appunti.md

LEZIONE 3 NETWORKING

Per vedere le interfacce di rete uso e interfacciarsi con esse uso la famiglia di software di *ip*.

Per visulizzare le interfacce di rete uso *ip address*:

```
[osboxes@osboxes ~]$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:55:02:a2 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global noprefixroute dynamic enp0s3
       valid_lft 85164sec preferred_lft 85164sec
    inet6 fe80::464a:ffed:9491:76a2/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:36:04:ee brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.102/24 brd 192.168.56.255 scope global noprefixroute dynamic enp0s8
       valid_lft 1043sec preferred_lft 1043sec
    inet6 fe80::587d:6ecb:f6d:11e1/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
4: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 52:54:00:99:8b:99 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
       valid_lft forever preferred_lft forever
5: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master virbr0 state DOWN group default qlen 1000
    link/ether 52:54:00:99:8b:99 brd ff:ff:ff:ff:ff:ff

[osboxes@osboxes ~]$ ip -s link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    RX: bytes  packets  errors  dropped overrun mcast
    0          0        0       0       0       0
    TX: bytes  packets  errors  dropped carrier collsns
    0          0        0       0       0       0
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:55:02:a2 brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped overrun mcast
    102217223  71522    0       0       0       0
    TX: bytes  packets  errors  dropped carrier collsns
    389351     6205     0       0       0       0
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:36:04:ee brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped overrun mcast
    34395      364      0       0       0       0
    TX: bytes  packets  errors  dropped carrier collsns
    37434      241      0       0       0       0
4: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DEFAULT group default qlen 1000
    link/ether 52:54:00:99:8b:99 brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped overrun mcast
    0          0        0       0       0       0
    TX: bytes  packets  errors  dropped carrier collsns
    0          0        0       0       0       0
5: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master virbr0 state DOWN mode DEFAULT group default qle
    link/ether 52:54:00:99:8b:99 brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped overrun mcast
    0          0        0       0       0       0
    TX: bytes  packets  errors  dropped carrier collsns
    0          0        0       0       0       0
```

Solitamente l'hypervisor resetta il MAC delle macchine virtuli per evitare conflitti. Le interfacce di rete hanno una policy sui nomi. Si hanno varie naming convencion:

- ***loopback***, solitamente chiamata *lo*, che serve per avere un loop sullo stack di rete, permettendo l'uso di localhost. Si hanno policy di sicurezza diverse e un livello di sicurezza minore
- ***eth*** che sono le normali interfacce di rete (nominate progressivamente), che ora vengono chiamate ora, sugli host fisici, *enp...* per evere dei nomi deterministici (a differenza delle *eth*)
- ***wlp*** per le reti wireless
- ***br*** per il bridge, ovvero un interfaccia virtuale che ne mette in comunicazione altre a livello due della scala ISO/OSI, quindi è uno *switch virtuale* e aggiunge una *MAC address table*

Su *inet* vedo l'*ipv4* con la mask e l'indirizzo di broadcast ottenuto con quellamask. Con *inet6* abbiamo un indirizzo *ipv6*, che presenta un campo indirizzo molto più grande di ipv4, infatti ha 128 bit ma è limitato dal vecchio hardware c he non lo supportano livello hardware ma, a volte, sono software. Si risolve il p roblema di esaurimento degli ipv4 (che sono finiti 4 anni fa). Non viene riportat o in decimale ma in esadecimale a gruppi di 16bit. Inet non mostra però tutti gli 8 campi infatti i blocchi di solo zero vengono omessi o rappresentati da un 0, tutto questo una volta sola per indirizzo. Solitamente si ha mask 64, a maschera fissa. Il doppio "::" significa "riempi con zero fino ad arrivare a 128". Si ha il *link local*. La rete link local di ipv4 si ha 169.254.0.0/16, col terzo e quarto byte random, che è un indirizzo di emergenza quando in mancanza di dhcp. Nell'ipv 6 il link local è la base. In ipv4 un host se vuole mandare un pacchetto ad un altro si usa il protocollo ARP che cerca tra tutti gli host, pericolosissimo per attacchi mim (man in the middle). Ipv6 non ha indirizzamento di broadcast e si hanno inter facce di rete utilizzabile su interfaccia volendo. Si ha il protocollo neighbor discovery protocol, *ndp*, che lavora a livello 3, parlando tra gli f80 e usando multicast.

Per dare dare un indrizzo ip usp *ip address add <ip> /dev/periferica*, per vedere la *routing table* poi ho:

```
[osboxes@osboxes ~]$ ip route show
default via 10.0.2.2 dev enp0s3 proto dhcp metric 100
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15 metric 100
192.168.56.0/24 dev enp0s8 proto kernel scope link src 192.168.56.102 metric 101
192.168.122.0/24 dev virbr0 proto kernel scope link src 192.168.122.1
```

Se voglio aggiungere un default gateway *ip r a 0.0.0.0/0 via <ip>*

ifconfig e route sono stati abbandonati perché, con l'avvento dei *namespace* e dei *container*, ifconfig e route non permettevano di visualizzarne routing table virtuali o altro, erano inutilizzabili. Ma tutto questo non sopravvive al reboot, serve quindi un metodo che renda persistente la configurazione di rete. La soluzione più banale è uno script che viene chiamato all'avvio. Red Hat usa *network manager* per fare tutto ciò, che è un *service* dotato di varie gui, anche testuali come *nmtui*. Ovviamente si ha *nmcli* con cui ci interfacciamo a network manager per "parlare" di connessioni. Vedo le connessioni con:

```
[osboxes@osboxes ~]$ nmcli con show
NAME                UUID                                  TYPE      DEVICE
virbr0              d65ae02a-46e5-4760-84c0-325eb163eee4  bridge    virbr0
Wired connection 1  e3a7c938-9ce4-3734-a73c-aa40c9dd3efa  ethernet  enp0s3
Wired connection 2  a5a1aa6b-86db-3093-995c-30ce6a4699ec  ethernet  enp0s8
```

Dove il nome della connessione è uguale a quello dell'interfaccia. Per cancellare una connessione *nmcli con del <nome interfaccia>* e per creare *nmcli con add <tipo> <nome interfaccia> con -name <nome>*, con tipo che specifica se è ethernet, wifi, bridge, il nome è opzionale ma deve essere diverso dal *nome interfaccia*. I file di config sono in:

```
DEVICE=lo
IPADDR=127.0.0.1
NETMASK=255.0.0.0
NETWORK=127.0.0.0
# If you're having problems with gated making 127.0.0.0/8 a martian,
# you can change this to something else (255.255.255.255, for example)
BROADCAST=127.255.255.255
ONBOOT=yes
NAME=loopback
```

Per rileggere i file di config *nmcli con reload*. Per dire che un interfaccia deve essere attiva all'interfaccia, con nmcli, si ha il parametro ONBOOT. (**sistemare questa parte**)

La sicurezza lato firewall è gestito dal kernel con un modulo chiamato *netfilter*. *netfilter* si richiama col comando *iptables* che ne è il client. Recentemente si è abbandonato *netfilter* per passare a *nft* con *nftables*, che è comunque un client runtime. Entrambi i moduli sono comunque gestiti da *firewalld*. firewalld può essere gestito via gui ma sopratutto via cli. Si ha il concetto di zone (trusted o meno, etc...). Le configurazioni possono essere automaticamente rese persistenti o modificate runtime. Da cli vedo le zomne con:

```
[osboxes@osboxes ~]$ sudo firewall-cmd --list-all
```

```
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3 enp0s8
  sources:
  services: ssh dhcpv6-client
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

per modificare per esempio si ha: *firewall-cmd --add-port=25/tcp --permanent*, facendolo permanente:

```
[osboxes@osboxes ~]$ sudo firewall-cmd --add-port=25/tcp
success
[osboxes@osboxes ~]$ sudo firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3 enp0s8
  sources:
  services: ssh dhcpv6-client
  ports: 25/tcp
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

e quindi si ha:

```
[osboxes@osboxes ~]$ sudo iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0           udp dpt:53
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:53
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0           udp dpt:67
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:67
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           ctstate RELATED,ESTABLISHED
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
INPUT_direct  all  --  0.0.0.0/0           0.0.0.0/0
INPUT_ZONES_SOURCE  all  --  0.0.0.0/0           0.0.0.0/0
INPUT_ZONES  all  --  0.0.0.0/0           0.0.0.0/0
DROP       all  --  0.0.0.0/0            0.0.0.0/0           ctstate INVALID
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            192.168.122.0/24    ctstate RELATED,ESTABLISHED
ACCEPT     all  --  192.168.122.0/24     0.0.0.0/0
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-port-unreachable
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-port-unreachable
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           ctstate RELATED,ESTABLISHED
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
FORWARD_direct  all  --  0.0.0.0/0           0.0.0.0/0
FORWARD_IN_ZONES_SOURCE  all  --  0.0.0.0/0           0.0.0.0/0
FORWARD_IN_ZONES  all  --  0.0.0.0/0           0.0.0.0/0
FORWARD_OUT_ZONES_SOURCE  all  --  0.0.0.0/0           0.0.0.0/0
FORWARD_OUT_ZONES  all  --  0.0.0.0/0           0.0.0.0/0
DROP       all  --  0.0.0.0/0            0.0.0.0/0           ctstate INVALID
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0           udp dpt:68
OUTPUT_direct  all  --  0.0.0.0/0           0.0.0.0/0
```

```
Chain FORWARD_IN_ZONES (1 references)
target     prot opt source               destination
FWDI_public  all  --  0.0.0.0/0            0.0.0.0/0           [goto]
FWDI_public  all  --  0.0.0.0/0            0.0.0.0/0           [goto]
FWDI_public  all  --  0.0.0.0/0            0.0.0.0/0           [goto]

Chain FORWARD_IN_ZONES_SOURCE (1 references)
target     prot opt source               destination

Chain FORWARD_OUT_ZONES (1 references)
target     prot opt source               destination
FWDO_public  all  --  0.0.0.0/0            0.0.0.0/0           [goto]
FWDO_public  all  --  0.0.0.0/0            0.0.0.0/0           [goto]
FWDO_public  all  --  0.0.0.0/0            0.0.0.0/0           [goto]

Chain FORWARD_OUT_ZONES_SOURCE (1 references)
target     prot opt source               destination

Chain FORWARD_direct (1 references)
target     prot opt source               destination

Chain FWDI_public (3 references)
target     prot opt source               destination
FWDI_public_log  all  --  0.0.0.0/0            0.0.0.0/0
FWDI_public_deny  all  --  0.0.0.0/0            0.0.0.0/0
FWDI_public_allow  all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0

Chain FWDI_public_allow (1 references)
target     prot opt source               destination

Chain FWDI_public_deny (1 references)
target     prot opt source               destination

Chain FWDI_public_log (1 references)
target     prot opt source               destination

Chain FWDO_public (3 references)
target     prot opt source               destination
FWDO_public_log  all  --  0.0.0.0/0            0.0.0.0/0
FWDO_public_deny  all  --  0.0.0.0/0            0.0.0.0/0
FWDO_public_allow  all  --  0.0.0.0/0            0.0.0.0/0

Chain FWDO_public_allow (1 references)
target     prot opt source               destination

Chain FWDO_public_deny (1 references)
target     prot opt source               destination

Chain FWDO_public_log (1 references)
target     prot opt source               destination

Chain INPUT_ZONES (1 references)
target     prot opt source               destination
IN_public  all  --  0.0.0.0/0            0.0.0.0/0           [goto]
IN_public  all  --  0.0.0.0/0            0.0.0.0/0           [goto]
IN_public  all  --  0.0.0.0/0            0.0.0.0/0           [goto]

Chain INPUT_ZONES_SOURCE (1 references)
target     prot opt source               destination

Chain INPUT_direct (1 references)
target     prot opt source               destination

Chain IN_public (3 references)
target     prot opt source               destination
IN_public_log  all  --  0.0.0.0/0            0.0.0.0/0
IN_public_deny  all  --  0.0.0.0/0            0.0.0.0/0
IN_public_allow  all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0

Chain IN_public_allow (1 references)
target     prot opt source               destination
```

```
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:22 ctstate NEW
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:25 ctstate NEW

Chain IN_public_deny (1 references)
target     prot opt source               destination

Chain IN_public_log (1 references)
target     prot opt source               destination

Chain OUTPUT_direct (1 references)
target     prot opt source               destination
```

Posso aggiungere servizi con *--add-service*.