Systemd-networkd & firewalld

David Alan Gilbert – March 2026

Time to refresh my router...

- **Was Ubuntu**
  - Debian style /etc/interfaces for network definition
  - Shorewall for firewalling
    - Has been unmaintained for a while
  - The version of Ubuntu I was using was EoL so...
- **Switching to Debian**
  - Lets go with modern tooling

Systemd-networkd & firewalld

- **Systemd-networkd**
  - Part of systemd (as the name says…)
  - man systemd-networkd.service
  - https://wiki.archlinux.org/title/Systemd-networkd
  - Maybe a better choice on servers than desktops?
- **Firewalld**
  - *NOT* part of Systemd (which I hadn’t realised)
  - man firewalld
  - https://firewalld.org/documentation/

[aside, netplan]

- **Mostly an abstraction**
  - On top of NetworkManager or systemd-networkd
    - So just learn one thing and drive both??
  - Default in current Ubuntu
    and Debian cloud images

Haven’t used it

Systemd-networkd intro

- **networkctl for status and simple changes**
- **systemctl reload systemd-networkd.service (or networkctl reload?)**
- **Config files in ‘ini’ format**
  - /usr/lib/systemd/network - system/package provided
  - /etc/systemd/network - ones you create and edit
  - Note: systemd-analyze can’t check them
  - General demon config in /etc/systemd/networkd.conf (didn’t touch)
  - Watch out for which [section] the option you’re changing needs to be in
- **Includes own dhcp client and server, but not DNS server (used bind)**

Networkctl examples

- networkctl up or down
- networkctl status
- ‘unmanaged’ means not being managed by systemd-networkd

```
dg@router:~$ networkctl
IDX LINK         TYPE     OPERATIONAL SETUP
  1 lo           loopback carrier     unmanaged
  2 ethdave      ether    routable    configured
  3 enp10s4      ether    off         unmanaged
  4              ether    no-carrier  configured
  5 ethexternal  ether    enslaved    configured
  6 enp3s0f1     ether    off         unmanaged
  7              ether    routable    configured
  8 enp10s6      ether    off         unmanaged
  9 ethmobile    ether    no-carrier  configuring
 10 ethwifi      ether    no-carrier  configured
 11              ether    no-carrier  configured
 12 brinternet   bridge   routable    configured

12 links listed.
```

Sets of 3 files

- **‘.link’**
  - man systemd.link
  - Interface naming, duplex, speeds, low level config
- **‘.network’**
  - man systemd.network !! Note no ‘d’ and no ‘-’
  - IP config, routing, dhcp client & server, queue discipline, masquerade etc
- **‘.netdev’ (rarer)**
  - man systemd.netdev
  - Virtual devices, bridges, bonds, tunnels, vlans, wireguard

General file notes

- **Typically per interface, and typically named for it**
  - e.g. 10-name-ethfoo.network
  - But can be general (e.g. tell all the interfaces that match … rule to use dhcp)
  - Can be switched on things like host or kernel command line flags
- **Same section name mean different things in different files**
  - e.g. [Link] is different in .link and .network
- **Ordering of files**
  - Rare but some can override others, and read at the end
  - Typical systemd alphanumeric loading order
  - See /usr/lib/systemd/network/99-default.link

.link

- ***Match* chooses**
  - By Mac address, IP,driver, type (e.g. ether or wlan), path (ie PCI bus or slot)
- **Link sets**
  - e.g. an explicit name
  - Name policy (choose old e.g. eth0 or the newer ones based on firmware/PCI
  - Can **set** a MACAddress=
- **Other uses**
  - e.g. set WakeOnLan off for all NICs, or a offload flag for particular driver

```
[Match]
  MACAddress=00:xx:yy:zz:aa:bb

[Link]
  Property=ID_NET_MANAGED_BY=io.systemd.Network
  Name=ethmynic
```

.netdev

- **Virtual devices**
  - Bridges, tunnels etc
- **NetDev common**
  - Pick a name
  - Pick a MAC
- **Separate block for each type**
  - e.g. set STP on a bridge
- **Can set most other tunnels etc that previously had own config**
  - e.g. Wireguard!

```
[NetDev]
Name=brmyname
Kind=bridge
MACAddress=00:aa:bb:cc:dd:ee

[Bridge]
....
```

.network

- **Most ‘normal’ config**
- **‘Online’**
  - Some systemd services wait for networking
  - Can switch it per interface
  - e.g. router boots fully with sshd even if external network down
- **DHCP client built in**

```
[Match]
Name=ethwhatever

[Link]
RequiredForOnline=no

[Network]
DHCP=yes
```

.network (2)

- **Built in DHCP server**
  - (Not kea, not isc..)
- **Note [Link]**
  - Different from .link
- **Can set**
  - Bridge=brwhatever to wire to bridge

```
[Match]
Name=ethwhatever

[Link]
RequiredForOnline=no

[Network]
DHCPServer=yes
Address=192.168.1.1/24
DefaultRouteOnDevice=no

[DHCPServer]
  PoolOffset=64
  DNS=192.168.1.1
```

```
[DHCPServerStaticLease]
  MACAddress=…
  Address=192.168.1.2
```

Firewalld

- **Drives nftables – you don’t need to learn it**
- **Interfaces** belong to ***Zones***
  - **Some default zones – e.g. ‘drop’ or ‘external’ - but be careful they’re not obvious**
  - **‘HOST’ is a special zone**
  - **Zones can have multiple interfaces in**
- ***Policies*** set up rules between **Zones**
- **There are also some rules on Zones themselves**
  - **Again some default ones, but take care**

Firewalld: Configuring

- XML files in /etc/firewalld
  - Fortunately rarely use them directly
  - *firewall-cmd* changes/queries state of running firewall
    - Add –permanent to remember it! (Means you can try something and reboot to recover)
  - *firewall-cmd-offline* needed if in a chroot or daemon not running
- /etc/firewalld/firewalld.conf
  - General config, e.g.

```
DefaultZone=drop
CleanupOnExit=no
LogDenied=all
```

Firewalld: Zones

- firewall-cmd --permanent --new-zone dave
- firewall-cmd --permanent --zone=dave –add-service=ssh
  - Rule on the Zone itself, lets ssh into the host from the interfaces in this zone (also want dhcp and DNS for a router)
- firewall-cmd --permanent --change-interface=ethdave –zone=dave
  - This is changing that interface to be in that zone, add as many interfaces as you like
  - Defaults to allowing traffic between interfaces in the same zone
- firewall-cmd --zone=external --remove-service ssh –permanent
  - *Was default enabled in the ‘external’ shipped with it!*

Firewalld: Policies

- Nothing flows between zones without a policy
- firewall-cmd --permanent --new-policy=outwards
- firewall-cmd --permanent --policy=outwards --add-egress-zone external
- firewall-cmd --permanent --policy=outwards --add-ingress-zone dave
- firewall-cmd --permanent --policy=outwards –set-target=ACCEPT
  - No need to add individual service rules with that, lets everything go from dave→external

Firewalld: Policies – a block, and HOST

- HOST as a zone is the machine running the firewalld
- firewall-cmd --permanent --new-policy=host-block
- firewall-cmd --permanent --policy=host-block --add-ingress-zone HOST
- firewall-cmd --permanent --policy=host-block --add-egress-zone dave
- firewall-cmd --permanent --policy=host-block –set-target=REJECT
  - Disallows the host making an inwards connection to the other zones
- Can have multiple policies between the same pair of zones, e.g. then one to allow something specific
- Can have multiple ingress or egress zones per policy

Firewalld: Querying

- firewall-cmd –get-policies
- firewall-cmd –info-policy=host-block
- firewall-cmd –state
- Can also restart & reload with that, or via systemctl

Firewalld: Logs

- Logs include hint as to which rule tripped:
  - filter_IN_external_REJECT: IN=brinternet OUT= MAC=... SRC=... ....
  - filter_IN_dave_REJECT: IN=ethdave OUT= MAC= ...
  -

Firewalld: Masquerade

- Firewall-cmd –permanent –zone=external –add-masquerade
  - Think that’s on by default, note IPv4 only
- Rich rule for IPv6
  - firewall-cmd --permanent --zone=external --add-rich-rule='rule family="ipv6" masquerade'
  - Other rules generally needed for IPv6 to allow different router soliciation etc to work