

Network

- [OpenWRT opkg-upgrade script](#)
- [OpenWRT SSH Public Key Authenticatie](#)
- [TAP interface maken in Ubuntu / Linuxmint](#)
- [Netstat Listening Ports](#)

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

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

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
```


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 -tulp
```

```
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
```