

Networking in Linux

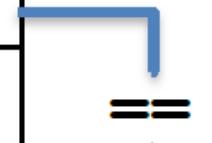
Terminology

- Network, subnetwork, addressing
- Bridging, Routing, Bonding, Tunneling
- VLAN (802.1Q), VPN
- Address translation NAT / PAT
- DNS, DHCP
- IPv4, IPv6, TCP, UDP, ICMP,
- Firewall (iptables)

Addressing

- Public IPs – RIPE (RIR, LIR, PI)
- Private IPs – for local use only (network maintained by one subject)
 - 10.0.0.0/8 10.0.0.0 - 10.255.255.255
 - 172.16.0.0/12 172.16.0.0 - 172.31.255.255
 - 192.168.0.0/16 192.168.0.0 - 192.168.255.255
- MultiCast IP's
 - 224.0.0.0/12 224.0.0.0 - 224.31.255.255
- Special
 - 127.0.0.0/8, 169.254.0.0/16, ...

Network, Mask, Broadcast (1/2)

Same subnet		
192.168.0.15 255.255.255.0	& 11000000.10101000.00000000.00001111 11111111.11111111.11111111.00000000 11000000.10101000.00000000.00000000	
192.168.0.31 255.255.255.0	& 11000000.10101000.00000000.00011111 11111111.11111111.11111111.00000000 11000000.10101000.00000000.00000000	

Different subnets		
192.168.0.15 255.255.255.0	& 11000000.10101000.00000000.00001111 11111111.11111111.11111111.00000000 11000000.10101000.00000000.00000000	
192.168.15.15 255.255.255.0	& 11000000.10101000.00001111.00011111 11111111.11111111.11111111.00000000 11000000.10101000.00001111.00000000	

Network, Mask, Broadcast (2/2)

- eg. 192.168.12.0/24
- Network Address
 - First IP address (not assignable) 192.168.12.0
- Network Mask
 - bit subnetwork mask 255.255.255.0
- Broadcast Address
 - Last IP address (not assignable) 192.168.12.255
- Network: 192.168.12.64/28 ?

N: 192.168.12.64, 255.255.255.240, 192.168.12.79

Usable addresses: 192.168.12.65 - 192.168.12.78

„ip“ - command (1/6)

- ip
 - ip help
- ip link add | link | delete | set | show
 - ip link help
- ip addr add | change | replace | delete | show
- ip route add | del | list | flush |change |
- ip rule add | del | list | flush
- ip tunnel add | del | change | list | show | ...
- ip

„ip“ - command (2/6)

- Renaming interface, eg. eth0 → eth2
 - ip link set eth0 name eth2
- Mac address setting
 - ip link set eth0 address 00:AF:22:CD:01:FF
 - ifconfig eth0 hw ether 00:AF:22:CD:01:FF
- Creating new VLAN
 - ip link add link eth0 name eth0.10 type vlan id 10
 - vconfig add eth0 10
- Start or shutdown of interface
 - ip link set eth0 up [down]
 - ifconfig eth0 up [down]

„ip“ - command (3/6)

- Assigning addresses to interface
 - ip addr add 192.168.10.254/24 dev eth0 brd +
 - ip addr add 192.168.20.254/24 dev eth0 brd +
 - Ifconfig eth0 192.168.10.254/24
 - Ifconfig eth0:1 192.168.20.254/24
- Removing addresses from interface
 - ip addr del 192.168.10.254/24 dev eth0
 - ifconfig eth0 del 192.168.10.254
- List IP addresses on all interfaces
 - ip addr show
 - ifconfig (does'n show secondary IPs cfg. by „ip“ cmd!)

„ip“ - command (4/6)

- Routes
 - ip route list | show
 - route
- Adding new route
 - ip route add 10.0.0.0/8 via 192.168.0.1 [dev eth0]
 - route add -net 10.0.0.0/8 gw 192.168.0.1
 - route add 192.168.10.12 gw 192.168.0.1
- Deleting route
 - Ip route del 10.0.0.0/8 via 192.168.0.1 [dev eth0]
 - route del -net 10.0.0.0/8 gw 192.168.0.1

„ip“ - command (5/6)

- Multiple Routing Tables
 - CONFIG_IP_MULTIPLE_TABLES=y (compiled in KERNEL)
 - /etc/iproute2/rt_tables - RT tables definitions
- Add new RT
 - echo "100 MyRT" >> /etc/iproute2/rt_tables
- Rules & Routes
 - ip rule add from 192.168.40.0/24 table Main (default)
 - ip rule add from 192.168.80.0/24 table MyRT
 - ip route add 192.168.80.0/24 dev eth4 table MyRT
 - ip route add default via 192.168.0.1 dev eth3 table MyRT

„ip“ - command (6/6)

- Tunnel interface – local (192.168.0.20)
 - ip tunnel add tun0 mode gre remote 192.168.0.10 / local 192.168.0.20
 - ip addr add 10.0.1.1/24 dev tun0
 - ip link set tun0 up
 - ip route add 192.168.0.0/24 dev tun0
- Tunnel interface – remote (192.168.0.10)
 - ip tunnel add tun0 mode gre remote 192.168.0.20 / local 192.168.0.10
 - ...

DNS - resolver

- Is it necessary?
 - Yes – if you know only DNS names: eg. google.com
 - Yes – Apache virtualhosting allows more than one domain per 1 IP address. Domains are accessible only by their domain names.
- Config
 - Edit /etc/resolv.conf
 - Add nameserver – add line: nameserver 8.8.8.8
 - Host domain search – add line: search fel.cvut.cz
www is than equal to **www.fel.cvut.cz**

DHCP - client

- DHCP
 - Dynamic Host Configuration Protocol
 - IP addressing
 - Routing (default gw, routes)
 - Host a domainname configuration
- Manually by command
 - dhclient eth2 - configures eth2 IF by DHCP server
- Automaticly after startup
 - edit /etc/network/interfaces
add line: allow-hotplug eth2
add line: iface eth2 inet dhcp

Linux Bridging – Layer 2

- bridge-utils package

- brctl command
- brctl addbr | delbr | addif | delif | show | ...

brctl addbr br0

- create bridge br0

brctl addif br0 eth1

- assign IF eth1 to bridge br0

brctl addif br0 tun0

- assign IF tun0 to bridge br0

brctl stp br0 on (if you want enable STP on br0)

ip addr add 10.0.0.1/24 dev br0

ip link set br0 up

Linux Routing – Layer 3

- Enable routing
 - echo '1' >/proc/sys/net/ipv4/ip_forward
 - sysctl -w net.ipv4.ip_forward="1"
- Enable routing permanently
 - edit /etc/sysctl.conf (add line: net.ipv4.ip_forward="1")
- Address translation – NAT / PAT
 - iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
 - iptables -t nat -A PREROUTING -i eth1 -d 1.2.3.4 -j DNAT – to 192.168.0.1

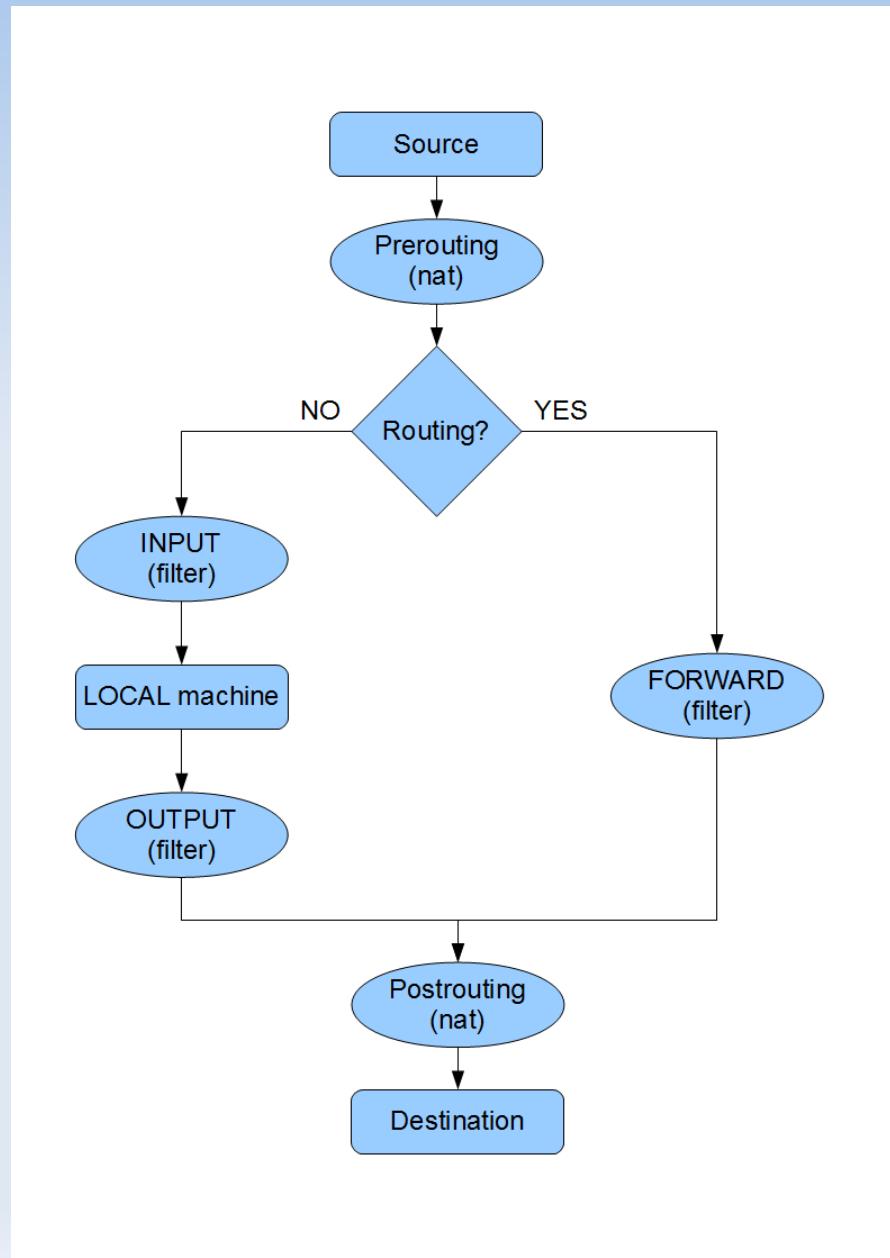
Linux network security

- TCP wrappers
 - /etc/hosts.allow eg.(sshd: 192.168.0.0/255.255.0.0)
 - /etc/hosts.deny eg.(ALL:ALL)
- Firewall
 - iptables – stateless

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```
 - iptables – statefull

```
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

Iptables – simplified diagram



iptables

- Default chains
 - INPUT, FORWARD, OUTPUT
- Tables
 - filter, nat, mangle, raw
- Actions
 - ACCEPT, REJECT, DROP, ...
- Default Policies
 - iptables -P INPUT DROP
 - iptables -P OUTPUT ACCEPT

Iptables rule parameters

- Iptables
 - -I Insert, -A Apped, -D Delete
- Iptables -I INPUT
 - -i input interface, -o output interface
 - -p protocol, -s source, -d destination
 - --sport source port, --dport destination port
 - -j jump target (Action)
- Iptables -I INPUT -p tcp –dport 22 -j
 - DROP – drop packet, ACCEPT – accept packet
 - REJECT – drop packet and send ICMP reply (rejected), ...
 - ...

iptables example

```
iptables -A OUTPUT -i lo -j ACCEPT
```

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

```
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

```
iptables -A INPUT -m state --state RELATED,ESTABLISHED  
-j ACCEPT
```

```
iptables -A INPUT -j LOG
```

```
iptables -P INPUT DROP
```

```
iptables -P FORWARD DROP
```

```
iptables -P OUTPUT ACCEPT
```

Utilities

- **tcpdump**
 - can catch all traffic incoming on interface
 - works in promiscuite mode
 - `tcpdum -i eth2` - all traffic on interf
 - `tcpdump -i eth2 host 192.168.0.10` - host traffic only
 - `tcpdump -i eth2 ether host 00:AF:12:14:CA:FF` – traffic of host with specific MAC address
- **netstat**
 - prints network connections, RT, multicast membership, ...
`netstat -l` - shows listening sockets
`netstat -g` - shows group membership

...

Excercises

- Setup network interfaces on PCs
 - 1st PC: eth0 10.0.1.1/24, 10.0.2.10/24
 - 2nd PC: eth0 10.0.2.1/24, 10.0.3.10/24
 - 3rd PC: eth0 10.0.3.1/24, 10.0.4.10/24
 - 4th PC: eth0 10.0.4.1/24, 10.0.5.10/24
- Setup routing through several computers
 - Try all to all PC reachability
 - from 1st PC ping 10.0.5.10
 - from 4th PC ping 10.0.1.1
- Modification
 - For 1 way communication (internet access) eg. 4th PC to 1st PC you can use address translation instead of setting routes
 - Default routes are still necessary!

Excercises

- Setup Firewall to prevent sshd access from other computers
 - enable ssh network access only for 10.0.1.0/24 network
- Setup firewall to enable ICMP probes for reachability testing
 - enable ICMP probes for everyone to reach you PC
 - enable pass of ICMP packets when routing packets
- Setup firewall to drop all other traffic
 - Set INPUT and FORWARD policy to DROP