Linux

Linux Documentatie

    - Python 3.8 installeren in Debian 10 (Buster)
    - Python installeren vanuit source in Debian 10
    - Virtualenv gebruiken in Python3

- Servers
    - Nginx installatie Debian 10 (Nginx repository)

- SSH
    - SSH Public Key Authenticatie
    - SSH shortcuts maken in Linux

- Ubuntu - LinuxMint
    - Opstartbare USB stick maken voor installatie

- Users en Groups
    - Een gebruiker aan een groep (of tweede groep) toevoegen

- Virtualisatie
    - VirtualBox problemen met Bridged netwerk en Wifi
    - Meer dan 4 netwerk interfaces maken in Virtualbox

- VPN
- Encryptie
    - Encryptie en decryptie met gpg

Algemeen

Algemeen

Rechten toekennen met chmod

Met het commando chmod kan je rechten toekennen en wijzigen op bestanden en directories. De syntax voor chmod is:

```
$ chmod options mode file
```

Meer informatie hierover kan je opvragen in de terminal met het commando:

```
man chmod
```

Gebruikersklassen en chmod

In Linux heeft elk bestand op een server individuele toegangsrechten. Hetzelfde geldt voor directories. De toegangsrechten worden geregeld volgens **drie klassen van gebruikers**:

- **Eigenaar (user)**: een gebruiker die een bestand aanmaakt in Unix wordt gewoonlijk automatisch gedefinieerd als de ‘user’ van het bestand. Het eigendom kan later worden gewijzigd door het commando ‘chown’. In de symbolische notatie krijgt de gebruikersklasse ‘user’ de letter ‘u’.
- **Groep (group)**: de gebruikersklasse ‘group’ omvat verschillende gebruikersaccounts op de server. In Unix-bestandssystemen wordt elk gebruikersaccount automatisch toegewezen aan een hoofdgroep. Lidmaatschap in andere groepen is ook mogelijk. Zowel de eigenaar als de rootgebruiker kan bestanden groeperen met behulp van het ‘chgrp’ commando. De gebruikersklasse ‘groep’ wordt in de symbolische notatie met de letter ‘g’ weergegeven.
- **Andere (others)**: deze gebruikersklasse omvat alle gebruikers die geen user zijn van het bestand en ook niet bij een groep horen. Deze gebruikersklasse heeft de symbolische notatie ‘o’.

Toegangsrechten

Het Linux bestandssysteem heeft **drie basisrechten**. Elk van de bovengenoemde gebruikersaccounts kan verschillende rechten hebben die door de eigenaar van het bestand zijn toegewezen:

- **Lezen (read)**: het toegangsrecht ‘read’ geeft een gebruiker leestoegang tot een bestand. In het geval van een directory kan hij de hele inhoud van die map lezen. Maar met het read-toegangsrecht kun je geen bestandsrechten bekijken. In de symbolische notatie wordt dit recht gecodeerd met de letter ‘r’, ook wel r-bit genoemd.
- **Schrijven (write)**: gebruikers met het toegangsrecht ‘write’ mogen de inhoud van het bestand wijzigen. Als ‘write’ is toegewezen aan een directory, mag de gebruiker in die map submappen en bestanden aanmaken. Het schrijfrecht wordt in de symbolische notatie weergegeven met een ‘w’. Het heet daarom ook wel w-bit.
- **Uitvoeren (execute)**: een gebruiker met dit toegangsrecht mag bestanden als programma uitvoeren en naar een andere directory gaan om daar te kijken welke submappen of bestanden er staan. De symbolische notatie kent aan dit recht de letter ‘x’ toe, ofwel het x-bit.

Denk er bij het toekennen van toegangsrechten aan dat **machtigingen in Linux niet worden geërfd**. Ofwel, als je een bestand in een directory aanmaakt, krijgt dat bestand niet automatisch de toegangsrechten die voor de hele directory gelden. Individuele bestandsrechten vloeien voort uit de machtiging van het programma dat het bestand heeft aangemaakt.

Weergave van toegangsrechten in de terminal

Met het volgende commando kan je de bestanden en direcories weergeven en de bijbehorende rechten zien:

```
ls -l
```

Hier onder zie je een voorbeeld:

```
alex@alex-Lenovo-G500:~/test-chmod$ ls -l
totaal 12
drwxr-xr-x 2 alex alex 4096 mei 4 13:47 dir01
drwxr-xr-x 2 alex alex 4096 mei 4 13:48 dir02
drwxr-xr-x 2 alex alex 4096 mei 4 13:48 dir03
-rw-r--r-- 1 alex alex 0 mei 4 13:47 file01.txt
```

De volgende illustratie laat duidelijk zien hoe de rechten in elkaar steken:

chmod01.jpg

De rwx (read, write, execute) parameters kan je aanpassen voor de **eigenaar (user)**, **groep (group)** en **anderen (others)**.
Dit is het zogenaamde rechtenmasker en bestaat uit een **octale notatie** van drie cijfers.
Onderstaande illustratie maakt dit duidelijk:

chmod02.png

Ieder cijfer bestaat dus uit drie bits en kan dus lopen van 0 t/m 7. Op deze manier kan je de rechten toekennen.

Voorbeelden:

Stel dat we in ons voorbeeld de rechten willen wijzigen van de directory “dir01”, zodanig dat iedereen alles mag, dan kan dat als volgt:

```
chmod 777 dir01
```

Stel dat we in ons voorbeeld de rechten willen wijzigen van het bestand “file01.txt”, zodanig dat alleen de gebruiker kan lezen en schrijven, dan kan dat als volgt:

```
chmod 600 file01.txt
```

Stel dat we alleen de rechten van **alle onderliggende directories willen wijzigen** en niet van de bestanden:

```
find /home/alex/test-chmod/ -type d -exec chmod 755 {} \;
```

Stel dat we alleen de rechten van **alle onderliggende bestanden** willen wijzigen en niet van de directories:

```
find /home/alex/test-chmod/ -type f -exec chmod 644 {} \;
```

Symbolische notatie

In plaats van de octale notatie kan men ook de rechten wijzigen door een symbolische notatie. Zie ook onderstaande illustratie:

chmod03.png

Voorbeelden:

Stel dat we in ons voorbeeld de rechten willen wijzigen van de directory “dir01”, zodanig dat iedereen alles mag, dan kan dat als volgt:

```
chmod ugo+rwx dir01
```

Onderstaand commando heeft hetzelfde effect:

```
chmod a+rwx dir01
```

Op deze manier kan men ook bepaalde rechten weer verwijderen:

```
chmod o-rwx dir01
```

Sticky bit:

Met sticky bit kan je de rechten op mappen en bestanden beperken. Als je een map gedeeld hebt met andere gebruikers, maar je wilt niet dat die gebruikers bestanden kunnen verwijderen of hernoemen, dan kan je de sticky bit aanzetten om dit te voorkomen. De rechten van de gebruikers worden daarmee ingeperkt. Alleen de eigenaar en de root gebruiker hebben dan nog wel alle rechten. Gebruikers kunnen dan nog wel bestanden toevoegen. Sticky bit wordt enkel toegepast op directories en niet op bestanden. Wanneer de sticky bit aan staat voor een directory, zijn de bestanden in die directory ook beschermd.
Het toepassen van de sticky bit is eenvoudig, hiervoor wordt de **t** vlag gebruikt met **chmod**:

Voorbeelden:

In onderstaand voorbeeld wordt de map “dir01” beschermd door het sticky bit aan te zetten:

```
chmod +t dir01
```

Uitzetten kan dan dus weer met:

```
chmod -t dir01
```

Je kunt zien wanneer er een sticky bit is gezet op een directory. Dit wordt aangegeven met de letter “t” in het laatste octet:

```
alex@alex-Lenovo-G500:~/test-chmod$ ls -l
totaal 12
drwxrwxrwt 2 alex alex 4096 mei 4 13:47 dir01
drwxr-xr-x 2 alex alex 4096 mei 4 13:48 dir02
drwxr-xr-x 2 alex alex 4096 mei 4 13:48 dir03
-rw------- 1 alex alex 0 mei 4 13:47 file01.txt
```

Stel dat je nu de sticky bit wilt aan zetten voor alle onderliggende directories, dan kan dat als volgt:

```
find /home/alex/test-chmod/ -type d -exec chmod +t {} \;
```

Uitzetten kan dan weer met:

```
find /home/alex/test-chmod/ -type d -exec chmod -t {} \;
```

Algemeen

TAR archief maken

Met behulp van onderstaand commando kan men een archiefbestand maken:

```
tar --exclude='db.*' -zcvf vaultwarden-$(date +%d-%m-%Y_%H-%M-%S).tar.gz /opt/dockerdata/vaultwarden
```

In bovenstaand voorbeeld wordt er van mijn Vaultwarden directory (/opt/dockerdata/vaultwarden) een archiefbestand gemaakt waarin ook de datum en tijd verwerkt zijn in de bestandsnaam.
De optie "--exclude=db.*" zorgt ervoor dat de bestanden die beginnen met "db." uitgesloten worden in het archief (in dit geval de Sqlite3 database bestanden).

Wanneer je veel bestanden hebt die je wilt uitsluiten kan je ook verwijzen naar een exclude file.
In deze file zet je dan per regel het bestand of directory in die je wilt uitlsuiten.
Het tar commando ziet er dan als volgt uit:

```
tar -zcvf vaultwarden-$(date +%d-%m-%Y_%H-%M-%S).tar.gz -X exclude_file.txt /opt/dockerdata/vaultwarden
```

De parameter -X verwijst naar de exclude file, in ons voorbeeld exclude_file.txt
Je mag ook de volledige parameter opgeven, dit is --exclude-from=exclude_file.txt

Algemeen

Het find commando gebruiken om directories te laten zien

Onderstaand commando kan handig zijn om even snel de directories te laten zien in Linux:

```
find directory_path -maxdepth 1 -type d
```

Het aantal levels van de directories wat je wilt zien bepaal je met de parameter "-maxdepth".

Bash

Bash

Wachten op toets in Bash script (press any key)

Soms kan het handig zijn dat je in een Bash script moet wachten op een toets aanslag voordat het terminal venster weer sluit.
Dit kan eenvoudig door onderstaand commando toe te voegen in je Bash script:

```
read -n 1 -s -r -p "Press any key to continue"
```

Na het uitvoeren van het script wacht deze nu op een willekeurige toets aanslag om vervolgens het terminal venster te sluiten.

Databases

Backup maken van een Sqlite3 database

Met behulp van onderstaand commando kan je veilig een backup maken van een in gebruik zijnde Sqlite3 database:

```
sqlite3 /opt/dockerdata/vaultwarden/db.sqlite3 ".backup dbbackup-$(date +%d-%m-%Y_%H-%M-%S).sqlite3"
```

In bovenstaand voorbeeld krijgt de gemaakte backup de datum en tijd mee in de bestandsnaam. Het backup bestand is eigenlijk een kopie van de werkelijke database.

Debian

Debian backports repository instellen

In de Debian backports repository vind je pakketten die vaak nieuwer zijn dan uit de originele “stable” repository.
Nadeel is wel dat deze Debian pakketten vaak in de fase “testing” of “unstable” zitten, dus niet volledig getest zijn op betrouwbaarheid, veiligheid of stabiliteit. Maar voor sommige toepassingen kan het wel handig zijn om deze backports repository te gebruiken.

We gaan nu de repository toevoegen aan de Debian sources.list. Open hiervoor een nieuw bestand in je editor:

```
nano /etc/apt/sources.list.d/backports.list
```

En voeg hier de volgende regel in:

```
deb http://deb.debian.org/debian buster-backports main
```

Hierna kan je de update uitvoeren:

```
apt-get update
```

Na deze actie kan je pakketten installeren uit de backports repository.
Wanneer je een pakket specifiek uit de backports repository wilt installeren kan dat met:

```
apt-get -t buster-backports install "package"
```

Debian

Installatie van PHP 7.4 / 7.3 / 7.2 / 7.1 in Debian 10

In Debian 10 is de standaard PHP versie 7.3
Wanneer je andere versies van PHP wilt gebruiken en eventueel naast elkaar wilt draaien kan je via onderstaande stappen dit voor elkaar krijgen.

Toevoegen van de third-party PHP repository

We moeten eerst een aantal benodigde pakketten installeren:

```
apt update
apt-get install curl wget gnupg2 ca-certificates lsb-release apt-transport-https
```

Importeren van de key:

```
wget https://packages.sury.org/php/apt.gpg
apt-key add apt.gpg
```

Toevoegen van de SURY repository:

```
echo "deb https://packages.sury.org/php/ $(lsb_release -sc) main" | tee /etc/apt/sources.list.d/php7.list
```

Update de repository index:

```
apt-get update
```

Installeren van de PHP pakketten

Afhankelijk van de PHP versie kan je nu de PHP pakketten installeren.
Voor PHP 7.4 kan je dit doen met:

```
apt-get install php7.4 php7.4-cli php7.4-common
```

Op deze manier kan je dus ook PHP 7.3 / 7.2 / 7.1 installeren door de versie aan te passen (dus 7.4 wordt 7.3 of 7.2 of 7.1).

Je kunt dus meerdere PHP versies naast elkaar installeren. Iedere PHP versie heeft zijn eigen configuratie bestanden in de directory /etc/php/7.X

De default PHP versie instellen

Met onderstaand commando kan je de default PHP versie instellen (in dit voorbeeld PHP versie 7.4):

```
update-alternatives --set php /usr/bin/php7.4
```

PHP versie checken

```
php -v
```

Firewall

Configureren van NAT in UFW

Als je gebruik wilt maken van NAT om data van de externe naar de interne netwerk interface te routeren, dan dienen er een aantal configuratiebestanden gewijzigd te worden. Dit zijn etc/default/ufw , /etc/ufw/before.rules en /etc/ufw/sysctl.conf. Open eerst /etc/default/ufw met de editor (bijvoorbeeld nano):

```
nano /etc/default/ufw
```

En wijzig de volgende regel:

```
DEFAULT_FORWARD_POLICY="ACCEPT"
```

Vervolgens dienen we ipv4 forwarding toe te staan. Dit doen we door het bestand /etc/ufw/sysctl.conf te openen in de editor:

```
nano /etc/ufw/sysctl.conf
```

En wijzig de volgende regel:

```
net/ipv4/ip_forward=1
```

Vervolgens openen we /etc/ufw/before.rules in de editor:

```
nano /etc/ufw/before.rules
```

En voeg het volgende toe **voor** de filter regels:

```
# NAT table rules
*nat
:POSTROUTING ACCEPT [0:0]
:PREROUTING ACCEPT [0:0]
-F
# Forward traffic through eth0 - Change to match your out-interface
-A POSTROUTING -s 192.168.1.0/24 -o eth0 -j MASQUERADE
```

```
# don't delete the 'COMMIT' line or these nat table rules won't
# be processed
COMMIT
```

De optie “-F” (Flush) heb ik er ingezet zodat de NAT tabel eerst wordt gewist. Wanneer men UFW uitschakelt en weer inschakelt, komen er dubbele regels in te staan. Hiermee wordt dit voorkomen. Sla het bestand op en herstart UFW met:

```
ufw disable
ufw enable
```

Configureren van Port Forwarding in UFW

Als je bijvoorbeeld verkeer van poort 80 en 443 wilt forwarden naar een server met IP adres 192.168.1.120, dan dien je eerst het bestand /etc/ufw/before.rules te openen in de editor:

```
nano /etc/default/before.rules
```

En wijzig je het bestand als volgt:

```
:PREROUTING ACCEPT [0:0]
-A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT --to-destination 192.168.1.120:80
-A PREROUTING -i eth0 -p tcp --dport 443 -j DNAT --to-destination 192.168.1.120:443
```

Vervolgens herstart je UFW weer met:

```
ufw disable
ufw enable
```

Vervolgens dien je nog wel kenbaar te maken dat tcp verkeer voor poort 80 en 443 toegestaan wordt:

```
ufw allow 80/tcp
ufw allow 443/tcp
```

Bovenstaande regels geven toegang voor **alle** IP adressen van buiten af voor poort 80 en 443. Om dit te beperken tot een specifiek publiek IP adres zou je het volgende kunnen opgeven:

```
ufw allow from <PUBLIEK IP-ADRES> to any port 80 proto tcp
ufw allow from <PUBLIEK IP-ADRES> to any port 443 proto tcp
```

Hier onder volgt nog een voorbeeld van port forwarding waarbij de WAN interface luistert op poort 1999 en vervolgens doorsluist naar poort 80 van de PC met IP-adres 192.168.10.211:

```
-A PREROUTING -i eth0 -p tcp --dport 1999 -j DNAT --to-destination 192.168.10.211:80
```

Bron:

Linuxconfig.org

Firewall

UFW Commando’s

UFW staat voor Uncomplicated FireWall en is de standaard firewall van Ubuntu. Eigenlijk is het een front-end om de standaard firewall configuratie in Linux (IPTables) te vergemakkelijken. Hier onder volgen veel gebruikte commando’s en voorbeelden.

UFW inschakelen

```
ufw enable
```

UFW uitschakelen

```
ufw disable
```

Status checken

```
ufw status verbose
```

Bekijk de regels in /etc/ufw (.rules)

```
ufw show raw
```

UFW Allow syntax en voorbeelden

```
ufw allow <port>/<optional: protocol>
Voorbeeld: To allow incoming tcp and udp packet on port 53
ufw allow 53
Voorbeeld: To allow incoming tcp packets on port 53
ufw allow 53/tcp
Voorbeeld: To allow incoming udp packets on port 53
ufw allow 53/udp
```

UFW Deny syntax en voorbeelden

```
ufw deny <port>/<optional: protocol>
Voorbeeld: To deny tcp and udp packets on port 53
```

ufw deny 53

Voorbeeld: To deny incoming tcp packets on port 53

ufw deny 53/tcp

Voorbeeld: To deny incoming udp packets on port 53

ufw deny 53/udp

Bestaande regels verwijderen

To delete a rule, simply prefix the original rule with delete. For example, if the original rule was:

ufw deny 80/tcp

Use this to delete it:

ufw delete deny 80/tcp

Port ranges

For port ranges you can use the colon (:) to separate the lowest and the highest port in the range.

For example:

ufw allow 10000:15000/udp

ufw deny 8000:8100/tcp

Comments

Example: Open port 53 and write a comment about rule too

ufw allow 53 comment 'open tcp and udp port 53 for dns'

UFW services

You can also allow or deny by service name since ufw reads from /etc/services

To see get a list of services:

less /etc/services

Allow by Service Name

ufw allow <service name>

Voorbeeld: to allow ssh by name

ufw allow ssh

Deny by Service Name

ufw deny <service name>

Voorbeeld: to deny ssh by name

ufw deny ssh

UFW Logging

To enable logging use:

ufw logging on

To disable logging use:

ufw logging off

UFW Advanced Syntax: Allow

Allow by Specific IP:

ufw allow from <ip address>

Voorbeeld:To allow packets from 207.46.232.182:

ufw allow from 207.46.232.182

Allow by Subnet:

You may use a net mask :

ufw allow from 192.168.1.0/24

Allow by specific port and IP address:

ufw allow from <target> to <destination> port <port number>

Voorbeeld: allow IP address 192.168.0.4 access to port 22 for all protocols

ufw allow from 192.168.0.4 to any port 22

Allow by specific port, IP address and protocol:

ufw allow from <target> to <destination> port <port number> proto <protocol name>

Voorbeeld: allow IP address 192.168.0.4 access to port 22 using TCP

ufw allow from 192.168.0.4 to any port 22 proto tcp

UFW Advanced Syntax: Deny

Deny by specific IP:

ufw deny from <ip address>

Voorbeeld:To block packets from 207.46.232.182:

ufw deny from 207.46.232.182

Deny by specific port and IP address:

ufw deny from <ip address> to <protocol> port <port number>

Voorbeeld: deny ip address 192.168.0.1 access to port 22 for all protocols

ufw deny from 192.168.0.1 to any port 22

Advanced Example

Scenario: You want to block access to port 22 from 192.168.0.1 and 192.168.0.7

but allow all other 192.168.0.x IPs to have access to port 22 using tcp

ufw deny from 192.168.0.1 to any port 22

ufw deny from 192.168.0.7 to any port 22

```
ufw allow from 192.168.0.0/24 to any port 22 proto tcp
```

Laat de regels zien met een regelnummer

```
ufw status numbered
```

Regels verwijderen of tussenvoegen op basis van regelnummers

You may then delete rules using the number.

This will delete the first rule and rules will shift up to fill in the list:

ufw delete 1

Insert numbered rule:

ufw insert 1 allow from <ip address>

Enable Ping

Note: Security by obscurity may be of very little actual benefit with modern cracker scripts.

By default, UFW allows ping requests. You may find you wish to leave (icmp) ping requests enabled

to diagnose networking problems.

In order to disable ping (icmp) requests, you need to edit /etc/ufw/before.rules and remove the following lines:

```
# ok icmp codes
-A ufw-before-input -p icmp --icmp-type destination-unreachable -j ACCEPT
-A ufw-before-input -p icmp --icmp-type source-quench -j ACCEPT
-A ufw-before-input -p icmp --icmp-type time-exceeded -j ACCEPT
-A ufw-before-input -p icmp --icmp-type parameter-problem -j ACCEPT
-A ufw-before-input -p icmp --icmp-type echo-request -j ACCEPT
```

or change the "ACCEPT" to "DROP"

```
# ok icmp codes
-A ufw-before-input -p icmp --icmp-type destination-unreachable -j DROP
-A ufw-before-input -p icmp --icmp-type source-quench -j DROP
-A ufw-before-input -p icmp --icmp-type time-exceeded -j DROP
-A ufw-before-input -p icmp --icmp-type parameter-problem -j DROP
-A ufw-before-input -p icmp --icmp-type echo-request -j DROP
```

Reload UFW

When you edit UFW configuration files, you need to run reload command.

For example, you can edit /etc/ufw/before.rules, enter:

nano /etc/ufw/before.rules

After saving the changes reload UFW with:

ufw reload

Resetting UFW to defaults and make inactive

```
ufw reset
```

Bron:

https://help.ubuntu.com/community/UFW

Netwerk

Netwerk

OpenWRT opkg-upgrade script

Binnen OpenWRT kan je met het onderstaand commando bekijken of er updates beschikbaar zijn voor de geïnstalleerde pakketten:

```
opkg list-upgradable
```

Wanneer dit het geval is, dan is het standaard niet echt makkelijk om deze pakketten allemaal tegelijk te upgraden. Het commando om pakketten binnen OpenWRT te upgraden is:

```
opkg upgrade pakket01 pakket02 pakket03
```

Er is een script beschikbaar die dit allemaal makkelijker maakt, namelijk opkg-upgrade. Het installeren van het script gaat het makkelijkst via het commando **git**. Wanneer je onderstaande commando's uitvoert, dan wordt het script gedownload en geïnstalleerd.

```
git clone git://github.com/tavinus/opkg-upgrade.git
cd opkg-upgrade
./opkg-upgrade.sh -i
```

Het script wordt dan in een directory geplaatst die ook in het pad staat. Het is dan direct uit te voeren met het volgende commando:

```
opkg-upgrade
```

Wanneer er dan upgrades beschikbaar zijn, dan krijg je direct de vraag of je ze allemaal wilt upgraden. Zie ook onderstaande screenshot:

openwrt-opkg-upgrade-1.jpg

Er zijn nog meer opties mogelijk met het script. Voor deze opties kan je de help opvragen met:

```
opkg-upgrade -h
```

Bron:

GitHub opkg-upgrade

Netwerk

OpenWRT SSH Public Key Authenticatie

In dit artikel binnen de kennisbank is al beschreven hoe je SSH authenticatie kan toepassen binnen Linux. In de OpenWRT router software kan je via de Luci web interface heel eenvoudig de inhoud van de public key plakken in het daarvoor bestemde veld. OpenWRT maakt standaard gebruik van de Dropbear SSH server. Log in via de OpenWRT web interface en ga vervolgens naar het menu **System –>> Administration**

openwrt-ssh01.jpg

Onder in de pagina vind je dan het veld **SSH-Keys**. Hier kan je de inhoud van de public key in plakken en vervolgens opslaan met **Save and Apply.**

openwrt-ssh02.jpg

Onder het kopje **SSH Access** kan je vervolgens nog aangeven of je SSH password authentication wilt toestaan of niet. Ook voor de root gebruiker.

openwrt-ssh03.jpg

Netwerk

TAP interface maken in Ubuntu / Linuxmint

Methode 1:

Maak een interface aan:

```
sudo nano /etc/network/interfaces.d/tap0.cfg
```

Plaats hier het volgende in:

```
auto tap0
iface tap0 inet manual
    pre-up ip tuntap add dev tap0 mode tap user alex group netdev
    pre-up ip a add 10.1.1.1/24 dev tap0
    up ip link set dev tap0 up
    post-down ip link del dev tap0
```

Handmatig kan je de interface verwijderen met:

```
sudo ifdown tap0
```

en weer up brengen met:

```
sudo ifup tap0
```

Methode 2:

Installeer pakket uml-utilities:

```
sudo apt-get install uml-utilities
```

Maak een script:

```
nano /home/alex/scripts/create-tap-int.sh
```

Plaats het onderstaande hier in:

```
#! /bin/bash
tunctl -u alex -g netdev -t tap0
ifconfig tap0 10.1.1.1 netmask 255.255.255.0 up
```

Pas de rechten aan:

```
sudo chmod u+x /home/alex/scripts/create-tap-int.sh
```

Zorg er voor dat het script automatisch start bij het opstarten van de PC.
Maak hiervoor het volgende aan:

```
sudo nano /etc/systemd/system/tap-int.service
```

En plaats hier het volgende in:

```
[Unit]
Description=Create TAP interface
After=network.target

[Service]
ExecStart=/home/alex/scripts/create-tap-int.sh

[Install]
WantedBy=multi-user.target
```

Onderstaande opdracht uitvoeren om bovenstaande script automatisch te starten tijdens opstarten:

```
sudo systemctl enable tap-int.service
```

En start dan de nieuw gemaakte service op met:

```
sudo systemctl start tap-int.service
```

Je kunt de TAP interface verwijderen met:

```
sudo tunctl -d tap0
```

Netwerk

Netstat Listening Ports

Met onderstaand commando kan je in Linux de "Listening Ports" bekijken.
Dit kan handig zijn wanneer er bijvoorbeeld een service niet wilt starten omdat er al een poort in gebruik is.

```
sudo netstat -tunlp
```

```
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:54443           0.0.0.0:*               LISTEN      2263244/nginx: mast
tcp        0      0 127.0.0.1:139           0.0.0.0:*               LISTEN      760/smbd
tcp        0      0 192.168.2.201:139       0.0.0.0:*               LISTEN      760/smbd
tcp        0      0 0.0.0.0:8080            0.0.0.0:*               LISTEN      1527051/docker-prox
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      2263244/nginx: mast
tcp        0      0 0.0.0.0:8081            0.0.0.0:*               LISTEN      2490499/mympd
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      598/dropbear
tcp        0      0 0.0.0.0:5432            0.0.0.0:*               LISTEN      1527161/docker-prox
tcp        0      0 0.0.0.0:3000            0.0.0.0:*               LISTEN      868/ruby2.7
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      644886/msmtpd
tcp        0      0 0.0.0.0:8090            0.0.0.0:*               LISTEN      1526981/docker-prox
tcp        0      0 0.0.0.0:8123            0.0.0.0:*               LISTEN      1538239/python3
tcp        0      0 0.0.0.0:8443            0.0.0.0:*               LISTEN      1527032/docker-prox
tcp        0      0 0.0.0.0:44443           0.0.0.0:*               LISTEN      2263244/nginx: mast
tcp        0      0 0.0.0.0:64443           0.0.0.0:*               LISTEN      2263244/nginx: mast
tcp        0      0 127.0.0.1:445           0.0.0.0:*               LISTEN      760/smbd
tcp        0      0 192.168.2.201:445       0.0.0.0:*               LISTEN      760/smbd
tcp        0      0 192.168.2.201:40000     0.0.0.0:*               LISTEN      1538239/python3
tcp        0      0 0.0.0.0:6052            0.0.0.0:*               LISTEN      1526995/docker-prox
```

Python

Default Python versie wijzigen in Debian 10

Je kunt meerdere Python versies installeren in Linux.
Om te zien welke versie op dit moment actief (default) is kan je het volgende uitvoeren:

```
python -V
```

Om te zien welke versies geinstalleerd zijn in Linux kan je het volgende commando uitvoeren:

```
ls /usr/bin/python*
```

Prioriteren van de verschillende Python versies

Wanneer je met bovenstaand commando weet welke Python versies zijn geinstalleerd, dan kan je met onderstaand commando de prioriteit aangeven:

```
update-alternatives --install /usr/bin/python python /usr/bin/python2.7 1
update-alternatives --install /usr/bin/python python /usr/bin/python3.8 2
```

In bovenstaand geval heeft Python versie 3.8 de hoogste prioriteit, dus dan is dit de default Python versie.

Wanneer je een Python versie hebt geïnstalleerd vanuit source (zelf gecompileerd), dan kan je deze versie toevoegen met:

```
update-alternatives --install /usr/bin/python python /usr/local/bin/python3.9 3
```

Hiermee wordt Python3.9 op prioriteit 3 gezet en heeft dus in dit geval de hoogste prioriteit. Deze versie is dan de actieve Python versie.

Switchen tussen default Python versie

Wanneer je een andere Python versie de default versie wilt maken, voer dan onderstaand commando uit:

```
update-alternatives --config python
```

Je kunt de default versie weer controleren met: python -V

Python 3.8 installeren in Debian 10 (Buster)

Python 3.8 installeren

In Debian 10 hebben we standaard alleen de beschikking over Python 2.7 en Python 3.7. Je kan met onderstaand commando controleren welke versies van Python geinstalleerd zijn in Debian:

```
ls /usr/bin/python*
```

De actieve Python versie kan je controleren met:

```
python -V
```

Nu kunnen we Python vanuit de source installeren, maar we hebben ook de beschikking over een repository die beschikbaar is gesteld door Pascal Roeleven. Het voordeel is dat we dan op de normale manier de Python 3.8 pakketten kunnen installeren.
Voordat we hiermee verder gaan moeten we eerst nog een paar benodigde pakketten installeren:

```
apt-get install gpg wget
```

Daarna kunnen we de GPG key installeren die nodig is:

```
wget https://pascalroeleven.nl/deb-pascalroeleven.gpg
apt-key add deb-pascalroeleven.gpg
```

Nu moeten we nog de sources.list aanmaken. Hiervoor maken we een nieuw bestand aan met de tekst editor:

```
nano /etc/apt/sources.list.d/python3.8-pascalroeleven.list
```

Hier plaatsen we het volgende in:

```
deb http://deb.pascalroeleven.nl/python3.8 buster-backports main
```

Hierna een update uitvoeren:

```
apt-get update
```

En dan uiteindelijk Python 3.8 installeren:

```
apt-get install python3.8 python3.8-venv python3.8-dev
```

BRON:

Python 3.8 backport for Debian buster

Python installeren vanuit source in Debian 10

De nieuwste Python versie is vrij eenvoudig te installeren vanuit de broncode (source). In onderstaande stappen wordt beschreven hoe men dit kan doen.

Benodigde pakketten installeren

```
apt-get install build-essential libreadline-gplv2-dev libncursesw5-dev libssl-dev libsqlite3-dev tk-dev libgdbm-dev
libc6-dev libbz2-dev libffi-dev zlib1g-dev
```

Downloaden Python broncode

Op het moment van schrijven is Python 3.9.4 de nieuwste versie.
Deze gaan we downloaden en uitpakken.

```
cd /usr/src
wget https://www.python.org/ftp/python/3.9.4/Python-3.9.4.tgz
tar xzf Python-3.9.4.tgz
```

Python compileren en installeren

```
cd Python-3.9.4
./configure --enable-optimizations
make altinstall
```

Het laatste commando “**make altinstall**” zorgt er voor dat de nieuwe versie naast de bestaande Python versies wordt geïnstalleerd. Doet men dit niet, dan wordt de bestaande versie overschreven.
Python 3.9.4 is nu geïnstalleerd in /usr/local/bin.

Controleren Python versie

Je kunt nu de Python versie controleren met:

```
python3.9 -V
```

Python verwijderen

Wanneer Python op deze manier wordt geïnstalleerd, dan kan men niet met een commando de installatie weer ongedaan maken. Hiervoor dient men de bestanden en directories met de hand te verwijderen. Onderstaande commando's zorgen ervoor dat alles verwijderd wordt:

```
rm -f /usr/local/bin/2to3-3.9
rm -f /usr/local/bin/easy_install-3.9
rm -f /usr/local/bin/idle3.9
rm -f /usr/local/bin/pip3.9
rm -f /usr/local/bin/pydoc3.9
rm -f /usr/local/bin/python3.9
rm -f /usr/local/bin/python3.9-config
rm -rf /usr/local/bin/include/python3.9
rm -f /usr/local/lib/libpython3.9.a
rm -rf /usr/local/lib/python3.9
rm -rf /usr/local/lib/pkgconfig
rm -f /usr/local/share/man/man1/python3.9.1
```

BRON:

How to Install Python 3.8 on Ubuntu, Debian and LinuxMint

Python

Virtualenv gebruiken in Python3

Virtualenv is een tool om een geïsoleerde Python omgeving te maken.
Deze omgeving heeft zijn eigen installatie directory die geen libraries deelt met andere virtuele omgevingen.

Verschil tussen virtualenv en venv

venv is een Python package dat standaard aanwezig is in Python3 (niet in Python2).
virtualenv is een python library die meer functionaliteiten biedt dan **venv**.
Via onderstaande link kan je zien wat de verschillen zijn tussen deze twee omgevingen:

- https://virtualenv.pypa.io/en/stable

Virtualenv installeren met behulp van pip3

We gaan er van uit dat Python3 is geïnstalleerd vanuit de source zoals beschreven in dit artikel.
Op het moment van schrijven is dit Python3.9.4. Je kan met onderstaand commando controleren welke versie Python actief is:

```
python3.9 -V
```

Met onderstaand commando kan je zien wat het pad is waar Python3.9 is geïnstalleerd:

```
which python3.9
```

In ons geval is dat /usr/local/bin/python3.9

We gaan nu eerst de package **pip** updaten. **pip** is de Python package installer.

```
python3.9 -m pip install --upgrade pip
```

Hierna kunnen we de **virtualenv** package gaan installeren:

```
pip3 install virtualenv
```

Virtuele Python omgeving maken

We kunnen nu een virtuele omgeving aanmaken met behulp van het commando **virtualenv**.
Let er op dat je ingelogd bent als de gebruiker waarvoor je de virtuele omgeving wilt maken.
In onderstaand voorbeeld log ik in vanuit de root console als de gebruiker alex:

```
sudo -su alex
```

En maak dan nu de virtuele omgeving aan met:

```
virtualenv -p /usr/local/bin/python3.9 /home/alex/pythonvenv
```

Je krijgt dan een vergelijkbare output te zien zoals hier onder:

```
alex@tux:~$ virtualenv -p /usr/local/bin/python3.9 /home/alex/pythonvenv


created virtual environment CPython3.9.4.final.0-64 in 274ms
creator CPython3Posix(dest=/mnt/sdb3/home/alex/pythonvenv, clear=False, no_vcs_ignore=False,
global=False)
seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy,
app_data_dir=/mnt/sdb3/home/alex/.local/share/virtualenv)
added seed packages: pip==21.0.1, setuptools==56.0.0, wheel==0.36.2
activators BashActivator,CShellActivator,FishActivator,PowerShellActivator,PythonActivator,XonshActivator
```

Je kan de virtuele omgeving een willekeurige naam geven. In bovenstaand voorbeeld is dit dus pythonvenv.
Er wordt dan automatisch een directory aangemaakt met deze naam waarin de virtuele omgeving in wordt gemaakt.
Wanneer je daarin kijkt zie je de volgende directories staan:

```
alex@tux:~/pythonvenv$ ls -l
total 12
drwxr-xr-x 2 alex alex 4096 Apr 23 11:59 bin
drwxr-xr-x 3 alex alex 4096 Apr 23 11:59 lib
-rw-r--r-- 1 alex alex  227 Apr 23 11:59 pyvenv.cfg
```

Je ziet dat er binnen de virtuele omgeving de directories **bin** en **lib** zijn aangemaakt, evenals een configuratie bestand **pyenv.cfg**.
De virtuele omgeving heeft nu zijn eigen python en pip versie en de benodigde libraries.
Het activeren van de virtuele python omgeving kan je doen met:

```
source /home/alex/pythonvenv/bin/activate
```

Je ziet dan ook de prompt veranderen in:

```
(pythonvenv) alex@tux:~$
```

De prefix tussen haakjes (pythonvenv) geeft aan dat je in de virtuele python omgeving werkt.
Het de-activeren kan nu met het commando:

```
deactivate
```

Je krijgt dan weer de normale prompt terug.

Verwijderen van de virtuele Python omgeving

Het verwijderen van de virtuele Python omgeving is niets meer dan het verwijderen van de complete directory inclusief onderliggende directories en bestanden:

```
rm -rf /home/alex/pythonvenv
```

BRON:

Installing and using virtualenv with Python 3

Servers

Nginx installatie Debian 10 (Nginx repository)

Nginx is een snelle en lichtgewicht webserver die veel gebruikt wordt (naast Apache webserver). In Debian heb je standaard de beschikking over de Nginx pakketten, maar dit is niet de nieuwste versie.
We kunnen de repository toevoegen van Nginx zelf. Hiermee hebben we dan de beschikking over de nieuwste versies.

Installeren benodigde pakketten

```
apt-get install curl gnupg2 ca-certificates lsb-release
```

Repository toevoegen voor de Nginx stable release

```
echo "deb http://nginx.org/packages/debian `lsb_release -cs` nginx" \ | sudo tee /etc/apt/sources.list.d/nginx.list
```

Repository toevoegen voor de Nginx mainline release

```
echo "deb http://nginx.org/packages/mainline/debian `lsb_release -cs` nginx" \ | sudo tee
/etc/apt/sources.list.d/nginx.list
```

Instellen repository pinning

Met onderstaand commando stellen we Debian zodanig in dat de Nginx repository voorrang heeft op de standaard Debian repository:

```
echo -e "Package: *\nPin: origin nginx.org\nPin: release o=nginx\nPin-Priority: 900\n" \     | sudo tee
/etc/apt/preferences.d/99nginx
```

Importeren van de Nginx signing key

```
curl -fsSL https://nginx.org/keys/nginx_signing.key | sudo apt-key add -
```

Wanneer je **OK** te zien krijgt is dit goed gegaan.

Installeren van Nginx

```
apt-get update
apt-get install nginx
```

Bron:

Nginx website

Hostup – How to Install Nginx and PHP 7.4 on Debian 10

SSH

SSH Public Key Authenticatie

public-private-300x127-1.png

SSH wordt gebruikt om op afstand in te kunnen loggen op een Linux systeem. Je kunt inloggen door middel van een gebruikersnaam en wachtwoord, maar veiliger is het om dit te doen met behulp van public key authenticatie. Hiervoor dienen we een public en private key te genereren op de PC waar vandaan we willen inloggen op het remote systeem. Hier onder volgen voorbeelden voor zowel een Linux als Windows PC.

SSH key’s genereren op een Linux PC

Open een terminal (Ctrl-Alt-T) en voer het volgende commando uit om een private en public key te genereren:

```
ssh-keygen
```

Er wordt nu gevraagd een bestandsnaam op te geven. Wanneer je dit niet doet, wordt standaard de naam **id_rsa** (private key) en **id_rsa.pub** (public key) gegeven aan de bestanden. Standaard worden de private en public key opgeslagen in de verborgen directory genaamd **.ssh** van de gebruiker waarmee ingelogd is. Dit is dus de directory **~/.ssh**. Er wordt ook gevraagd om een “passphrase” tijdens het genereren. Dit is een wachtwoord waarmee de private key beveiligd wordt. Je kunt dit eventueel overslaan door bij de vraag niets op te geven. Het wachtwoord wordt gevraagd tijdens de inlog sessie op de remote machine. Indien geen wachtwoord is opgegeven wordt je direct ingelogd.

ssh-keygen-e1520435536763.png

Wanneer je een bestandsnaam opgeeft tijdens het genereren, dan dien je deze bestandsnaam op te geven in het **ssh** commando door middel van de optie **-i (identity file)**. Bij de standaard naam (id_rsa) is dit niet nodig. Hier onder staat een voorbeeld:

```
ssh -i ~/.ssh/linuxfun.key username@linuxfun.nl
```

Als je gebruik maakt van een SSH config bestand in de .ssh directory, dan kan je dit opgeven met de parameter **IdentityFile**. Zie het voorbeeld hier onder:

```
Host linuxfun
    HostName linuxfun.nl
    User username
```

```
IdentityFile ~/.ssh/linuxfun.key
```

Meer informatie over het SSH config bestand vind je **hier**. Nu we de private en public key gegenereerd hebben, moeten we de public key op de remote machine zien te krijgen. Dit kan o.a. door middel van het commando **scp**. Een voorbeeld staat hier onder:

```
scp ~/.ssh/id_rsa.pub username@linuxfun.nl:
```

De public key **id_rsa.pub** wordt hiermee gekopieerd naar de home directory van de gebruiker **<username>.** Wanneer de public key op de remote machine staat, dien je nu daarop in te loggen. We moeten de inhoud van de public key in het bestand **~/.ssh/authorized_keys** zien te krijgen. Wanneer de directory .**ssh** en het bestand **authorized_keys** nog niet bestaan op de remote machine, dan dienen we deze eerst aan te maken. Zie de commando's hier onder:

```
mkdir ~/.ssh
touch ~/.ssh/authorized_keys
```

Wanneer het bestand **authorized_keys** al bestaat, dan wordt door bovenstaande commando's het bestand niet overschreven. We kunnen nu de inhoud van de public key toevoegen aan het bestand **authorized_keys** met het commando:

```
cat ~/id_rsa.pub >> ~/.ssh/authorized_keys
```

Je kunt controleren of de public key goed in het bestand **authorized_keys** staat met het commando:

```
more ~/.ssh/authorized_keys
```

Je kunt nu veilig de public key verwijderen van het remote systeem met:

```
rm ~/id_rsa.pub
```

Wanneer je een kopie wilt bewaren van de public key, dan kan je deze het beste verplaatsen naar de **.ssh** directory:

```
mv ~/id_rsa.pub ~/.ssh/
```

Het is nu mogelijk in te loggen op het remote systeem door middel van de ssh key's. Dit gebeurt nu automatisch zonder een inlognaam en wachtwoord op te hoeven geven. Wanneer men een private key heeft gegenereerd met een wachtwoord (passphrase), dan dient men dit wachtwoord op te geven tijdens het inloggen.

SSH key’s genereren op een Windows PC

Onder Microsoft Windows kan je de public en private key’s genereren met behulp van de tool **PuTTYgen**. Dit is een tool die geïnstalleerd wordt samen met de SSH client **PuTTY**. Je kunt dit **hier** downloaden. Na installatie vind je de tools terug in het Windows start menu onder **PuTTY**. Start nu het programma **PuTTYgen** op en klik op **Generate**.

puttygen01.jpg

Je kunt nu de key’s genereren door de muis te bewegen in het vak **Key**.

puttygen02.jpg

Wanneer dit klaar is, zie je het volgende resultaat:

puttygen03.jpg

Door nu op de knop **Save private key** te klikken, kan je de private key opslaan op je Windows PC. Er wordt dan gevraagd of je de private key wilt opslaan zonder wachtwoord (passphrase). Wil je de private key beveiligen met een wachtwoord, vul dan de regel **Key passphrase** in en bevestig dit in de regel **Confirm passphrase**. De private key wordt opgeslagen met de extensie **.ppk** en kan direct gebruikt worden met de SSH client **PuTTY**. De public key, voor de remote machine, kunnen we naar het klembord kopiëren door met de rechter muisknop in het veld **public key** te klikken. Hier kiezen we vervolgens **Alles selecteren** (als dit nog niet geselecteerd is) en daarna voor **Kopiëren**. Door nu in te loggen op de remote machine, kan je vervolgens de public key plakken in het bestand **~/.ssh/authorized_keys** met behulp van je favoriete tekst editor.

puttygen04.jpg

Om de private key te gebruiken in **PuTTY** open je dit programma en maak je een nieuwe connectie aan of je opent een bestaande connectie. Vervolgens ga je in de linker boom structuur naar de categorie **Connection** en selecteer je hier onder **Data**. In het veld **Auto-login username** kan je de gebruiker invullen waarmee je automatisch wilt inloggen.

putty01.jpg

Vervolgens kies je onder **Connection** de optie **SSH** en hier onder optie **Auth**. In het veld **Private key file for authentication** blader je naar het private key bestand met de .ppk extensie.

putty02.jpg

Om de configuratie op te slaan selecteer je boven in de boom structuur voor **Session** en vervolgens **Save**.

putty03.jpg

Wanneer dit is opgeslagen, kan je de verbinding maken met de remote machine en log je automatisch in. Mocht je de private key beveiligd hebben met een wachtwoord, voer dit dan in wanneer je inlogt en hierom gevraagd wordt.

Uitschakelen password login in SSH

Nu we gebruik maken van public en private keys, is het verstandig om het inloggen via wachtwoorden uit te schakelen. Hier onder volgen de instructies voor zowel **OpenSSH** als **Dropbear** voor Debian Linux:

OpenSSH

Open het volgende bestand in een tekst editor:

```
nano /etc/ssh/sshd_config
```

Pas daar de volgende opties aan:

```
ChallengeResponseAuthentication no
PasswordAuthentication no
UsePAM no
```

Wanneer je niet wilt dat je als root kan inloggen, pas dan onderstaande optie aan:

```
class="lang:default decode:true " title="Disable root login">PermitRootLogin no
```

Sla het bestand op en herstart de ssh daemon met:

```
/etc/init.d/ssh reload
OF
sudo systemctl reload ssh
```

Dropbear

Open het volgende bestand in een tekst editor:

```
nano /etc/default/dropbear
```

En pas de volgende optie aan:

```
DROPBEAR_EXTRA_ARGS="-g -s"
```

Wanneer je niet wilt dat je als root kan inloggen, dan kan je de parameter **-w** nog toevoegen.
Herstart de Dropbear daemon met:

```
/etc/init.d/dropbear restart
```

Bronnen:

Set up SSH public-key authentication to connect to a remote system

Dropbear, SFTP and passwordless logins in Debian

How to disable ssh password login on Linux to increase security

SSH

SSH shortcuts maken in Linux

ssh-300x110-1.jpg

SSH wordt veelvuldig gebruikt in Linux om via een terminal in te loggen op een remote machine. Dit gaat dan via de command line en ziet er bijvoorbeeld als volgt uit: **ssh -p 2222 root@192.168.7.6** (Hierin is -p de poort waar ssh op luistert, standaard poort 22). Nu kan je in Linux zogenaamde ssh shortcuts maken, zodat je niet iedere keer die lange commando's hoeft in te tikken. Je krijgt dan bijvoorbeeld een commando als: **ssh thuis** Dit kan je vastleggen in het bestand **config**. Dit bestand hoort dan te staan in de verborgen directory **.ssh** die op zijn beurt weer staat onder de Home directory van een Linux gebruiker. We gaan dus eerst naar de juiste directory met behulp van onderstaand commando:

```
cd ~/.ssh
```

Hierna kunnen we het bestand **config** gaan maken met:

```
nano config
```

Hier kunnen we de regels plaatsen die we nodig hebben. **Voorbeeld:**

```
Host thuis
    HostName 192.168.7.6
    User root
    Port 2222
Host example2
    HostName example.com
    User root
Host example3
    HostName 64.233.160.0
    User userxyz123
    Port 56000
Host amazon1
    HostName ec2.amazon.com
    User ec2-user
```

```
    IdentityFile /path/to/special/privatekey/amazon.pem
```

Sla het bestand op. De ssh shortcuts zijn direct actief. Uitgaande van bovenstaand voorbeeld kan je nu verbinding maken met de commando’s: **ssh thuis, ssh example2, ssh example3, ssh amazon1**

Bron:

How to Create an SSH Shortcut

Ubuntu - LinuxMint

Opstartbare USB stick maken voor installatie

Met onderstaand commando kan men vanaf de commandline een opstartbare USB stick maken om bijvoorbeeld LinuxMint te installeren. Deze instructie kan ook gebruikt worden om een opstartbare USB stick te maken voor Debian 10 (Buster).

```
sudo dd bs=4M if=/path/to/linuxmint-19.3-xfce-64bit.iso of=/dev/sdx status=progress oflag=sync
```

Zorg er voor dat je het juiste pad naar het ISO bestand invoert.
/dev/sdx is de USB stick waar je naar toe wilt schrijven.
Controleer met **sudo fdisk -l** welk device jouw USB stick is.

Users en Groups

Een gebruiker aan een groep (of tweede groep) toevoegen

Een nieuwe groep aanmaken

```
groupadd mynewgroup
```

Een bestaande gebruiker aan een groep toevoegen

```
usermod -a -G examplegroup exampleusername
```

Voorbeeld: om de gebruiker ***geek*** aan de groep ***sudo*** toe te voegen voer je het volgende commando uit:

```
usermod -a -G sudo geek
```

Primary group wijzigen van een gebruiker

```
usermod -g groupname username
```

Let hier op de optie **-g**. De kleine letter **g** wil zeggen dat je een Primary Group toekent.
Wanneer je de hoofdletter **-G** gebruikt, dan ken je de gebruiker toe aan de Secondary Group.

Laat alle groepen zien waar de huidige gebruiker aan toegekend is

```
groups
```

Om de numerieke ID’s te zien die toegekend zijn aan een groep kan je het commando ***id*** gebruiken:

```
id
```

Om de groepen te laten zien waar een andere gebruiker aan toegekend is kan je onderstaand commando geven:

```
groups exampleusername
```

En de numerieke ID’s met:

```
id exampleusername
```

Een nieuwe gebruiker en aan een groep toekennen in één commando

```
useradd -G examplegroup exampleusername
```

Voorbeeld: maak een nieuwe gebruiker ***jsmith*** en voeg hem toe aan de groep ***ftp***:

```
useradd -G ftp jsmith
```

Een wachtwoord toekennen aan de gebruiker kan dan achteraf met:

```
passwd jsmith
```

Een gebruiker toevoegen aan meerdere groepen

```
usermod -a -G group1,group2,group3 exampleusername
```

Je kan een gebruiker toevoegen aan net zoveel groepen als je zelf wilt door deze te scheiden met een komma.

Laat alle groepen zien in het systeem

```
getent group
```

Bron:

https://www.howtogeek.com/50787/add-a-user-to-a-group-or-second-group-on-linux/

Virtualisatie

Virtualisatie

VirtualBox problemen met Bridged netwerk en Wifi

Ik maak al lange tijd gebruik van Virtualbox onder Linux om het een en ander uit te proberen.
Op een gegeven moment merkte ik dat de bridged network adapter in VirtualBox niet meer werkte in combinatie met de wifi adapter in mijn laptop (LinuxMint 20.1). Bridged mode werkte wel met de vaste netwerk adapter, en het werkte wel met de wifi adapter wanneer ik een vast IP adres toekende aan de netwerk adapter in de virtuele machine. Blijkbaar werkt dan de DHCP client niet goed.
Ik kreeg dit weer werkend door de optie **Promiscuous Mode** op **VM’s toestaan** te zetten in de geavanceerde instellingen van de virtuele netwerk adapter (zie onderstaande afbeelding).

VirtualBox_Bridged.png

Virtualisatie

Meer dan 4 netwerk interfaces maken in Virtualbox

Set Mode and enable NIC

When a Mode is set the NIC will be enabled.

1. Set NIC five to Host-Only Mode and User vboxnet0 (Initially existing Host-Only Net)

```
VBoxManage modifyvm network-test --nic5 hostonly
VBoxManage modifyvm network-test --hostonlyadapter5 "vboxnet0"
```

2. Set NIC five to Bridge eth1

```
VBoxManage modifyvm network-test --nic5 bridged
VBoxManage modifyvm network-test --bridgeadapter5 "eth1'
```

3. Set NIC to NAT Mode

```
VBoxManage modifyvm network-test --nic5 nat
```

4. Use the internal VM net “test01”

```
VBoxManage modifyvm network-test --nic5 intnet
VBoxManage modifyvm network-test --intnet5 "test01"
```

Configuring NIC

- To use VLAN and some other things in VMs the Promiscuous mode have to be enabled

```
VBoxManage modifyvm network-test --nicpromisc5 allow-all
```

- Use other hardware type (Intel Pro/1000 MT Server)

```
VBoxManage modifyvm network-test --nictype5 82545EM
```

- Dis/connect cable

```
VBoxManage modifyvm network-test --cableconnected5 off
```

BRON: More than 4 Network Cards in Virtualbox

VPN

Encryptie

Encryptie en decryptie met gpg

Met behulp van de Linux tool gpg kan je bestanden versleutelen (encrypten) en ook weer ontsleutelen (decrypten).
De methode die we hier bespreken is "Symmetric Encryption". We versleutelen hierbij het bestand met een enkel wachtwoord (passphrase).

In Debian Linux kan je eerst controleren of gpg al geïnstalleerd is met:

```
gpg --version
```

Mocht dit nog niet het geval zijn, dan kan je gpg installeren met:

```
sudo apt install gnupg
```

Encryptie van bestanden

Je kunt een bestand versleutelen met de volgende opdracht. Hierbij geven we het versleutelwachtwoord op in de commandoregel:

```
gpg --batch -c --passphrase 'passphrase' file.txt
```

Veiliger is om het wachtwoord (passphrase) in een apart bestand te zetten en deze zodanig rechten te geven dat deze bijvoorbeeld enkel te lezen is door de root gebruiker. In onderstaand voorbeeld staat het wachtwoord in /etc/gpg/pass.txt

```
gpg --batch -c --passphrase-file /etc/gpg/pass.txt file.txt
```

Het versleutelde bestand krijgt automatisch de extensie .gpg, dus in ons voorbeeld file.txt.gpg

Decryptie van bestanden

Wanneer je het wachtwoord meegeeft in de commandline:

```
gpg --batch --output file.txt --passphrase 'passphrase' --decrypt file.txt.gpg
```

Met behulp van het wachtwoordbestand wordt dit:

```
gpg --batch --output file.txt --passphrase-file /etc/gpg/pass.txt --decrypt file.txt.gpg
```

Hierbij geeft de optie --output aan welke naam je het ontsleutelde bestand wilt geven, in dit voorbeeld file.txt

Decryptie van meerdere bestanden tegelijk

Onderstaand commando is handig wanneer je meerdere bestanden tegelijk wilt decrypten.
Het voordeel hiervan is dat je geen "output" file hoeft op te geven.
De bestandsnaam wordt dezelfde naam zonder de .gpg extensie.
Je kunt hiermee ook een enkel bestand decrypten door de wildcard aan te passen.

```
gpg --batch --passphrase-file /etc/gpg/pass.txt --decrypt-files *.gpg
```