluiten zonder wijzigingen

`:wq` afsluiten met wijzigingen

Modes van vim

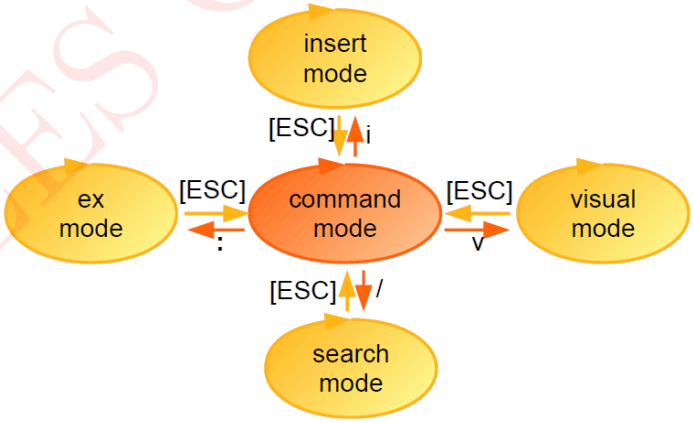
Command mode: opslaan, afsluiten,...

Insert mode: editeren, tekst intypen

Ex mode: last line mode

Visual mode: navigatie met pijltoetsen

Search mode: zoekterm intikken



Figuur 6: Vi(m) modes

touch Het `touch` commando is de eenvoudigste manier om een nieuw leeg bestand aan te maken.

```
touch test.rot
```

[2]Visual Interface improved

1.11.2 Zoeken in tekstbestanden

'`find`' print de naam van het bestand uit indien het bestaat, en "`Bestand of map bestaat niet`" indien het niet bestaat. Voorbeelden:

```
find /etc -name 'inetd.conf'
```

Opmerking: De backquotes zijn hier niet verplicht, we moeten ook de optie `-name` gebruiken, om te zoeken op een patroon.

We zoeken naar alle bestanden die een a of z bevatten met het commando "`find *a* *z*`".

Zoeken van bestanden en deze gelijktijdig verwijderen

```
find . -type f -name "*.rot" -exec rm -f {} \;
```

Zoeken van bestanden en de inhoud ervan weergeven
We zoeken het bestand `menu.lst` met het commando:

```
find -name menu.lst | xargs cat
```

of

```
find -name menu.lst -exec cat {} \;
```

1.12 Rechten

1.12.1 Standaard rechten

De standaard rechten bij het aanmaken van een bestand zijn `-rw- r-- r--`. Als we te maken hebben met eerst een streepje geeft dit weer dat het om een bestand of toepassing gaat. Bij een toepassing zal de owner standaard `rwx`-rechten hebben, bij een bestand enkel `rw`-rechten. Indien we met een device zitten zal dit streepje vervangen worden door een '`b`' en bij een directory door een '`d`'.

UUU	Owner	rw-
GGG	Group	r–
OOO	Others	r–

Tabel 1: Standaard rechten per groep

1.12.2 Wijzigen van rechten

Via het commando `chmod` is het mogelijk om rechten te wijzigen. Dit kunnen we op 2 manieren:

- Octale-methode
  `chmod [0-7] [0-7] [0-7] <file_name>`
  Met waarden r=4, w = 2 en x=1.

- String-methode
  `chmod [ugoa] +,-,= [rwx] <file_name>`
  + toevoegen, - wegnemen, = gelijkstellen van permities
  Voorbeeld: `chmod g+r bestand`
  Opmerking: a staat voor all users.

1.12.3 Aanmaken dropbox dir

Een "drop box" directory is een directory waarin je bestanden kan schrijven, maar waarin je niet kan lezen.

```
mkdir dropbox
chmod 222 dropbox
```

of

```
mkdir -m 222 dropbox
```

Door de `-m` optie kunnen we octale rechten meegeven aan de directory.

2 Processen

Een proces stelt een lopend programma voor, het gebruikt geheugen, processor- en I/O resources en heeft een prioriteit en wordt beheert door het procesbeheer.

2.1 Oerproces

Het init proces (PID=1) is het eerste proces, alle andere processen zijn kinderen van init. Het heeft dan ook als hoofdtaak, andere processen creëren.

2.2 Exec vs fork

Elke proces heeft een PID (process identifier) en kan een `fork` of `exec` instructie uitvoeren. Bij een `fork` instructie zal er een kind proces worden gecreëerd (met een verschillende PID). Bij een `exec` instructie zal het proces zichzelf vervangen en wordt het PID behouden.

2.3 het `ps` commando

Via het `ps` commando kunnen we alle uitvoerende processen bekijken.

`ps -A`	alle actieve processen
`ps -U <login_name>`	eigen processen
`ps aux`	alle processen
`ps -af`	hiërarchie van processen

2.4 Stoppen, starten naar achtergrond brengen van processen

Stoppen voorgrondproces
We kunnen via de toetsencombinatie `<CTRL>-<c>` het voorgrond proces beëindigen. Met de toetsencombinatie `<CTRL>-<z>` zullen we het proces stoppen, maar niet beëndigen.

Proces op achtergrond starten
via het commando "`<proces_name> &`" kunnen we het proces op de achtergrond starten.

Proces op achtergrond weergeven
We kunnen de achtergrondprocessen bekijken door het "`jobs`" commando.

Achtergrondproces naar de voorgrond brengen
We kunnen een achtergrondproces naar de voorgrond brengen door het commando "`fg`". Het omgekeerde doen we door het commando "`bg`".

2.5 Systeem-processen en daemons

Daemons zijn achtergrond processen in Linux. Deze processen eindigen op een '`d`', voorbeeld `/etc/init.d`. De meeste *daemons* aanvaarden argumenten zoals start, stop, status en restart. Een voorbeeld is " `/etc/init.d.networking restart`".

Het init proces start (symbolische links) script op in `/etc/rcX.d`, waarbij `X` staat voor het runlevel. Hoe hoger het runlevel, hoe meer functionaliteit. De symbolische links bestaan uit:

- een letter K of S

  De K staat voor het stoppen van de dienst bij het verlaten van het runlevel.

  De S staat voor het starten van een dienst bij het betreden van het runlevel.

- een getal tussen 0 en 99
  De cijfers 0..99 bepalen de volgorde van oproep.

- de naam van de dienst

Voor de typische runlevels in Linux zie tabel 2 op pagina 36.

3 Regulaire expressies (Labo 3)

Reguliere expressies zijn een krachtigere versie van wildcards: ze zijn een tool die gebruikt kan worden om efficiënt naar patronen te zoeken in teksten. Zie tabel 3 op pagina 64.

3.1 Basis regulaire expressies

3.1.1 Beschrijving strings

Enkele voorbeelden van het beschrijven van strings d.m.v. regulaire expressies:

`v_[0-9]+`
String: `v_<onbepaald aantal nummers, met aantal groter dan 0>`

`v_[0-9]+(_[0-9]+)`
Deze regex beschrijft elke string die een substring van dezelfde vorm van de eerste regex bevat, waaraan nog een substring van het type "`_<onbepaald aantal nummers, met aantal groter dan 0>`"hangt.

`v_[0-9]+(_[0-9]+){0,1}`
Deze regex is een combinatie van de vorige twee. Het deel uit de eerste regex is altijd aanwezig, maar het extra deel uit de tweede regex is ofwel wel ofwel niet aanwezig.

3.1.2 Belgisch rijksregisternummers

Belgische rijksregisternummers bestaand uit [6 cijfers - spatie - 3 cijfers - spatie - 2 cijfers]. Hierbij kunnen we twee regexen vormen:

```
\d{6}\s \d{3} \s \d{2}
```

```
^[0-9]{6} [0-9]{3} [0-9]{2}$
```

Voorbeeld: (match) 165569 654 22

3.1.3 Eenvoudige herkenning e-mailadressen

```
^[0-9a-zA-Z\.]+@([0-9a-zA-Z]+\.)+[0-9a-zA-Z]$

[a-zA-Z0-9]+@[a-zA-Z0-9]+\.[a-zA-Z0-9]+

^[a-z0-9]+(\.[a-z0-9]+)*@[a-z0-9]+(\.[a-z0-9]+)*(\.[a-z
  ]{2,6})
```

We moeten steeds beginnen met 1 of meer letters/cijfers. Daarna komt een uitdrukking die voldoet aan nul of meer keer .cijfer of .letter een @ teken en dan moet weer 1 of meerdere keren een letter of cijfer voorkomen en mag je nog een uitdrukking met '.letter' of '.cijfer' 0 of meerdere keren gebruiken vooraleer af te sluiten met .cijfers met cijfers minimum 2 en maximum 6.

3.1.4 Belgische telefoonnummers

```
^((0032)|(\+32))[1-9]{2}\d{6})|^(0[1-9]\d{7}
```

We houden geen rekening met zonenummers die niet geldig zijn (tussen 10 en 09).

3.2 Zoeken in bestanden

Een tweede gebruik van reguliere expressies is patronen zoeken in een string. We gaan dus niet de volledige string matchen, maar zoeken in de string of een bepaald patroon erin voorkomt. Dit is nuttig om te zoeken in teksten of bestanden. Een belangrijk basiscommando op Linux om patronen te zoeken m.b.v. reguliere expressies is `grep`.

3.2.1 Uitprinten van alle lijnen uit een bestand

We beelden alle lijnen af uit het bestand /etc/network/interfaces met daarin het woord iface met het commando:

```
grep -wo iface interfaces
```

`-w` zoek op woord

`-o` printen van match, niet heel de lijn tonen

3.2.2 Zoeken naar patronen in een bestand

- We maken het bestand "`boek.txt`" aan met daarin enkele contacten.

  personen wiens naam een a bevat:
  `grep a boek.txt`
  regels die beginnen met een D:
  `grep ^D boek.txt`
  regels die eindigen met 75:
  `grep 75$ boek.txt`
  telefoonnummers uit de zone 09:
  `egrep 09 [0-9]{7} boek.txt`
  telefoonnummers niet uit de zone 09:
  `egrep [^09] [0-9]{6}[0-9]?  boek.txt`

  Bij het laatste konden we ook gebruik maken van de parameter '`-v`' die de selectie inverteert, deze selecteert non-matching lijnen: `grep -v 09 [0-9]{7} boek.txt` Probleem hierbij is dat er ook niet-telefoonnummers kunnen worden geselecteerd.

- Zoeken van patronen in `bash.bashrc`
  Opmerking: `grep` is *case sensitive*

  Alle lijnen die beginnen met a,b,c,d of e:
  `grep ^[a-e] bash.bashrc`
  Alle lijnen die zowel if als HOME bevatten, ongeacht de volgorde:
  `grep if testfile.txt | grep HOME`
  of
  `egrep HOME.*if|if.*HOME bash.bashrc`
  Alle lijnen die "term" bevatten, en "term" moet case insensitive gezocht worden:
  `grep -i term bash.bashrc` [**-i, –ignore-case**]
  Alle lijnen die eindigen op een punt:
  `grep '\.$' bash.bashrc`
  Opmerking: Hier moeten de backquotes expliciet staan!

3.2.3 Zoeken naar bestanden, zoeken naar lijnen en deze uitschrijven

We zoeken met `find` alle bestanden in `/etc` met `passwd` in de bestandsnaam. Om via de resultaten van `find` alle lijnen op te slaan waar `student` in voorkomt in het bestand `passstudent.txt`.
2 mogelijkheden:

`find /etc -name passwd -exec egrep 'student' {} \;`
Hierbij zoeken we alle bestanden met passwd in de bestandnaam met "`find`" en de lijnen met student met het commando "`egrep`". -exec roept een subprocess op dat gebruik maakt van de resultaten van "`find`'.

```
find /etc -name ''*passwd*'' | xargs grep student >
   passstudent.txt
```

xargs: De resultaten van het find commando worden nu als argument meegegeven aan het grep commando.

3.3 Enkele andere gebruiken - Filteren

Uitfilteren processen uit `/usr/bin` en `/usr/sbin`

```
ps aux | grep '/usr/bin\|/usr/sbin'
```

of korter:

```
ps aux | grep '/usr/\?bin'
```

Uitfilteren regels die beginnen met `cpu` of `fpu` in `/proc`

```
grep '^\(cpu\|fpu\)' cpuinfo
```

of:

```
grep '^[c|f]pu' cpuinfo
```

4 Virtualisatie (Labo 4)

Virtualisatie is steeds meer een hot topic. Dit is vooral zo in servermiddens, waar virtualisatie het mogelijk maakt om verschillende virtuele servers te consolideren op één hardware-server. Maar ook op thuis-PC's vindt virtualisatie meer en meer ingang, aangezien de laatste jaren ook de processoren voor deze machines virtualisatie steeds meer in hardware ondersteunen. Je kan bijvoorbeeld virtualisatie gebruiken om op je Mac of Linux PC ook een Windows installatie te hebben of viceversa, en beide tegelijk te gebruiken. Dit in tegenstelling tot dual-boot, waar je het ene OS moet verlaten om in het andere te kunnen werken.

4.1 Installatie

Installeren van VirtualBox en Debian ISO file koppelen aan de virtuele DVD-lezer. Nadien opstarten van de VM en Debian installeren.

4.2 Snapshots

We maken een snapshot van de huidige toestand van het systeem zodat, indien er iets misloopt, we de machine nog steeds naar een stabiele toestand kunnen terugbrengen. Een snapshot is dus een staat van de virtuele machine. Om dit te testen verwijderen we met het commando "`rm -rf *`" (opties: recursief, force) alles wat als *superuser* kan verwijderd worden. Hierdoor werkt het besturingssysteem niet meer als gewenst. Als we de VM opnieuw opstarten krijgen we volgende error:
"`GRUB Loading stage1.5 GRUB loading, please wait...  Error 15`"

Nu kunnen we de staat van het besturingssysteem eenvoudig terugzetten naar voor het verwijderen van de bestanden door de snapshot terug in te laden.

4.3 Verschilt dualboot en virtualisatie

Er is een duidelijk verschil tussen virtuele machines en dual booting. Dual booting betekent het installeren van een tweede besturingssysteem op een aparte partitie van een harde schijf (of op een andere harde schijf). Bij het opstarten van het systeem wordt dan ook gevraagd welk operating system je wilt starten, terwijl een virtuele machine draait op een besturingssysteem. Door dat besturingssysteem wordt dan het andere besturingssysteem in een venster geëmuleerd. Men kan dankzij virtualisatie dus op één machine meerdere besturingssystemen tegelijk draaien.

5 Shell Scripting (labo 5)

Een shell is een interactief computerprogramma waarmee een gebruiker met een CLI (command–line-interface) een opdracht kan geven aan het besturingssysteem van een computer.

Shell scripts zijn een krachtige uitbreiding van wat je normaal in de command line kan doen in Linux. Ze zijn eigenlijk een bijna volwaardige programmeertaal, die je kan gebruiken om allerlei administratie-taken te automatiseren. Je kan ze ook gebruiken om andere bewerkingen, zoals bijvoorbeeld zoeken in bestanden of het analyseren van log-files, te automatiseren zodat je dit meerdere keren kan doen zonder telkens alle commando's opnieuw in te moeten typen. De bouwstenen van shell scripts zijn enerzijds een aantal ingebouwde commando's en constructies in de BASH shell, zoals if, case en ls, en anderzijds alle externe commando's die ter beschikking staan in Linux, zoals find, grep of awk. Aangezien je gemakkelijk zelf nieuwe externe commando's kan maken met C, Java of een andere programmeertaal, kan je de shellprogrammeertaal ook zelf uitbreiden.

Scripts zijn in feite gewone tekstbestanden, die een speciale hoofding krijgen die aangeeft welke interpreter aangeroepen moet worden om het script uit te voeren, en die een speciale execute permissie hebben die aangeeft dat ze als programma uitgevoerd mogen worden.

5.1 Script aanmaken

We maken een script aan `<script_name>.sh`.
De eerste lijn begint altijd met een sha-bang (dit is de #!) gevolgd door een pad. Dit pad geeft ons weer welke interpreter we gaan gebruiken. In het labo maakten we gebruik van #!/bin/sh, hierdoor gaan we onze file gaan uitvoeren met de Broune shell. Indien we als eerste lijn #!/bin/perl gebruiken, gaan we onze file uitvoeren met de interpreter Perl. We moeten het script wel nog executierechten toekennen, `chmod +x <script_name>.sh`.

Argumenten We kunnen aan de argumenten via '$?', '$0' geeft de scriptnaam weer, de volgende integers bepalen de argumenten ($1, $2, $3, ...).

Speciale variabelen $ is om de waarde van een variabele te gebruiken

$#	dit geeft ons het aantal parameters
$0	dit geeft ons de naam van het script
$1	dit geeft ons de eerste parameter
$$	dit geeft ons het PID (Process ID) van de shell
$HOME	dit geeft ons de home directory

5.2 Eenvoudige scripts

5.2.1 Externe commando's aanroepen

Grote home dir weergeven

```
#!/bin/bash
a=$(du -sk /home | awk '{print $1}')
```

```
echo Total size of all home dirs: $a
```

Datum en huide dir weergeven

We maken een script `datedir.sh`. Dit commando geeft de huidige datum weer (door het "`date`" commando te gebruiken) en de huidige directory weer (door het "`pwd`" commando te gebruiken).

```
#!/bin/bash
echo $(date)
echo $(pwd)
```

We maken een tweede script `datedir2.sh`. We maken hier gebruik van de backqoutes tekens om de commando's "`date`" en "`pwd`" uit te voeren. We kunnen externe commando's uitvoeren door het gebruik van backqoutes in combinatie met assignment. We gebruiken backqoutes als we returnwaarden verwachten. Dit is dus gelijkaardig aan het vorige voorbeeld.

```
#!/bin/bash
echo De huidige datum is : `date`
echo De huidige directory is : `pwd`
```

Bestand of directory uitprinten (ls/more)

We maken een script `see`, dat als argument een bestandsnaam meekrijgt en directories met `ls` weergeeft, terwijl bestanden met `more` uitgevoerd worden. We maken hierbij gebruik van een if-structuur en de opties "`-d`" en "`-f`" die respectievelijk staan voor "`directory`" en "`file`".

```
#!/bin/bash
param=$1
if [[ -d $param ]]
        then
                ls $param
elif [[ -f $param ]]
        then
                more $param
else
        echo "not a file or a directory"
fi
```

Argumenten uitprinten

We maken een script dat alle argumenten op het scherm afprint. Dit script noemen we `for.sh`, aangezien we gebruik maken van een for-lus.

```
#!/bin/bash
        for i in $@
                do
                echo $i
        done
```

```
#!/bin/bash
        for var; do echo  "$var"; done
```

```
#!/bin/bash
        for i
                do echo $i
        done
```

awk
We printen voor alle processen de gebruiker en de procesnaam af met behulp van `awk`. `awk` is een programmeertaal die gebruikt kan worden om gemakkelijk met tekst te werken.

```
ps aux | awk '{print $1 "\t-> \t" $11}'
```

Waarbij &1 en &11 respectievelijk de kolom van de gebruiker en procesnaam is.

5.3 Geavanceerde mogelijkheden

5.3.1 Toevoegen van datum in bestand

```
#!/bin/bash
        echo $(date) >> /tmp/dates.txt
```

of

```
#!/bin/bash
        var = $(date)
        cat /tmp/dates.txt $var
```

5.3.2 Crontab

Het `crontab` (*cron* komt van *chronos*, Grieks voor tijd; *tab* staat voor *table*) commando, wordt gebruikt om commando's periodisch uit te voeren.

Lopende crontabs `crontab -l`

Editeer crontabs `crontab -e * * * * * /bin/execute/this/script.sh`
Die vijf sterren representeren om welk tijdstippen het script moet worden uitgevoerd:

* minute (from 0 to 59)
* hour (from 0 to 23)
* day of month (from 1 to 31)
* month (from 1 to 12)
* day of week (from 0 to 6) (0=Sunday)

Door de 5 sterren zal het script elke minuut worden uitgevoerd. We kunnen ook een interval definiëren om bijvoorbeeld het script enkel van maandag tot vrijdag uit te voeren.
`0 1 * * 1-5 /bin/execute/this/script.sh`
Het script zal uitgevoerd worden op:

- minute: 0
- of hour: 1
- of day of month: * (every day of month)
- of month: * (every month)
- and weekday: 1-5 (=Monday til Friday)

Als we een script willen uitvoeren om, bijvoorbeeld, 10 minuten dan gebruiken we de volgende regel: `*/10 * * * * /bin/execute/this/script.sh`

6 Package Manager

De packet manager is een stuk software dat zorgt voor het automatisch beheer van geïnstalleerde software (verwijderen/updaten). Sommige programma's hebben andere programma's nodig om te kunnen functioneren. De packet manager zal ervoor zorgen dat alle nodige software is geïnstalleerd door te kijken naar de dynamische linken van dit programma.

De package manager zorgt er ook voor dat verschillende systeemarchitecturen worden ondersteund door configuratiebestanden te schrijven. Dit configuratiebestand zorgt ervoor dat alle nodige libraries voor het programma aanwezig zijn voordat we beginnen met het compileren van het programma. Bij updates zit ook hier vaak het probleem namelijk moet de oude config file weg of moeten deze geüpdatet worden? Als we de oude config vervangen verliezen we namelijk de user config, en de oude file updaten is complex. Debian behoudt bij apt-get de config file als deze aanwezig is, en als ze niet aanwezig is maakt hij één aan. Indien we in Debian apt-get gebruiken, roepen we feitelijk de package manager dpkg (Debian Package Manager) op.

7 Server Services (Labo 6-7)

7.1 SSH

SSH staat voor Secure SHell; dit houdt in dat we een Terminal-venster kunnen openen waarin we via de command-line opdrachten kunnen laten uitvoeren door de machine. In tegenstelling tot een normaal Terminal-venster echter, kunnen we via SSH opdrachten geven aan andere dan de eigen machine.

SSH maakt gebruik van een public key en een private key. De public key dient enkel voor te decrypteren, de private key dient voor te encrypteren. Bij SSH wordt er automatisch een keypair gegenereerd om de netwerkconnectie te encrypteren.

7.1.1 SSH installeren

Op Debian machines gebruiken we het commando `apt-get` om pakketten te installeren d.m.v. een package manager. We willen via de host-machine remote in loggen op onze virtuele server. Voor we dit proberen maken we een snapshot van de VM.

1. VM
   We bepalen het IP-adres van de VM via het commando (sudo) `ifconfig`. Het IP-adres is `10.128.48.44`.

2. HOST
   We proberen te connecteren met onze VM via ssh `<username_on_VM>@<IP_VM>`. We krijgen echter: `connection refused` (de SSH is nog niet geinstalleerd op de VM)

3. VM
   Installeren van het pakket SSH d.m.v. de packet manager van Debian `sudo apt-get install ssh`.

4. HOST
   herhalen stap 2 en krijgen nu wel toegang tot de SSH op de VM.

5. HOST
   `uname -a` [-a staat voor ALL (alle info tonen)]
   Output:

   ```
   Linux debianGillesJelmer 2.6.26-2-686 #1 SMP Mon Aug
   30 07:01:57 UTC 2010 i686 GNU/Linux
   ```

   Hiermee tonen we aan dat we op de VM zijn ingelogd via ons HOST-machine.

7.1.2 Andere software in het SSH-pakket

Het SSH-pakket bevat een `SSH daemon`. Dit programma (`/usr/sbin/sshd`) draait in de achtergrond, en laat toe om vanop een andere machine in te loggen op (bv.) onze VM. Om aan te tonen dat `sshd` draait voeren we het volgende commando uit:

```
ps aux | grep sshd
```

Naast SSH zijn er nog andere programma's die SSH als protocol gebruiken, deze zijn `SCP` (Secure Copy) en `SFTP` (Secure File Transfer Protocol). Deze worden gebruikt om beveiligd bestanden van en naar een remote host te up- of downloaden.

SCP
HOST:

```
scp passstudent.txt jelmergilles@10.128.48.44:/home/
   jelmergilles/
```

Output:

```
jelmergilles@10.128.48.44's password:
passstudent.txt          100%   68      0.1KB/s   00:00
```

Via dit commando zijn we in staat op het bestand `passstudent.txt` (van de HOST) up te loaden naar onze VM (path: `/home/jelmergilles/Desktop`).

SFTP
HOST:

```
sftp jelmergilles@10.128.48.44 put oudelijst.txt
```

Output:

```
Uploading oudelijst.txt to /home/jelmergilles/Desktop/
   oudelijst.txt
oudelijst.txt            100%  425      0.4KB/s   00:00
```

We kopiëren dus het bestand `oudelijst.txt` van de host naar de VM. Om te navigeren gebruiken we volgende commando's:

- `lls`: local ls
- `lcd`: local cd
- de andere commando's (`ls` en `cd`) zijn degenen die invloed hebben op de remote PC.

7.1.3 Hostname gebruiken

We kunnen ook een host-name gebruiken om toegang te krijgen tot onze VM. We openen de hosts file via nano (`nano /etc/hosts`). En voegen de volgende lijnen toe:

```
REMOTE machine:
10.128.48.44     debianGillesJelmer..kuleuven
   virtualmachine
LOCAL machine:
10.128.48.44     virtualmachine
```

Vanaf nu zijn we in staat op een ssh connectie op te starten via het volgende commando:

```
ssh jelmergilles@virtualmachine
```

We mogen dit uiteraard enkel toepassen op IP-adressen die statisch zijn toegewezen. Anders zal men telkens opnieuw de hostfile moeten aanpassen.

7.2 Port scans

Een server levert verschillende server-diensten via dezelfde netwerk-interface. Om dit in goede banen te leiden, is een netwerkinterface software-matig opgesplitst in poorten. Elke service heeft dan één of meerdere poorten via dewelke het zijn diensten levert. Bijvoorbeeld: SSH in de bovenstaande opdrachten levert zijn service via poort 22. De meeste services hebben een vast poortnummer, dan bij conventie vastligt.

7.2.1 Installeren nmap

Via het commando nmap kunnen we poorten scannen. We installeren nmap met het commando: `sudo apt-get install namp`.

7.2.2 Poorten scannen van een server

We scannen de poorten van de server `scanme.nmap.org` via het volgende commando:

```
nmap scanme.nmap.org
```

Output (vereenvoudigd):

```
22        tcp/open ssh
80        tcp/open http
```

7.3 Heartbleed

`SSL` (Secure Sockets Layer) is een software-pakket dat geëncrypteerde communicatie toelaat tussen twee computers. Het maakt gebruik van standaard netwerkprotocollen (zoals `TCP` of `UDP`), maar voegt daar een onderhandelingsprotocol en functionaliteit aan toe waardoor de twee computers in staat zijn om een netwerkverbinding te maken waarvan de informatie die erover verzonden wordt niet te decoderen is door een derde partij die de communicatie tussen de twee computers kan onderscheppen. Op die manier kan gevoelige informatie verzonden worden zonder het risico te lopen dat deze kan gelezen worden door anderen dan de zender en de ontvanger.

7.3.1 "Common Vulnerabilities and Exposures" (CVE)

Heartbleed op `nvd.nist.gov`

CVE-nummer: CVE-2014-0160

Ernst van de bug: 5.0 (MEDIUM)

Hoe lang bestaat de bug al?
Sinds 04/07/2014 bestaat de bug al
Revised: 03/23/2015

Lage ernst-score, waarom veel commotie?
Heartbleed heeft een relatief lage ernst-score omdat hun score enkel de impact van de vulnerability op de host waarop de vulnerability gelocaliseerd is waardeert. Heartbleed geeft geen onbeperkte toegang tot het geheugen van de host, maar kan toegang geven tot gevoelige informatie zoals encryptie-sleutels en passwoorden. Aan de hand van die sleutels kan dan communicatie afgeluisterd worden. Het grote probleem is

ook het gemak van het misbruiken van de bug, de vele programma's die die OpenSSL lib. gebruiken en de daders kunnen afluisteren zonder iets achter te laten.

7.3.2 Heartbleed-bug

De bug komt uit de heartbeat-extensie. Heartbleed is een foute implementatie van een memory-buffer die gebruikt wordt om een communicatielink te testen. Een client stuurt een string en lengte van de string (default 6) door naar de server, die exact dezelfde string moet terug sturen. Er wordt echter niet gecontroleerd of de meegegeven lengte van de string even groot is dan de werkelijke hoeveelheid karakters. Indien er een grote waarde wordt gestuurd dan stuurt de server als response de string terug met daarachter de eerstvolgende tekens uit het actieve geheugen. Deze tekens kunnen gevoelige informatie bevatten over paswoorden en cryptografische sleutels. Een aanvaller kan niet kiezen welke *content* teruggestuurd wordt, maar kan zo wel gevoelige informatie terugkrijgen.

7.3.3 Heartbleed op onze server

OpenSSL versies met bug
De OpenSSL versies die de bug bevatten zijn 1.0.1 tot 1.0.1f.

Versie OpenSSL op VM
Op onze server, met besturingssysteem Debian 5.06, staat OpenSSL version:0.9.8g 19 Oct 2007. Deze versie is niet getroffen door Heartbleed. Versie van OpenSSL bekijken: We gebruiken het commando "`dpkg -s openssl`" om de versie van openssl te bekijken (of "`openssl`" en "`version`").

Manueel installeren van OpenSSL
In principe is het altijd beter om de pakketten te gebruiken die voor je Linux-versie beschikbaar zijn. Als er echter geen pakket bestaat, kan je open-source software upgraden door ze zelf te builden op je machine. Voor OpenSSL voeren we volgende stappen uit:

1. Standaard zijn op onze server geen C compiler en andere build-tools geïnstalleerd. Deze moeten we dus eerst installeren:
   - `sudo apt-get install gcc`
   - `sudo apt-get install make`
2. We maken een tijdelijke directory in de temp-directory en downloaden de source code voor OpenSSL:
   - `cd /tmp ; mkdir openssl ; cd openssl`
   - `wget https://www.openssl.org/sources/openssl-1.0.1g.tar.gz`
3. We configureren de code door het config-script te runnen:
   `./config`
   Dit is nodig om de configuratie files te lezen, controleren van dependencies, includes en maken van een Makefile die belangrijke info bevat voor het installatieproces.
4. We compileren alle onderdelen van het pakket, en testen ze:
   - make

- make test

5. We installeren de onderdelen in de default directory (`/usr/local/ssl/`): `sudo make install`

6. We testen de versienummers:

   - van de origineel geïnstalleerde versie: `/usr/bin/openssl version -a`
   - van de nieuwe versie: `/usr/local/ssl/bin/openssl version -a`

7. We passen het PATH aan zodat naar de juiste locatie verwezen wordt voor OpenSSL. We voegden de volgende lijn toe in het begin van de PATH string "`/usr/local/ssl/bin:`". Vervolgens loggen we uit en testen we de versie met "`openssl version -a`". Hiervoor passen we de tweede "PATH"-lijn aan in `/etc/profile`, zodat deze "`/usr/local/ssl/bin`" bevat. Wanneer we, na in- en uitloggen nogmaals de versie opvragen met "`openssl version -a`", krijgen we te zien dat we openssl versie 1.0.1g draaien.

7.4 Samba

"Samba" is de uitspreekbare versie van `SMB` ("`Server Message Block`"), en is een open-source implementatie van het SMB-protocol dat in Windows gebruikt wordt om bestands-uitwisseling mogelijk te maken. In principe heeft Microsoft de specificaties van het protocol niet vrijgegeven, maar door *reverse engineering* is men toch in staat geweest om het voldoende te ontleden om er een implementatie van te maken die op andere dan Windows-systemen kan werken.
We zijn dus via de service Samba in staat om directories te delen met andere Linux, Windows en MAC-systemen.

7.4.1 Samba installeren

We installeren Samba via een root-terminal en volgend commando, `apt-get install samba`. We geven ACAD op als workgroup naam en kiezen om de WINS settings niet te importeren. Dit gebeurt er bij de installatie van Samba:

- voorconfigureren van pakketten
- uitpakken Samba
- instellen Samba
- config file voor Samba aanmaken
- accounts importeren
- accounts aan groep toevoegen

- daemons[3] starten (nmbd[4] en smbd[5])

We stellen alsook de VM-netwerkadapter in als Bridged adapter.

7.4.2 Samba starten en stoppen

In Linux worden services gestart en gestopt m.b.v. shell scripts, deze zijn te vinden in de `/etc/init.d` directory. Als we het Samba-script(`/etc/init.d/samba`) bekijken, zien we dat volgende acties kunnen worden uitgevoerd:

- start
- stop
- reload
- restart — force-reload
- status
- * (default)

Samba daemons Een deamon is een proces op Linux dat onafhankelijk in de achtergrond draait. Bij het opstarten wordt telkens `nmbd` (network samba-deamon) en vervolgens `smbd` (samba deamon) opgestart, bij het afsluiten worden ze in dezelfde volgorde afgesloten.

Runlevels Unix-systemen kunnen opereren op verschillende run-levels. Elke run-level beschrijft een configuratie van systeem-services die geactiveerd zijn. Bij het opstarten van het systeem, zal het in zijn "definitieve" runlevel gaan, en daar blijven tot het afgesloten wordt. In tabel 2 op de volgende pagina zien we de typische runlevels in Linux. In Debian linux wordt er geen onderscheidt gemaakt tussen runlevels 2 tot en met 5.

Welke services dat moet worden opgestart in elke runlevel wordt geconfigureerd d.m.v. symbolische links in specifieke directories: `/etc/rc?.d`, waarbij '?' staat voor het betreffende runlevel. De symbolische link verwijst naar de eigenlijke script die staan onder `/etc/init.d`.

Als we kijken naar het samba-script voor de verschillende runlevels dan zien we het volgende:

```
rc0 rc1 rc6      bevat    K19Samba
rc2 t.e.m. rc5   bevat    S20Samba
```

[3]Daemon is de term die binnen Unix gebruikt wordt om een proces aan te duiden dat op de achtergrond actief is om bepaalde (onderhouds)taken uit te voeren of om diensten te verlenen aan andere computerprogramma's.

[4]nmbd is een daeomon die alle naam registratie en resolution requests behandelt. Het is het primaire *vehicle* betrokken bij netwerk browsing. Het behandelt alle UDP-gebaseerde protocollen. Het nmbd daemon moet als eerste commando gestart worden bi het opstarten van Samba.

[5]De smbd daemon is voor meer geavanceerd networking waarbijer netwerk simulaties nodig zijn of servers in een guest moeten runnen.

Opmerking: Vermits in Debian geen onderscheidt wordt gemaakt tussen runlevels 2 t.e.m. zal de link dezelfde zijn.

De prefix K wijst duidt aan dat de service gestopt (Kill) moet worden en de S duidt aan dat de service gestart (Start) moet worden.
Samba wordt gestart bij runlevels 2 tot en met 5, en niet opgestart bij levels 0, 1 en 6.

ID	Name	Description
0	Halt	Shuts down the system.
1	Single-user Mode	Mode for administrative tasks.
2	Multi-user Mode	Does not configure network interfaces and does not export networks services.
3	Multi-user Mode with Networking	Starts the system normally.
4	Not used/User-definable	For special purposes.
5	X11 (with GUI)	Same as runlevel 3 + display manager.
6	Reboot	Reboots the system.

Tabel 2: Typische Linux runlevels

7.4.3 Samba configureren

1. Users toevoegen
   We selecteren onze user zodat we vanop een andere machine Samba share kunnen mounten. We dienen hiervoor een paswoord van deze user in te geven.

2. Directories configureren
   We maken onze lokale home-directory deelbaar en zorgen ervoor dat deze enkel met read-rechten kan gemount worden

3. Paswoord configureren
   We stellen een paswoord in met het commando sudo "`smbpasswd -a <username>`". Hierbij staat de `-a` voor de optie dat de username moet toegevoegd worden aan de lokale `smbpasswd`-file, samen met het getypte paswoord. Indien de username reeds aanwezig is in de file, wordt deze optie genegeerd.

4. Mounten op host
   We mounten de directory met volgende gegevens:

   Server: `<ip_address>`

   Share: `<user_name>`

   Username: `<username>`

   Domain name: `ACAD`

   Password: `**********`

   Indien we deze GUI niet verkregen konden we via het pad `smb://<user_name>@<PATH_to>` aan de juiste directory.

5. We kregen geen toegang tot het schrijven op de directory.

6. We passen in de `smb.conf` file de read permissie aan of we doen dit via de GUI (beiden in de VM).

7. Wanneer van een andere host, via het pad: `smb://<IP_address>/`, de home-directory proberen te mounten lukt dit niet, aangezien we het paswoord niet kennen. Dit lukt enkel wanneer we hun account en wachtwoord gebruiken.

7.5 Selectieve root-rechten

7.5.1 Gebruikers root-toegang geven

Via het `visudo` commando zijn we in staat om de `sudoers` file te bewerken en een gebruiker sudo-rechten te verlenen.
`<user_name> ALL =(ALL:ALL) ALL`
In ons geval: `jemergilles ALL=(ALL:ALL) ALL`
Nadere verklaring:

`demo` ALL=(ALL:ALL) ALL
Het eerste veld is de username waarvoor de regel zal gelden.

demo `ALL`=(ALL:ALL) ALL
De eerst "`ALL`" zegt dat de regel geldt voor alle hosts.

demo ALL=(`ALL`:ALL) ALL
Deze "`ALL`" zegt dat de 'demo' user alle commando's kan runnen als alle users.

demo ALL=(ALL:`ALL`) ALL
Deze "`ALL`" zegt dat de 'demo' user alle comando's kan runnen als alle groepen.

demo ALL=(ALL:ALL) `ALL`
De laatste "`ALL`" zegt dat deze regel geldt voor alle commando's.

Dit betekent dat onze 'demo' user alle commando's kan uitvoeren d.m.v. `sudo`, zolang het paswoord wordt opgegeven.

Via het ifconfig commando kunnen we dan testen of we root-toegang hebben verkregen:

```
ifconfig
        OUTPUT:
        command not found

sudo ifconfig:
        OUTPUT:
        We trust you have received the usual lecture from the
            local
        System Administrator. It usually boils down to
        these three things:

        #1) Respect the privacy of others.
        #2) Think before you type.
        #3) With great power comes great responsibility.
```

7.5.2 Groep root-toegang geven

Als je meerdere beheerders hebt op je systeem, is het niet altijd handig om voor iedereen individueel de sudo-rechten te beheren. Daarom kan je ook via het ingebouwde systeem van groups werken. We kunnen onszelf toevoegen aan de sudo groep. Dit kan door het bestand `/etc/group` als root te editeren. We editoren de sudoers file om de sudo groep root rechten te geven.

In de file `/etc/group/` zoeken we naar de regel die met sudo begint en voegen onze gebruikersnaam toe.

```
sudo:x:27: jelmergilles
```

We geven de sudo-groep root rechten door in het bestand "`%sudo ALL=(ALL:ALL) ALL`" toe te voegen.
Betekenis '`%`': elke member van de groep krijgt deze rechten.

7.5.3 Gebruiker selectieve root-rechten geven

`ifconfig` is eigenlijk al door iedereen uitvoerbaar, maar de locatie van `ifconfig` is niet in het `PATH` van de gewone gebruiker opgenomen, waardoor het niet uit te voeren is. Wanneer we de locatie van `ifconfig` opzoeken met "`whereis ifconfig`" en deze locatie (`/sbin`) toevoegen aan het PATH, lukt het wel om als gewone gebruiker ifconfig uit te voeren. Een snelle manier om het toe te voegen aan het PATH is met het commando "`export PATH=$PATH:/sbin`". Dit appendeert `/sbin` aan het PATH van de gebruiker.

We kunnen ook het volgende doen:
In de `sudoers`-file de volgende lijn toevoegen:
`<user_name> ALL=(ALL) NOPASSWD: /sbin/ifconfig`
met `/sbin/ifconfig` het pad naar het commando.

8 Netwerken - Basis (Labo 8, 9 en 12)

We gaan de basis-netwerkinfrastructuur van Linux bekijken. Aan de hand van terminalcommando's gaan we de netwerkinterfaces van ons virtuele systeem inspecteren, en onderzoeken welke programma's er netwerkconnecties geopend hebben, of luisteren naar connecties via sockets. Hierna bespreken we hubs en switches die zorgen voor het verstuuren en ontvangen van ethernet pakketten.

8.1 Communicatie technologieën

8.1.1 Packet switching

Hierbij gaat men de data die men moet overbrengen onderverdelen in kleinere pakketten. Eens dit is gebeurd, zullen de pakketten individueel verstuurd worden. Hierdoor kan elk pakket een andere route nemen en zo op verschillende tijdstippen bij de ontvanger aankomen. Bij de ontvanger worden alle pakketten terug samengevoegd als 1 geheel datablok.

- ☺ Geen vaste routes en dus minder afhankelijk van problemen
- ☺ Capaciteit kan toegekend worden op basis van nood
- ☹ Kwaliteitsgarantie is moeilijker
- ☹ Opsplitsing: volgorde van aankomst?

8.1.2 Circuit switching

Hierbij gaat men een langdurige connectie opzetten tussen zender en ontvanger. Men gaat hiervoor bij elke connectie een vast percentage van de netwerk-capaciteit geven.

- ☺ Kwaliteitsgarantie mogelijk
- ☺ Transfers gegarandeerd na opzetten verbinding
- ☹ Netwerk-capaciteit ongebruikt bij "stiltes"
- ☹ Vaste datasnelheden
- ☹ Niet efficiënt voor verschillende types data

8.2 Het internet

8.2.1 Het OSI model

OSI Model			
	DATA UNIT	LAYER	FUNCTION
Host layer	data	Application	Network process to application
		Presentation	Data representation, encryption and decryption, convert machine dependent data to machine independent data
		Session	Interhost communication
	segment	Transport	End-to-end connections and reliability, flow control
Media layer	Packet/Diagram	Network	"routen" tussen lokale netwerken, logische adressering,...
	Frame	Data link	Error checking, pakketten afleveren in lokaal netwerk
	Bit	Physical	Media, signal and binary transmission

Figuur 7: OSI model

8.2.2 Netwerk laag

Een lokaal netwerk bestaat uit verschillende apparaten die verbonden zijn met elkaar. Elk apparaat heeft een zijn eigen netwerkinterface. Bij grote netwerken gaat men meerdere kleinere netwerken met elkaar verbinden. Hiervoor gaat men verbindingsapparaten (vb. Switch, hub en router) gebruiken die meerdere netwerk-interfaces bevatten.

8.2.3 Transport laag (TCP - UDP)

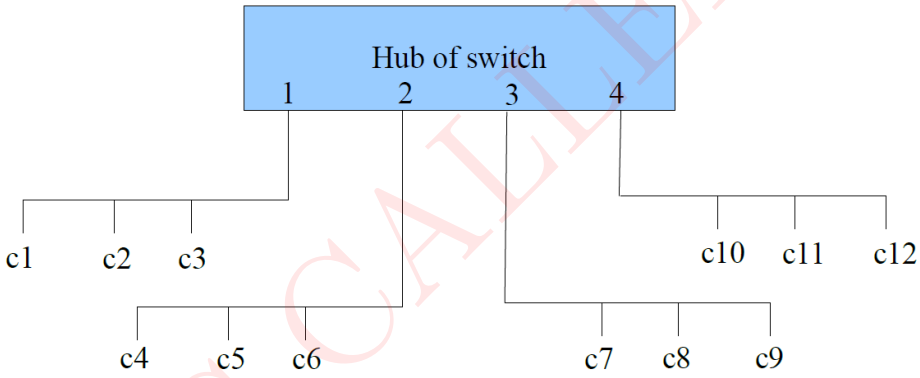
Dankzij poorten kunnen we met 1 netwerk-interface meerdere connecties aanmaken. Hierbij gaan we multiplexen van meerdere TCP-connecties over dezelfde hardware-connectie. Er zijn voor gedefinieerde poorten (vb. http via poort 80 of SSH via poort 22) en er zijn poorten die vrij zijn voor programma's die draaien op een machine.

	Betrouwbaar	Pakket volgorde	Overhead
UDP	Geen garantie	Geen link tussen paketten	Weinig
TCP	Gegarandeerde aankomst	Aankomst in dezelfde volgorde al vertrek	Veel (opzetten connectie, garanties, volgorde)

Figuur 8: Transportlaag protocollen

8.3 Verbindingsapparaten

Om een lokaal netwerk (ethernet) uit te bouwen kunnen we hubs en switches gebruiken. Ethernet is een packet-switched netwerksysteem, waarin elk apparaat een MAC-adres heeft (MAC = media access control); dit is bij de productie van de netwerkadapter geconfigureerd door de fabrikant. Data wordt verstuurd onder de vorm van pakketten (packets), en elk pakketje wordt voorzien van een afzend-adres, een ontvanger, en een datablok. Een afzender zet een ethernetpakket (frame) op de kabel, en alle andere devices die op dat netwerksegment zitten, ontvangen het pakket. Aan de hand van het ontvanger-adres, kunnen ze zien of het pakket voor hen bestemd is of niet. Hubs en switches zijn gelijkaardig, maar



Figuur 9: Gekoppelde netwerksegmenten d.m.v. een hub of switch

daar waar hubs domweg binnenkomende pakketten kopiëren naar alle andere netwerksegmenten, bevatten switches hardware en software die binnenkomende pakketten onder de loupe kan nemen, en kan beslissen naar welke netwerksegment (of netwerksegmenten) een pakket moet gekopieerd worden. (Terminologie: een netwerksegment is aangesloten aan een poort op de hub of switch.)

8.3.1 Switch

```java
/**
 * Created by Gilles on 24-4-2015.
 */
public class Test {
    public static void main(String args[]){
        Switch switcher = new Switch();

        switcher.forwarding(1,new Packet("c1","c4",5,"data"));
        switcher.forwarding(4,new Packet("c12","c1",5,"data"));
        switcher.forwarding(2,new Packet("c4","c7",5,"data"));
        switcher.forwarding(4,new Packet("c10","c4",5,"data"));
    }
}
```

```java
/**
 * Created by Gilles on 24-4-2015.
 */
public class Packet {
    private String senderMACAddress;
    private String receiverMACAddress;
    private int lengthBytes;
    private String data;

    public Packet(String senderMACAddress, String receiverMACAddress,
        int lengthBytes, String data) {
        this.senderMACAddress = senderMACAddress;
        this.receiverMACAddress = receiverMACAddress;
        this.lengthBytes = lengthBytes;
        this.data = data;
    }

    public String getSenderMACAddress() {
        return senderMACAddress;
    }

    /* non switch getters */

    public String getReceiverMACAddress() {
        return receiverMACAddress;
    }

    public String getData() {
        return data;
    }

    public int getLengthBytes() {
        return lengthBytes;
    }
}
```

```java
import java.util.HashMap;

/**
 * Created by Gilles on 24-4-2015.
 */
public class Switch {
        private HashMap<String,Integer> ports;

        public Switch() {
                ports = new HashMap<String,Integer>();
        }

        public void forwarding(Integer incomingPortNumber, Packet packet){
                String senderMacAddress = packet.getSenderMACAddress();
                saveSenderPortNumber(incomingPortNumber,senderMacAddress);
                sendPacketToReceiver(incomingPortNumber, packet);
        }

        private void saveSenderPortNumber(Integer incomingPortNumber, String
            senderMacAddress) {
                if (!ports.containsKey(senderMacAddress)){
                        ports.put(senderMacAddress, incomingPortNumber);
                }else if(ports.get(senderMacAddress).toString().equals(
                    incomingPortNumber.toString())){
                // overschrijven sender zit op een nieuwe port
                        ports.remove(senderMacAddress);
                        ports.put(senderMacAddress, incomingPortNumber);
                }
        }

        private void sendPacketToReceiver(Integer incomingPortNumber, Packet packet) {
                String receiverMacAddress = packet.getReceiverMACAddress();
                if(ports.containsKey(receiverMacAddress)){
                        Integer sendingPort = ports.get(receiverMacAddress);
                        if(incomingPortNumber == sendingPort){
                        // niets doen, pakket blijft in het versturende netwerksegment
                        } else {
                                System.out.println("Pakket wordt verzonden op poort:" +
                                    sendingPort);
                        }
                }else{
                        floodPacket(packet);
                }
        }

        private void floodPacket(Packet packet) {
                System.out.println("Pakket wordt geflood naar alle poorten");
        }

}
```

We kunnen de tijd om alle poort-MAC mappings te vinden, niet op voorhand inschatten vermits we enkel poortnummers kunnen weten van versturende nodes. De snelheid naar

convergentie hangt dus af van de versturende nodes. Vermits er op een netwerk iedereen een pakket verstuurd zullen alle nodes (in mapping met hun poortnummer) snel gekend zijn.

8.3.2 Hubs

Stel node `c2` wil een reeks pakketten sturen naar node `c8`.
In het geval van een hub worden de pakketten verstuurd naar alle netwerkpoorten anders dan degene waar het pakket binnenkomt. In het voorbeeld komt het pakket binnen op poort 1, dus wordt het gekopieerd naar poorten 2, 3 en 4. Dit is niet zo efficiënt, aangezien de ontvangende computer op slechts één van de andere poorten kan aangesloten zijn, en alle andere netwerksegmenten dus een pakket krijgen dat sowieso voor geen enkel device op dit segment bestemd is. Dit beperkt de beschikbare bandbreedte op deze netwerksegmenten, ook voor pakketten waarvan de afzender en de ontvanger zich op hetzelfde netwerksegment bevinden.

8.3.3 Router

Een router heeft zelf netwerkadres en kan pakketten ontvangen en gericht doorsturen naar een subnet. Een router bevat:

- DHCP server (interne adressen: typisch 192.168.?.?)
- NAT (=Network Address Translation)
- naar buiten toe: alle devices gebruiken IP-adres van provider
- Router ontvangt alles en verdeelt binnen thuisnetwerk

IP-adres ontvangen Er zijn 2 mogelijkheden om een IP-adres te ontvangen:

- Dit kan handmatig gebeuren en is statisch.
- Dit kan ook via DHCP (Dynamic Host Configuration Protocol). Hierbij vraagt de PC een IP-adres aan de DHCP-server (dit is dikwijls de router). Het adres is geldig voor bepaalde tijd, deze moet daarna opnieuw aangevraagd worden.

8.4 Netwerkinterfaces

8.4.1 Status van netwerkinterface(s)

Via het commando `ifconfig` zien we alle netwerkinterfaces. In ons geval krijgen we 2 connectie te zien waarbij er één de ethernet-adapter is (`eth0 [10.128.48.48]`) en de andere ons eigen loopback adres (`127.0.0.1`) is, deze verwijst naar ons eigen systeem.

8.4.2 Netwerkinterfaces stoppen en starten

We kunnen netwerkinterfaces starten en stoppen door de commando's `ifup` en `ifdown` respectievelijk. Als we het commando `ifdown eth0` ingeven hebben we geen internetconnectie, als we `ifup eth0`, starten we opnieuw de interface en hebben we wel toegang tot het internet.

8.5 TCP/IP tools

TCP laat toe om meerdere soorten netwerkconnecties tegelijkertijd te maken door aan de verschillende diensten een poortnummer toe te kennen.

8.5.1 Port scans

`nmap` is een tool die je kan gebruiken om port scans te doen. Dit betekent dat `nmap`, voor een gegeven server, alle poorten zal testen, en kijken achter dewelke een dienst actief is.

Vanop host-machine naar VM
Commando: `nmap 10.128.48.48`

```
PORT            STATE           SERVICE
22/tcp          open            ssh
111/tcp         open            rpcbind
139/tcp         open            netbios-ssn
445/tcp         open            microsoft-ds
```

Vanop VM naar eigen systeem (localhost)
Commando: `nmap 127.0.0.1`

```
PORT            STATE           SERVICE
22/tcp          open            ssh
25/tcp          open            smtp
111/tcp         open            rpcbind
139/tcp         open            netbios-ssn
445/tcp         open            microsoft-ds
631/tcp         open            ipp
```

De diensten smtp en ipp worden niet weergegeven buiten het systeem.

8.5.2 Sockets

Server/netwerkdiensten gebruiken een bepaalde poort om hun diensten aan te bieden. Op softwareniveau doen ze dat door sockets te openen voor de betreffende poort, en te "luisteren" of er binnenkomende connecties zijn. Op Unix systemen bestaat er ook zoiets als "lokale" sockets. Deze hebben niet rechtstreeks te maken met het netwerk waarmee je computer verbonden is. Unix gebruikt sockets ook voor communicatie tussen verschillende processen op het systeem.

netstat
`netstat` is een tool waarmee je kan kijken welke sockets er bestaan en welke applicaties ze geopend hebben.
Bekijken van alle tcp sockets:

```
netstat -p | grep ''tcp''
```

Optie `-p`: Toon de PID (*process identifier*) en naam van het programma waarvan de socket behoort.

Starten firefox:
Als we het commando van hierboven opnieuw uitvoeren zien we meer `tcp` connecties die

telkens `ESTABLISHED` zijn.

Starten ssh-sessie:
Wanneer we een ssh-sessie starten en het commando van hierboven uitvoeren verkrijgen we een extra tcp socket.

8.6 De service-file

Veel standaard-services hebben een vast poortnummer. Deze staan opgelijst in de services file, deze wordt gebruikt om nummers naar namen te mappen.
Bestandspad: `/etc/services`
De locatie van de service-file wordt gedefinieerd door `_PATH_SERVICES` in `<netdb.h>`.

8.6.1 Poortnummers

```
HTTPS:          443/tcp of udp
SMTP:           25/tcp
IMAP3:          220/tcp of udp
POP3:           110/tcp of udp
```

8.7 Ethernet tools

Wanneer we hebt programma `ethstats` hebben geïnstalleerd op de virtuele machine, krijgen we een live feed van de ethernetverbindingen. De ethernet statistieken worden om elke 10 seconden getoond voor elke netwerk interface. In de terminal zien we de gegevens van het verbruik bij up- en download, deze worden weergegeven in Mb/s en packets/s.

8.8 Wireshark

Wireshark is een netwerktool die je kan gebruiken om de pakketten die toekomen op je PC, op het niveau van de netwerkkaart, te bekijken. Alle pakketten worden zichtbaar gemaakt, en kunnen bestudeerd worden. Op die manier kan je, op zeer laag niveau, bekijken welke pakketten er toekomen en verstuurd worden. Je kan het netwerkverkeer in real-time bekijken, er filters op toepassen om trafiek met specifieke kenmerken nader te bestuderen, en je kan logfiles bewaren om ze later te bestuderen.

8.8.1 Protocollen

We zien enkele protocollen verschijnen wanneer we de pakketten bekijken die we ontvangen of versturen.

```
TCP             Transfer Control Protocol
HTTP            HyperText Transport Protocol
DNS             Domain Name Systems/Servers
ARP             Adddress Resolution Protocol
OCSP            Online Certificate Status Protocol
TLS             Transport Layer Security
SSL             Secure Sockets Layer
```

8.8.2 Filteren

Als we `google.com` raadplegen zien we dat we via DNS het IP-adres van `google.com` opzoeken, en dan een TCP connectie opstarten (wat gepaard gaat met enkele TCP gerelateerde pakketten) waarbij er meerdere http requests worden verstuurd.

Wanneer we filteren op TCP of UDP pakketten die toekomen op poort 80, kunnen we het IP-adres van `google.com` vinden (`216.58.210.67`), zien dat er een tiental GET requests worden verstuurd en dit op poorten hoger dan 1024 (vb. 44144, 44145,...).

Als we filteren op ssh (via `ssh` of `tcp.port ==22` filter) en een ssh-connectie opstarten met onze VM dan zien we een reeks van SSH-pakketten die telkens geëncrypteerd zijn.

9 Firewall (Labo 10)

Firewalls zijn een essentieel onderdeel van computer-security. Ze dienen om pakketjes die via het netwerk aankomen op een computer of netwerksegment te bekijken, en enkel degene door te laten doe aan bepaalde regels voldoen.

Firewalls kunnen (dikwijls) ook aanpassingen uitvoeren aan pakketjes. Dit is nuttig bij de overgang van privaat netwerk naar het internet, waarbij er één toegangspunt is (de router). Voor de servers op het internet lijkt het dan alsof alle interne connecties afkomstig zijn van één IP-adres.

Firewalls komen voor op alle niveaus in een netwerk: op individuele PCs, als onderdeel van routers, of om grotere netwerken met elkaar te verbinden.

9.1 IP tables

De firewall-software die bij de meeste Linux-distributies geleverd wordt, heet "`iptables`". hierin worden regels gedefinieerd om pakketten te matchen, en daaraan acties gekoppeld die uitgevoerd worden.

9.1.1 Regels oplijsten

Commando:

```
iptables -L
```

Er worden 3 verschillende chains weergegeven: `INPUT`, `FORWARD` en `OUTPUT`. Voor elke chain is de default policy `ACCEPT`.

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

9.1.2 Default policies

Als we de policy voor de `INPUT` chain willen veranderen naar `DROP` gebruiken we volgend commando:

```
iptables --policy INPUT DROP
```

of

```
iptables -P INPUT DROP
```

We controleren of de `INPUT` chain inderdaad de pakketten dropt, door een ssh-connectie te proberen starten. De firewall accepteert de pakketten niet en de pakketten worden gedropt zonder ze zelf maar te bekijken, de ssh-connectie lukt dus niet.

9.1.3 Speficieke regels

SSH access blokkeren we met commando:

```
iptables -A INPUT -p tcp --dport 22 -j DROP
```

Opties: `-A` (append) `-p` (protocol) `--dport` (destination port) `-j` (target)

9.1.4 Regels bewaren

Opslaan van de iptables naar een bestand via:

```
iptables-save > iptables.txt
```

Als we nu echter de VM opnieuw opstarten zien we dat de regels niet zijn ingeladen. Regels opnieuw inladen van het bestand via:

```
iptables-restore < iptables.txt
```

of

```
iptables.txt|iptables-restore
```

9.1.5 Regels inladen bij het opstarten

We maken een script in de directory `/etc/network/if-pre-up.d/`. De scripts in deze directory zullen worden uitgevoerd telkens als de netwerkinterfaces opgestart worden. We vergeten niet het scriptje executable te zetten!
Opmerking: We laten `.sh` extensie weg vermits scripts die automatisch worden opgestart geen tekens buiten ASCII lower- en uppercase mogen hebben, er mogen dus geen punten staan in de file.

Script, firewall:

```
#!/bin/bash
sudo iptables-restore < /home/jelmergilles/iptables
```

10 Herhalingsoefeningen (Labo 11)

10.1 Comprimeren en opslaan in één bestand

Via de utilities `tar` en `gzip` kunnen we bestanden respectievelijk opslaan in één bestand en comprimeren. Om onze home directory in één gecomprimeerd bestand te krijgen maken we gebruik van het volgende commando:

```
tar -zcvf home.tar.gz /home/
```

Voor het unzippen van het bestand voeren we het volgende commando uit: `tar -zxvf home.tar.gz`.

10.2 Crontab

Via het commando "`contrab -e`" kunnen we de crontab aanpassen. Om het resultaat van de vorige opdracht uit te voeren om de 5 minuten, voegen we volgende regel toe aan de crontab:

```
*/5 * * * * /home/student/tar_script.sh
```

`tar_script.sh` bevat volgende lijnen:

```
#!/bin/bash
tar -zcvf /tmp/home.tar.gz /home/
```

Als we het script willen laten uitvoeren elke 5 minuten (overdag) dan moeten we dit commando invoeren in de contrab:

```
*/5 7-22 * * * /home/student/tar_script.sh
```

Het script zal dus uitgevoerd worden elke 5 minuten tussen 7u en 22:59u.

10.3 Secure CoPy (scp)

Telkens als we een SSH connectie wouden aanmaken met onze virtuele machine, moesten we ons wachtwoord ingeven om een keypair te generen. We maakten dus bij elke connectie een nieuwe keypair aan. Indien we manueel een keypair generen, blijft deze keypair waardoor we niet meer ons wachtwoord moeten ingeven. We zorgen er dus eerst voor dat de host-pc via SSH kan bereikt worden zonder een paswoord in te geven met de commando's: "`ssh-keygen`" en "`ssh-copy-id student@10.128.58.4`". We passen het script aan, en voegen de volgende lijn toe:

```
scp /tmp/home.tar.gz student@10.128.58.4:home.tar.gz
```

Hierdoor zal het backup-bestand worden gekopieerd van onze VM naar de host PC.

10.4 Kopies met timestamp

We maken nu elke 5 minuten een backup, maar deze overschrijft telkens de vorige. We passen het systeem aan zodat er elk uur ook een backup-bestand gemaakt wordt, waarvan de bestandsnaam telkens een timestamp bevat. Script:

```
#!/bin/bash
        file_name=/tmp/home.tar.gz
        current_time=$(date "+%Y.%m.%d-%H.%M.%S")
        new_fileName=/tmp/.$current_time.tar.gz
        cp $file_name $new_fileName
```

Aanvulling in crontab:

```
0 7-22 * * * /home/student/tar_timestamp_script.sh
```

11 Veelgebruikte commando's

11.1 `ls`

`ls` is het commando om bestanden of directories op te lijsten. Enkele opties voor ls zijn:

`-l`
geeft een lange weergave van alle bestanden en directories in de huidige directory. In deze lange weergave zitten bijvoorbeeld de rechten.

`-a`
geeft alle bestanden en directories in de huidige directory weer, dus ook de hidden files

`-g`
doet hetzelfde als `-l`; maar toont de eigenaars niet

`-h`
kan enkel uitgevoerd worden in combinatie met -l, en zorgt ervoor dat de grootte van bestanden en directories in voor mensen leesbare vorm wordt weergegeven

`i, --inode`
with -l, print the index number of each file

Via het commando `ls -shR` kunnen we recursief alle bestanden tonen in 'human readable format' en met de grootte erbij.

`-s, --size`
Geef de grote van elke file weer in blocks

`-h, --human-readable`
Toon de grote weer in human readable format (e.g., 1K 234M 2G)

`-R, --recursive`
Lijst alle subdirectories recursief op

11.2 `cp`

`cp` is het commando om bestanden of directories te kopiëren. Enkele opties voor cp zijn:

`-R`
kopiëren van directories op een recursieve manier

`-u`
het te kopiëren bestand wordt enkel gekopieerd als het nieuwer is dan een eventueel al aanwezig bestand met dezelfde naam

`-v`
tijdens het kopiëren wordt uitgelegd wat gedaan wordt.

`-s`
in plaats van het bestand te kopiëren wordt een symbolische link gelegd naar het bestand.

`-l`
in plaats van het bestand te kopiëren wordt een harde link gelegd naar het bestand.

11.3 rm

`rm` is het commando om bestanden of directories te verwijderen. Enkele opties voor rm zijn:

`-f`
het verwijderen van de bestanden of directories wordt geforceerd.

`-d`
alle lege directories in de huidige directory worden verwijderd.

`-i`
voor het verwijderen van elk bestand wordt een bevestiging gevraagd.

`-r`
recursief verwijderen van een map. Hierbij wordt ook alle inhoud van submappen verwijderd.

11.4 whoami

Via het commando `whoami` kunnen we de huidige gebruiksnaam opvragen.

11.5 finger

We kunnen info weergeven van *system users* die recentelijk zijn ingelogd geweest op de terminal. Dit doen we via het commando `finger`.

11.6 More vs less

`more <FILE_NAME>` opent het bestand in de commandline, onder het ingegeven commando. `less <FILE_NAME>` zal het bestand in een leeg terminalvenster openen. In `less` is het mogelijk om terug naar boven te scrollen in het bestand, terwijl bij more alleen wordt toegelaten om voorwaarts te gaan. `less` leest ook niet heel de inputfile meteen bij het openen, hierdoor zal hij dus sneller openen.

11.7 tail

Het `tail <FILE_NAME>` commando zal de laatste (default) 10 lijnen weergegeven (uitprinten naar standaard output) van het bestand.

`-f --follow`
zal toegevoegde data tonen (als de file groeit)

`-n --lines=K`
tonen van de K laatste lijnen

11.8 df

Het `df` commando zal het fle system disk space usage weergeven.

`-B --block-size = SIZE`
Schaalt groottes naar een gegeven size format

`-k`
Zal gebruik maken van size format = 1K

`-h --human-readable`
tonen van groottes in machten van 1024

11.9 tar

11.9.1 Toevoegen aan bestand en comprimeren

```
tar -zcvf workspace.tar.gz workspace
```

`-z` compress via gzip programma

`-c` create archive

`-v` verbose (toon progress)

`-f`(doelnaam specifiëren aan de tarbal)

11.9.2 Bestand decomprimeren en uitpakken

```
tar -zxvf home.tar.gz
```

`-x` extract

11.10 touch

Het `touch` commando is de eenvoudigste manier om een nieuw leeg bestand aan te maken.

11.11 grep

Grep searches the named input FILEs (or standard input if no files are named, or the file name - is given) for lines containing a match to the given PATTERN. By default, grep prints the matching lines.

`-w, --word-regexp`
Select only those lines containing matches that form whole words. The test is that the matching substring must either be at the beginning of the line, or preceded by a non-word con- stituent character. Similarly, it must be either at the end of the line or followed by a non-word constituent character. Word- constituent characters are letters, digits, and the underscore.

`-o` printen van match, niet heel de lijn tonen

`-i, --ignore-case`
Ignore case distinctions in both the PATTERN and the input files.

11.12 wget

Via het `wget` commando kunnen files over HTTP, HTTPS en FTP downloaden.

11.13 visudo

Het `visudo` commando laat ons toe om de `sudoers`[6] file te bewerken.

11.14 ifconfig

`ifconfig` staat voor "*interface configuration*". Het wordt gebruikt om de configuratie van de netwerkinterfaces van het systeem te bewerken en weer te geven.

`-a`
Door de `-a` optie zien we alle netwerkinterfaces, niet enkele degenen die actief zijn.

`ifconfig <device> up`
We activeren het netwerkinterface `<device>`.

`ifconfig <device> down`
We stoppen het netwerkinterface `<device>`.

We kunnen ook gebruik maken van volgende commando's: `ifdown eth0` en `ifup eth0`.

11.15 nmap

`nmap` is een tool die je kan gebruiken om port scans te doen. Dit betekent dat `nmap`, voor een gegeven server, alle poorten zal testen, en kijken achter dewelke een dienst actief is.

11.16 cat

`Cat` zal de content van een file uitprinten naar de standaard output of kan ook gebruikt worden om lijnen tekst toe te voegen aan een bestand.
Voorbeelden:

```
cat /etc/passwd
```

```
cat /tmp/dates.txt $var
```

11.17 du

`du` zal de ingenomen file ruimte schatten.

`-a, --all`
write counts for all files, not just directories

`-h, --human-readable`
print sizes in human readable format (e.g., 1K 234M 2G)

Voorbeeld: Via de commando-lijn '`du -ah /boot/vmlinuz*`' komen we te weten hoe groot de kernel is.

[6] De sudoers policy bepaalt de gebruikers sudo rechten

11.18 `find`

Via `find` kunnen we bestanden zoeken in een directory hiërarchie. Enkele opties voor `find`:

- -type c
  File is of type c:

  b block (buffered) special

  c character (unbuffered) special

  d directory

  p named pipe (FIFO)

  f regular file

  l symbolic link; this is never true if the -L option or the -follow option is in effect, unless the symbolic link is broken. If you want to search for symbolic links when -L is in effect, use -xtype.

  s socket

  D door (Solaris)

11.19 `chmod`

Via het commando `chmod` is het mogelijk om toegangsrechten te wijzigen. Dit kunnen we op 2 manieren:

- Octale-methode
  `chmod [0-7] [0-7] [0-7] <file_name>`
  Met waarden r=4, w = 2 en x=1.

- String-methode
  `chmod [ugoa] +,-,= [rwx] <file_name>`
  + toevoegen, - wegnemen, = gelijkstellen van permities

11.20 `ln`

Via het commando `ln` kunnen we links maken tussen bestanden.

11.20.1 Symbolische links

Via het commando `ln -s <src_file> <soft_copy_file>` maken we een symbolische koppeling van het eerste bestand naar het tweede.

11.20.2 Harde links

Harde link aanmaken: `ln <src_file> <hard_copy_file>`

11.21 awk

awk is een patroon scanning en text processing language.
Voorbeeld:
`ps aux | awk '{print $1 "\t-> \t"$11}'`
Waarbij &1 en &11 respectievelijk de kolom van de gebruiker en procesnaam is.

11.22 uname

uname zal bepaalde systeem informatie afprinten.

`-a, --all`
print all information

Voorbeeld output:

```
Linux gillesubuntu-Inspiron-1750 3.19.0-21-generic
#21-Ubuntu SMP Sun Jun 14 18:31:11 UTC 2015 x86_64 x86_64
   x86_64 GNU/Linux

kernel-name:
        Linux
network node hostname:
        gillesubuntu-Inspiron-1750
kernel-release:
        3.19.0-21-generic
kernel-version:
        #21-Ubuntu SMP Sun Jun 14 18:31:11 UTC 2015
machine hardware name:
        x86_64
processor type:
        x86_64
hardware platform:
        x86_64
operating-system:
       GNU/Linux
```

11.23 smbpasswd

Veranderen van een gebruikers SMB paswoord.

`-a`
Hierbij staat de -a voor de optie dat de username moet toegevoegd worden aan de lokale smbpasswd-file, samen met het getypte paswoord. Indien de username reeds aanwezig is in de file, wordt deze optie genegeerd.

Gebruik: `sudo smbpasswd -a <username>`

11.24 scp

scp, secure copy is een remote file copy program. scp kopieert bestanden tussen hosts in een netwerk. Het gebruik ssh voor zijn data transfer, en gebruikt dezelfde authenticatie en beveiliging als ssh.

When copying a source file to a target file which already exists, scp will replace the contents of the target file (keeping the inode).
Gebruik: `scp [-1246BCpqrv] [-c cipher] [-F ssh_config] [-i identity_file] [-l limit] [-o ssh_option] [-P port] [-S program] [[user@]host1:]file1 [...] [[user@]host2:]file2`

Voorbeeld: `scp passstudent.txt jelmergilles@10.128.48.44:/home/jelmergilles/Desktop`

11.25 sftp

sftp (secure file transfer program) is an interactive file transfer program, similar to ftp, which performs all operations over an encrypted ssh transport. It may also use many features of ssh, such as public key authentication and compression. sftp connects and logs into the specified host, then enters an interactive command mode.

Gebruik:

```
sftp [[user@]host[:file [file]]]

sftp [[user@]host[:dir[/]]]
```

Voorbeeld: `sftp jelmergilles@10.128.48.44 put oudelijst.txt`

11.26 nmap

`nmap - Network exploration tool and security / port scanner`

nmap is een tool die je kan gebruiken om port scans te doen. Dit betekent dat nmap, voor een gegeven server, alle poorten zal testen, en kijken achter dewelke een dienst actief is.
Gebruik: `nmap [Scan Type...] [Options] target specification`
Voorbeeld: `nmap scanme.nmap.org`

11.27 gcc

`gcc - GNU project C and C++ compiler`

11.28 make

`make - GNU make utility to maintain groups of programs`

The purpose of the make utility is to determine automatically which pieces of a large program need to be recompiled, and issue the commands to recompile them.

11.29 iptables

`iptables - administration tool for IPv4 packet filtering and NAT`

Iptables is used to set up, maintain, and inspect the tables of IP packet filter rules in the Linux kernel. Several different tables may be defined. Each table contains a number of built-in chains and may also contain user-defined chains. Each chain is a list of rules which can match a set of packets. Each rule specifies what to do with a packet that matches. This is called a 'target', which may be a jump to a user-defined chain in the same table.
Opties:

`-L, --list [chain]`
List all rules in the selected chain. If no chain is selected, all chains are listed.

`-A, --append chain rule-specification`
Append one or more rules to the end of the selected chain.

`-P, --policy chain target`
Set the policy for the chain to the given target.

`-p, --protocol [!]  protocol`
The protocol of the rule or of the packet to check.

`-j, --jump target`
This specifies the target of the rule; i.e., what to do if the packet matches it.

`-dport`
This specifies the destination port.

Andere iptables gerelateerde commando's:

```
iptables-save

iptables restore
```

Voorbeelden:

```
iptables -P INPUT DROP

iptables -A INPUT -p tcp --dport 22 -j DROP

iptables-save > iptables.txt

iptables-restore < iptables.txt
```

11.30 whereis

`whereis` locates source/binary and manuals sections for specified files.

11.31 netstat

`netstat` - Print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships

textttnetstat is een tool waarmee je kan kijken welke sockets er bestaan en welke applicaties ze geopend hebben.

`-p, --program`
Show the PID and name of the program to which each socket belongs.

Voorbeeld: Bekijken van alle tcp sockets via het commando
`netstat -p | grep ''tcp''`

11.31.1 netstat states

ESTABLISHED	The socket has an established connection.
SYN_SENT	The socket is actively attempting to establish a connection.
SYN_RECV	A connection request has been received from the network.
FIN_WAIT1	The socket is closed, and the connection is shutting down.
FIN_WAIT2	Connection is closed, and the socket is waiting for a shutdown from the remote end.
TIME_WAIT	The socket is waiting after close to handle packets still in the network.
CLOSE	The socket is not being used.
CLOSE_WAIT	The remote end has shut down, waiting for the socket to close.
LAST_ACK	The remote end has shut down, and the socket is closed. Waiting for acknowledgement.
LISTEN	The socket is listening for incoming connections. Such sockets are ot included in the output unless you specify the –listening (-l) or –all (-a) option.
CLOSING	Both sockets are shut down but we still don't have all our data sent.
UNKNOWN	The state of the socket is unknown.

11.32 ethstats

We verkrijgen een live feed van de ethernetverbindingen. De ethernet statistieken worden om elke 10 seconden getoond voor elke netwerk interface. In de terminal zien we de gegevens van het verbruik bij up- en download, deze worden weergegeven in Mb/s en packets/s.

11.33 mv

`mv - move (rename) files`

Voorbeeld verplaatsen:

```
mv myfile.txt destination-directory
```

Voorbeeld rename file:

```
mv myfile.txt myfile_renamed.txt
```

11.34 het ps commando

Via het `ps` commando kunnen we alle uitvoerende processen bekijken.

`ps -A`	alle actieve processen
`ps -U <login_name>`	eigen processen
`ps aux`	alle processen
`ps -af`	hiërarchie van processen

11.35 mkdir

Via dit commando kunnen we een map aanmaken. Enkele opties voor mkdir zijn:

`-p` : zorgt ervoor dat alle mappen die naar die directory wijzen ook aangemaakt worden.

`-v` : tonen van alle directories dat mkdir heeft aangemaakt

`-m` : specifiëren van de octal permissions4 van de directory die door mkdir

Voorbeeld:

```
mkdir -p sysnetbeheer/labo
```

Via dit commando willen we de map labo aanmaken. `-p` zal ervoor zorgen dat de map sysnetbeheer ook wordt aangemaakt indien deze nog niet bestond.

11.36 pwd

Dit commando geeft de huidige directory weer.

12 Belangrijke files

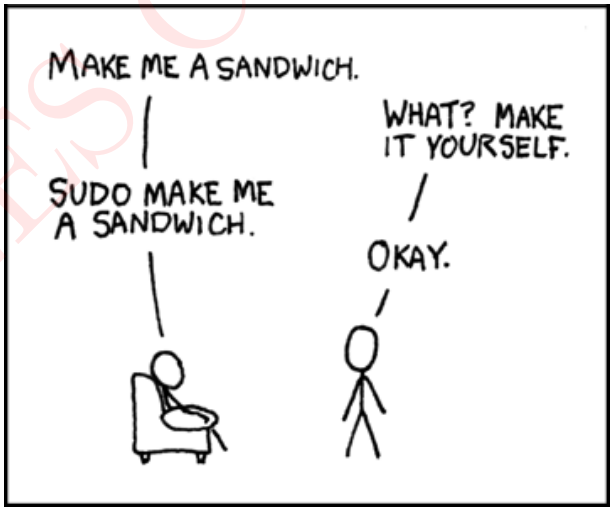
12.1 sudoers

12.1.1 Gebruikers root-toegang geven

Via het `visudo` commando zijn we in staat om de `sudoers` file te bewerken en een gebruiker sudo-rechten te verlenen.
`<user_name> ALL =(ALL:ALL) ALL`
In ons geval: `jemergilles ALL=(ALL:ALL) ALL`
Nadere verklaring:

> `demo ALL=(ALL:ALL) ALL`
> Het eerste veld is de username waarvoor de regel zal gelden.
>
> `demo ALL=(ALL:ALL) ALL`
> De eerst "`ALL`" zegt dat de regel geldt voor alle hosts.
>
> `demo ALL=(ALL:ALL) ALL`
> Deze "`ALL`" zegt dat de 'demo' user alle commando's kan runnen als alle users.
>
> `demo ALL=(ALL:ALL) ALL`
> Deze "`ALL`" zegt dat de 'demo' user alle comando's kan runnen als alle groepen.
>
> `demo ALL=(ALL:ALL) ALL`
> De laatste "`ALL`" zegt dat deze regel geldt voor alle commando's.

Dit betekent dat onze 'demo' user alle commando's kan uitvoeren d.m.v. `sudo`, zolang het paswoord wordt opgegeven.



12.1.2 Groep root-toegang geven

Als je meerdere beheerders hebt op je systeem, is het niet altijd handig om voor iedereen individueel de sudo-rechten te beheren. Daarom kan je ook via het ingebouwde systeem van groups werken. We kunnen onszelf toevoegen aan de sudo groep. Dit kan door het bestand `/etc/group` als root te editeren. We editoren de sudoers file om de sudo groep root rechten te geven.

In de file `/etc/group/` zoeken we naar de regel die met sudo begint en voegen onze gebruikersnaam toe. lijn `sudo:x:27: jelmergilles` We geven de sudo-groep root rechten door in het bestand "`%sudo ALL=(ALL:ALL) ALL`" toe te voegen.
Betekenis '`%`': elke member van de groep krijgt deze rechten.

12.1.3 Gebruiker selectieve root-rechten geven

We willen een user enkel rechten geven tot het uitvoeren van het ifconfig commanndo:
In de `sudoers`-file de volgende lijn toevoegen:
`<user_name> ALL=(ALL) NOPASSWD: /sbin/ifconfig`
met `/sbin/ifconfig` het pad naar het commando.

12.2 services

Veel standaard-services hebben een vast poortnummer. Deze staan opgelijst in de services file, deze wordt gebruikt om nummers naar namen te mappen.
Bestandspad: `/etc/services`
De locatie van de service-file wordt gedefinieerd door `_PATH_SERVICES` in `<netdb.h>`.

12.2.1 Poortnummers

```
HTTPS:          443/tcp of udp
SMTP:           25/tcp
IMAP3:          220/tcp of udp
POP3:           110/tcp of udp
```

12.3 .bashrc

`.bashrc` is een configuratiebestand die verschillende "initialisatie" comando's bevat voor de shell.

- Creating useful aliases (for example `alias ll=''ls -l''`).
- Adding more directories to PATH.
- Setting new environment variables.

De system-wide configuratiebestand staat in `/etc/bash.bashrc`, het gebruiker specifieke configuratiebestand staat in `/home/student/.bashrc`.

Karakter klassen	
`.`	elk karakter behalve newline
`1 of a`	een enkel karakter
`\w \d \s`	woord, digit, whitespace
`\W \D \S`	geen woord, digit, whitespace
`[abc]`	een karakter a, b, of c van de set [abc]
`[^abc]`	geen karakter a, b, of c van de set [abc]
`[a-g]`	een karakter tussen a & g
Anchors	
`^abc$`	begin / einde van de string
`\b`	word boundary, hele woorden vb. `\bwoord\b`
Escaped characters	
`\.  \* \\`	escaped special characters
`\t \n \r`	tab, linefeed, carriage return
`\u00A9`	unicode escaped ©
Groups & Lookaround	
`(abc)`	capture group
`\1`	backreference to group #1
`(?:abc)`	non-capturing group
`(?=abc)`	positive lookahead
`(?!abc)`	negative lookahead
Quantifiers & Alternation	
`a* a+ a?`	0 or more, 1 or more, 0 or 1
`a{5} a{2,}`	exactly five, two or more
`a{1,3}`	between one & three
`a+?  a{2,}?`	match as few as possible
`ab|cd`	match ab or cd

Tabel 3: Cheatsheet Regex