Linux Networking Basics

Franz Schäfer

Linux LV, WU Wien

April 27, 2018

Copyleft: This Document may be distributed under GNU GFDL or under Creative Commons CC BY-SA 3.0
Table of contents

1 preface

2 Networking Basics

3 commands to access interfaces

4 Linux Firewalling, VLANs
http://mond.at/cd/
This slides are Copyleft: CC-BY-SA, Use them as you like.
This is the first in a series of 3 lectures on Linux:

1. Networking
2. Server, SSH
3. Backup, Boot, LVM, Virtualization
About Me

- Storage Architect @ sIT-Solutions
- Sysadmin @ IST Austria, head of IT
- Sysadmin @ ZID WU
- ISP (akis, silverserver, ...)
- Nachtrichtentechnik, Regelungstechnik, Computertechnik
- Linux User since 1995 (kernel 1.1.18)
Network Abstraction in Linux

- Physical Connection
  - Ethernet, UTP, Wireless
  - Serial Cable
  - Virtual Connection (Tunnel, VPN)
- Linux Kernel: Interface
- Network Stack: e.g. TCP/IP (in Kernel)
- Userspace Programs: E.g. Webbrowser
Network Abstraction in Linux

Userspace Programs

Linux Kernel

Hardware

Network Stack
Device Driver

libc
interface

ssh client
ifconfig

Browser
# /sbin/ifconfig

```
eth0  Link encap:Ethernet  HWaddr 80:ee:73:83:a9:1e
     inet addr:192.168.79.79  Bcast:192.168.79.255  Mask:255.255.255.0
     inet6 addr: fe80::82ee:73ff:fe83:a91e/64 Scope:Link
     UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
     RX packets:260357  errors:0  dropped:0  overruns:0  frame:0
     TX packets:225288  errors:0  dropped:0  overruns:0  carrier:0
     collisions:0  txqueuelen:1000
     RX bytes:261709698 (249.5 MiB)  TX bytes:29802129 (28.4 MiB)
```
# ip addr

2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 80:ee:73:83:a9:1e brd ff:ff:ff:ff:ff:ff
    inet 192.168.79.79/24 brd 192.168.79.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::82ee:73ff:fe83:a91e/64 scope link
        valid_lft forever preferred_lft forever

# ip -s link

2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT group default qlen 1000
    link/ether 80:ee:73:83:a9:1e brd ff:ff:ff:ff:ff:ff
    RX: bytes packets errors dropped overrun mcast
        261967909 261306 0 0 0 0
    TX: bytes packets errors dropped carrier collsns
        29989420 226506 0 0 0 0
Ethernet

- All nodes can "see" each other
- addressing via MAC address: e.g.: A3:07:56:3C:F3:02
- broadcast to all is possible
2^{32} addresses written in the 256^4 notation:
e.g.: 113.251.19.71
not a valid address: 64.311.17.92
On ethernet: relation of MAC addresses and IP addresses via *arp* protocol

```
# arp -n
```
IPv6

# host -t AAAA www.google.com
www.google.com has IPv6 address 2a00:1450:400c:c0b::68

$2^{128}$ addresses written as 8 blocks of 4 hex digits.

consecutive blocks of 0 can be written as :: (only once per address)
e.g.: ::1

Tools: ping6, traceroute6, “ip -6”
CIDR

Classless Internet Domain Routing
123.24.67.0/24 = 123.24.67.XXX
137.208.0.0/16 = WU-Network = 137.208.xxx.xxx
123.24.67.128/25 =123.24.67.128 to 123.24.67.255
Alternativ: netmask: 255.255.255.128
Private IP Space: RFC 1918

- 10.0.0.0 to 10.255.255.255
  10.0.0.0/8 or e.g. divided into 65536 times /24
- 172.16.0.0 to 172.31.255.255
  172.16.0.0/12 e.g. divided into 1024 /24 networks
- 192.168.0.0 to 192.168.255.255
  192.168.0.0/16 gives 256 networks with /24

e.g.: your home IP and network:
192.168.1.13/24
not routed in the public internet: you need NAT
GUI interface uses NetworkManager to manage networks.
should be disabled on a server
can be controlled via commandline via nmcli
alias interface

# ifconfig eth0:2 192.168.201.42 \  
netmask 255.255.255.0  \  
broadcast 192.168.201.255  
# ifconfig eth0:2 192.168.201.42/24

additional IP address on an existing interface:

# ip addr add 192.168.202.123/24 dev eth0
tcpdump - look at your traffic

```shell
# tcpdump -ni eth0 not port 22

   193.238.157.20.4569: UDP, length 12
13:40:09.322544 IP 141.89.64.1.27650 >
   193.238.157.20.53: 16832% [1au] AAAA? dns.mond.at. (40)
13:40:09.322785 IP 193.238.157.20.53 >
   141.89.64.1.27650: 16832* 0/1/1 (88)
13:40:09.483043 arp who-has 192.168.30.32 (ff:ff:ff:ff:ff:ff) tell 192.168.30.32
13:40:09.516130 IP 194.168.8.110.32771 >
   193.238.157.20.53: 57265 MX? area23.mond.at. (32)
```
# ping www.google.com
PING www.l.google.com (209.85.135.147) 56(84) bytes of data.
64 bytes from mu-in-f147.google.com (209.85.135.147):
icmp_seq=1 ttl=241 time=22.6 ms
64 bytes from mu-in-f147.google.com (209.85.135.147):
icmp_seq=2 ttl=241 time=22.6 ms
# traceroute www.google.com

traceroute to www.l.google.com (209.85.135.103),
  30 hops max, 40 byte packets

1  gw-2-254.wu-wien.ac.at (137.208.254.254)
   0.793 ms  0.769 ms  0.752 ms
2  box-1-19.wu-wien.ac.at (137.208.19.135)
   0.849 ms  0.810 ms  0.879 ms
...

14  mu-in-f103.google.com (209.85.135.103)
   23.536 ms  23.664 ms  23.336 ms
route - how the packets find their way

192.168.1.0/24

192.168.1.94

192.168.1.7

192.168.2.0/24

192.168.2.21

192.168.2.37

192.168.1.0/24

reachable via 192.168.2.1
route - a few examples

# route -n
# route add default gw 192.168.1.1
# route add -net 192.168.2.0/24 gateway 192.168.1.7
turn on ip forwarding

per default packets are not forwarded from one interface to another

in /etc/sysctl.conf

net.ipv4.ip_forward=1
net.ipv4.conf.all.rp_filter=0

or

# echo 1 > /proc/sys/net/ipv4/ip_forward
# echo 0 > /proc/sys/net/ipv4/conf/all/rp_filter
# dhclient -v eth0

Internet Systems Consortium DHCP Client 4.3.1
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/eth0/80:ee:73:83:a9:1e
Sending on LPF/eth0/80:ee:73:83:a9:1e
Sending on Socket/fallback
DHCPREQUEST on eth0 to 255.255.255.255 port 67
DHCPNAK from 192.168.79.1
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 8
DHCPREQUEST on eth0 to 255.255.255.255 port 67
DHCPOFFER from 192.168.79.1
DHCPACK from 192.168.79.1
bound to 192.168.79.108 -- renewal in 34746 seconds.
ifup / ifdown

# ifup eth1
# ifup -a

config file: /etc/network/interfaces

auto lo
iface lo inet loopback

auto eth1
iface eth1 inet dhcp
auto eth0
  iface eth0 inet static
  address 192.168.17.42
  network 192.168.17.0
  netmask 255.255.255.0
  broadcast 192.168.17.255
  gateway 192.168.17.1
  up /root/myfirewall.sh
troubleshooting part 1

- `ifconfig eth0` works?
  - check `modprobe`
  - for wireless: `iwconfig`, `wpa_supplicant`
- do we have the right IP address in `ifconfig` or `ip addr`
  - e.g. use `dhclient`
- check `route -n`
troubleshooting part 2

- ifconfig shows incoming packets?
- tcpdump -ni shows packets?
- ping a machine in the local network (e.g. gateway)
- check arp -n
  - do we see the mac address of the gateway?
- try a traceroute to an outside address
- maybe it is a dns problem
  - ip address works but names do not.
TCP and UDP port numbers

TCP — network stack takes care about providing the illusion of a connection
UDP — you only send packets. they may get lost or may arrive in the wrong order.

Well known ports

- tcp 80  www
- tcp 25  smtp (email sending)
- tcp 22  ssh
- udp 53  dns
iptables

Network

PREROUTING

POSTROUTING

FORWARD

INPUT

OUTPUT

local processes

Routing

Franz Schäfer

Linux Networking Basics
iptables filter examples

show rules:
# iptables -L -n
# iptables -L -n -t nat

flush rules:
# iptables -F

protect access to SSH:
# iptables -I INPUT -j DROP -i eth1 -p tcp \  --dport 22 -s 0/0
# iptables -I INPUT -j ACCEPT -s 182.16.21.0/24 \  -p tcp --dport 22
# iptables -t nat -I POSTROUTING -j SNAT \
-s 10.0.0.0/8 -d ! 10.0.0.0/8 \
--to-source 123.231.12.222

# iptables -t nat -I POSTROUTING \
-j MASQUERADE -s 192.168.1.0/24 \
--out-interface eth1
why VLANs?

We want multiple networks on the same physical cable to connect networks over different switches:
IEEE 802.1q adds a 12-bit VLAN tag to each ethernet packet so we can have about 4096 different VLANs.
**Linux VLAN commands**

```bash
# ifconfig eth0 up
# vconfig add eth0 101
# vconfig add eth0 201

# ifconfig eth0.101 192.168.123.45 ....

can also be done in /etc/network/interfaces
```
installing openvpn

```bash
# apt-get install openvpn
# cd /usr/share/doc/openvpn/examples/sample-config-files
# zcat examples/sample-config-files/server.conf.gz > /etc/openvpn/mondbasis.conf
# openssl dhparam -out dh2048.pem 2048
# chdir /etc/openvpn/
# mkdir cd

copy easy-rsa scripts
and edit ./vars
# ./build-ca
# ./build-key-server openvpn.mond.at
# ./build-key-pass mondhandy@mond.at
```
installing openvpn

edit /etc/default/openvpn

to select the configuration to start on boot
# /etc/init.d/openvpn restart

check logs
# journalctl -xn
# tail -100 /var/log/syslog

openvpn should be listening on port 1194 udp
# netstat -nu --listen -p
openvpn point to point link

```bash
# ifconfig
tun0   Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
    inet addr:10.17.17.1  P-t-P:10.17.17.2  Mask:255.255.255.255
    UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
    RX packets:0  errors:0  dropped:0  overruns:0  frame:0
    TX packets:0  errors:0  dropped:0  overruns:0  carrier:0
    collisions:0  txqueuelen:100
    RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```