Part1: Linux Networking Basics, SSH

Franz Schäfer

Linux LV, WU Wien

November 8, 2019

Table of contents

About this slides

`http://mond.at/cd/`
the slides are Copyleft: CC-BY-SA, Use them as you like.

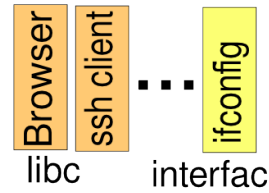
About Me

- System Architect @ s-ITsolutions
- Sysadmin @ IST Austria, Head of IT Team
- Sysadmin @ ZID WU
- ISP (akis, silverserver, ...)
- Nachtrichtentechnik, Regelungstechnik, Computertechnik
- Linux User since 1995 (kernel 1.1.18)

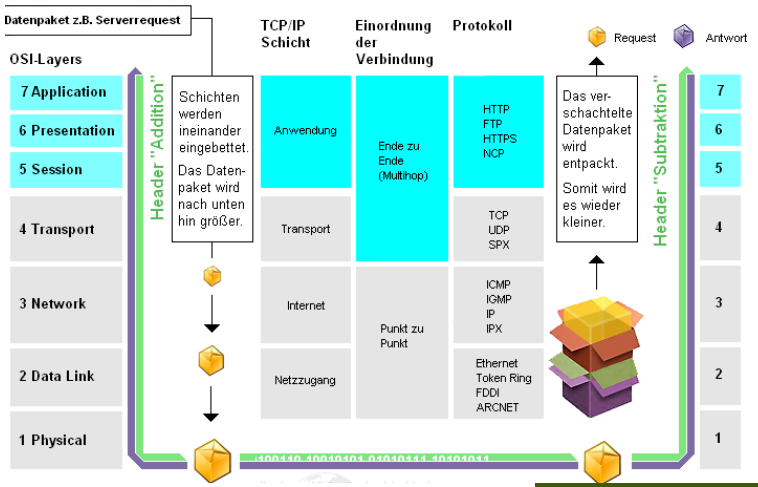
Network Abstraction in Linux

- 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



ISO OSI 7 Layer



ifconfig

```
# /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 tool

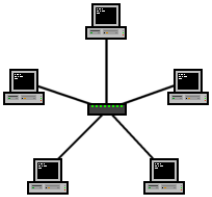
```
# ip addr

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
```

Ethernet



- All nodes can "see" each other
- addressing via MAC address: e.g.: A3:07:56:3C:F3:02
- broadcast to all is possible

IPv4

2^{32} addresses writen 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 writen a 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

network manager

GUI interface uses NetworkManager to manage networks.
should be disabled on a server
can be controlled via comandline 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

```
# tcpdump -ni eth0 not port 22

13:40:09.295326 IP 213.235.242.217.4569 >
     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

```
# 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

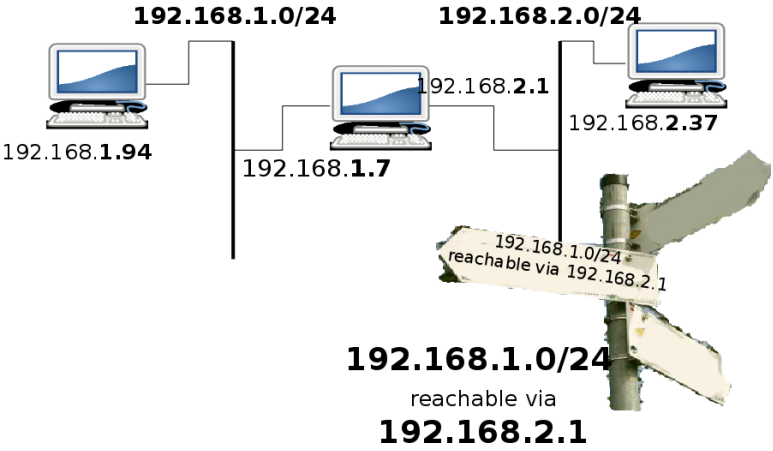
```
# traceroute www.google.com

 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)
```

route - how the packets find their way



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
```

dhcp client

```
# dhclient -v eth0

Internet Systems Consortium DHCP Client 4.3.1
Copyright 2004-2014 Internet Systems Consortium.
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/ifdown used in debian/ubuntu/..
# redhat/fedora/centos: uses /etc/sysconfig/network-scripts/ifcfg-eth0
#
# ifup eth1
# ifup -a

config file: /etc/network/interfaces

auto lo
iface lo inet loopback

auto eth1
iface eth1 inet dhcp
```

/etc/network/interfaces

```
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/myfirwall.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

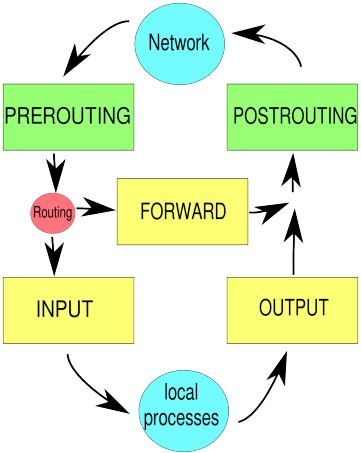
- ifconfig shows incoming packets?
- tcpdump -ni shows packets?
- ping a maschine in the local network (e.g. gateway)
- check arp -n
  do we see the mac address of the gateway?
- try a traceroute to an outside address
- maybe it is a dns problem
  ip address works but names do not.

TCP and UDP port numbers

TCP — network stack takes care about providing the ilusion 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
Routing
FORWARD
INPUT
OUTPUT
local processes

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 nat

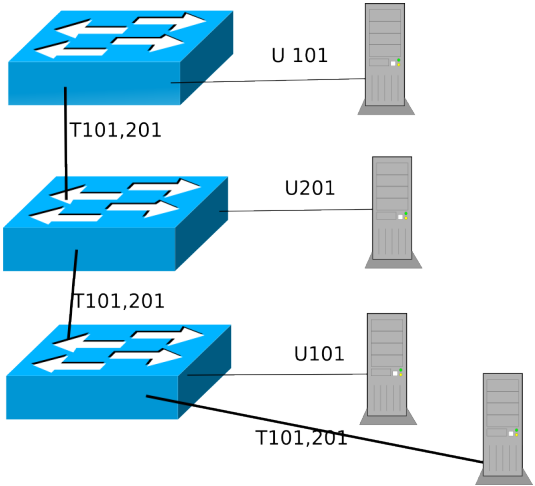
```
# iptables -t nat -I POSTROUTING -j SNAT \
  -s 10.0.0.0/8 -d ! 10.0.0.0/8 \
  --to-source 123.231.12.222

# 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 physcial cable to connect networks over different switchs:
IEEE 802.1q addes a 12bit VLAN tag to each ethernet packet so we can have about 4096 different VLANs.

VLANs example diagram



Linux VLAN commands

```
# 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

```
# 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
```

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

```
# 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)
```

Basic SSH

- Text based connection to remote servers
- Copy files to and from remote servers
- Remote command execution

```
# ssh anna@meinserver.daheim.at
# scp meinedatei.txt hansi@woanders.at:/var/www/
# sftp root@meinserver.daheim.at:blabla.tgz .
# ssh hansi@meinserver.daheim.at "find /tmp"
```

What is Secure in SSH

- Encrypts traffic
- Checks the identitiy of remote hosts

- does **NOT** protect you from compromized local host
- does **NOT** always protect you from compromized remote host

Interactive Remote Login with SSH

```
# ssh h7788999@login.wu-wien.ac.at
The authenticity of host 'login.wu-wien.ac.at (137.208.3.70)' can't be
# established.
RSA key fingerprint is a2:61:d0:f8:1a:13:f7:71:51:26:b8:c2:5f:6f:00:97.
Are you sure you want to continue connecting (yes/no)?
```

Kerberos and SSH

```
# kinit h7788999
# ssh -K h7788999@pecuchet
# klist

-K .... Enables forwarding (delegation) of GSSAPI credentials to the serve


a .5login files enables passwordless login:

e.g. /root/.k5login

user1@WU-WIEN.AC.AT
user2@WU-WIEN.AC.AT
```

Man in the Middle

```
# ssh  irgendwohin

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now
(man-in-the-middle attack)!
It is also possible that the RSA host key has just been
changed. The fingerprint for the RSA key sent by the
remote host is
90:9c:46:ab:03:1d:30:2c:5c:87:c5:c7:d9:13:5d:7
```

Copy Files with SSH

```
# echo bla >bla.txt
# scp bla.txt  anna@example.com:
# scp jemand@irgendwo.at:bla .
# scp -r diplomarbeit/ h7788999@login:
# sftp
```

SSH Configuration

- .ssh/config

```
Host wu
HostName pecuchet.wu-wien.ac.at
User h7788999
```

- .ssh/known_hosts
- .ssh/authorized_keys
- Sitewide: see /etc/ssh/sshd_config and /etc/ssh/ssh_config

Public Key (RSA/DSA) Login

```
# ssh-keygen (eventuell -f)

(choose a good passphrase)

# ssh-copy-id  hansi@meinserver.at

or do it manually:

# scp .ssh/id_rsa.pub  wu:
# cat id_rsa.pub >>.ssh/authorized_keys

you can have your keys in any file e.g:
```

Remote Command Execution

```
# ssh hans@meinserver.at  "ls -l /tmp"

echo bla | ssh hans@meinserver.at \
"cat - > bla.txt"

ssh hans@meinserver.at "ls -l /tmp" | grep bla
```

Usefull for scripts...

Limit the rights of a key

In the file .ssh/authorized_keys

```
from="137.208.77.7",no-pty,no-port-forwarding,
  command="/root/bla.sh" ssh-dss AAAUH7T9Y....
```

X11 Forwarding

```
# ssh -X user@woanders.at
# echo $DISPLAY
# localhost:10.0
# xterm &
# xauth list
```

Beware: A remote attacker might be able to spy on you. You have to trust the remote host in this case.

ssh agent

ssh-agent can cache the access to your private key.

```
# ssh-agent xterm &
# ssh-add

... type passphrase ...
```

Usually included in the graphical login.
SSH option a allows forwarding of access to the ssh agent.

Port Forwarding

```
# ssh  -L 3333:proxy.wu-wien.ac.at:8080 h778899@login

allows access to remote proxy on local port 3333

optional: use -g


# ssh -R 4567:localhost:80 h778899@login

allow a remote user to conntect to port
4557 to access your local server
```

Built in Socks Proxy

```
# ssh -D 9999 hans@woanders.at
```

GUI access to Files

Use URL like `sftp://hans@remote.at:bla/blah/`