Linux Network Administration

© 2016–05–03 Martin Bruchanov, bruchy@gmail.com

Cheat-sheet under construction!

Open Systems Interconnection model (OSI model)

7. Application – Data generation (SMTP, NNTP, SSH, Telnet, HTTP)
6. Presentation – Encryption and formating (JPEG, ASCII, EBDIC, GIF,...)
5. Session – Sync. & send to ports (RPC, SQL, NFS, NetBIOS)
4. Transport – TCP/UDP, message segmentation, message traffic control
3. Network – Packets, IP addr., routing, subnet traffic (IPv4/6, ICMP)
2. Data Link – Frame traffic control, sequencing (ARP, MAC)
1. Physical – cables, hubs, physical medium transmission
'**P**eople **D**on't **N**eed **T**hose **S**tupid **P**ackets **A**nymore!'

2. Internet Protocol (IP) Addresses

2.1. IPv4 addresses and mask

CIDR Notation:	192.168.1.130/25	
IPv4 (32bit):	192.168.1.130	11000000.10101000.00000001.10000010
Mask:	255.255.255.128	11111111.11111111.11111111.10000000
Subnet:	(IP and Mask)	11000000.10101000.00000001.10000000
Subnet:	192.168.1.128	
Usable Host Range:	192.168.1.129–254	
Broadcast Address:	192.168.1.255	

Use: `ipcalc`, `sipcalc` for IP/net calculations.

2.2. IPv6 addresses and mask

- IPv6 – `y : y : y : y : y : y : y : y`
- IPv6 with IPv4 part – `y : y : y : y : y : y : x . x . x . x`
- IPv6 – `y ::`
- Loopback: `127.0.0.1/8`; `::1/128`
- Unspecified address: `0.0.0.0/8`; `::/128`
- Multicast: `224.0.0.0/4`; `ff00::/8`
- Private: `10.0.0.0/8`, `172.16.0.0/12`, `192.168.0.0/16`; `fc00::/7`
- Automatic Private IP: `169.254.0.0/16`
- IPv4 mapped addresses: `::ffff:0:0/96` (`::ffff:0.0.0.0` – `::ffff:255.255.255.255`)
- IPv4/IPv6 translation: `64:ff9b::/96`
- For documentation examples: `192.0.2.0/24`, `198.51.100.0/24`, `203.0.113.0/24`; `2001:db8::/32`

2.3. Most common ports (`/etc/services`)

Privilege port < 1024 can be opened just by root user!

- 20, 21 FTP (File Transfer Protocol)
- 22 SSH (Secure Shell)
- 23 Telnet
- 25 SMTP
- 42 WINS
- 53 DNS
- 135-139, 445 Windows file sharing, login, RPC
- 80, 8080 HTTP (Hypertext Transfer Protocol)
- 88 Kerberos
- 110 POP3
- 111 Portmapper - Linux
- 119 NNTP (Network News Transfer Protocol)
- 123 NTP (Network Time Protocol)
- 135 RPC-DCOM
- 139 SMB
- 143 IMAP
- 161, 162 SNMP
- 389 LDAP
- 443 HTTPS (HTTP Secure)
- 445 CIFS
- 514 Syslog
- 636 Secure LDAP
- 1080 Socks5
- 1194 OpenVPN
- 1241 Nessus Server
- 1433, 1434 SQL Server
- 1494, 2598 Citrix Applications
- 1521 Oracle Listener
- 2512, 2513 Citrix Management
- 3389 RDP
- 5432 PostreSQL
- 6662–6667 IRC

3. Basic network setup

- Manage networking:
  - SysV Init script: `service network start/stop/restart`, `/etc/init.d/network start/stop/restart`
  - Systemd: `servicectl start/stop/restart NetworkManager.service`
- Set hostname:
  - `hostname` *`name`*
  - `nmcli general hostname` *`name`*
  - edit file `/etc/hostname`;
  - `hostnamectl set-hostname` *`name`*
- Check if physical link exits: `ethtool eth0`
- Loop-back interface: `ifconfig lo 127.0.0.1`
- Loop-back route: `route add 127.0.0.1`
- List devices: `cat /proc/net/dev`
- Show devices and configuration: `ifconfig`; `ip addr show`; `ip link show`
- Disable device: `ifconfig eth0 down`; `ip link set eth0 down`; `nmcli connection down eth0`
- Rename device (when disabled): `ip link set` *`enp0s25`* `name` *`eth0`*
- Enable device: `ifconfig eth0 up`; `ip link set eth0 up`; `nmcli connection up eth0`
- Set IP address:
  - `ifconfig eth0 192.168.0.1`; `ip addr add 192.168.0.1 dev eth0`
  - `ifconfig eth0 192.168.0.1 netmask 255.255.255.0 broadcast 192.168.0.255`
  - `ip addr add 192.168.0.1/24 broadcast 192.168.0.255 dev eth0`
  - `nmcli con add con-name eno2 type ethernet ifname eno2 ip4 192.168.0.5/24 gw4 192.168.0.254`
  - `dhclient -v eth0`
- Delete IP address: `ip addr del 192.168.0.1/24 dev eth0`
- Add alias interface: `ifconfig eth0:1 10.0.0.1/8`; `ip addr add 10.0.0.1/8 dev eth0 label eth0:1`
- Set promiscuous mode: `ifconfig eth0 promisc` (`-promisc` to disable); `ip link set eth0 promisc on/off`
- Change MAC address: `ifconfig eth0 hw ether AA:BB:CC:DD:EE:FF`; `ip link set dev eth0 address AA:BB:CC:DD:EE:FF`
- Default gateway:
  - `route add default gw 192.168.1.1 eth0`;
  - `ip route add 192.168.1.0/24 dev eth0`
  - `ip route add 192.168.1.0/24 via 192.168.1.1`
- Kernel network parameters: `sysctl -a | grep net`

3.1. Wi-Fi Networking

- Scan available networks: `iwlist wlan0 scan`; `nmcli dev wifi`
- Display available channels: `iwlist wlan0 freq`
- Connect with WEP network: `iwconfig wlan0 essid` *`"Network SSID"`* `key` *`HEX_KEY`*
- Connect with WEP network: `iwconfig wlan0 essid` *`"Network SSID"`* `key s:`*`ASCII_KEY`*
- Connect with WEP network: `nmcli dev wifi connect` *`"Network SSID"`* `password '123...'`
- Connect with WPA: `wpa_supplicant -B -i wlan0 -DWext -c /etc/wpa_supplicant.conf`
- Examples of WPA configuration: `man wpa_supplicant.conf`
- Watch signal quality: `watch -n 1 cat /proc/net/wireless` (link = SNR, level in dBm)

3.2. Configuration files of network interface settings

Stored in: `/etc/sysconfig/network-scripts/`*`ifcfg-inteface`*

Static	Dynamic	Either
BOOTPROTO=none IPADDR=192.168.0.2 PREFIX0=24 GATEWAY0=192.168.0.1 DEFROUTE=yes DNS1=8.8.8.8	BOOTPROTO=dhcp	DEVICE=eth0 NAME="System eth0" ONBOOT=yes UUID=a1b1c122-2... USERCTL=yes

3.3. NetworkManager, nmcli, nmtui

- Text user interface for NetworkManager: `nmtui`
- Manage service: `systemctl enable/disable/start/restart/stop NetworkManager.service`
- List all devices: `nmcli dev status`
- List all connections: `nmcli connection show`
- Show detail about connection: `nmcli con show eth0`
- Add connection: `nmcli con add con-name "default" type ethernet ifname eth0`
- Set IPv4: `nmcli con add con-name "static" ifname eth0 autoconnect no type ethernet ip4 172.125.X.10/24 gw 172.25.X.254`
- Set IPv4: `nmcli connection modify eth0 ipv4.addresses 10.0.0.2/8 ipv4.gateway 10.0.0.1`
- Activate/deactivate connection: `nmcli con up/down "static"`
- Reload configuration: `nmcli con reload`
- Bring down interface and disable autoconnect: `nmcli dev dis` *`DEV`*
- Disable all managed interfaces: `nmcli net off`
- Add, modify, delete connection: `nmcli con add / mod "ID" / del "ID"`
- Set DNS: `nmcli con modify eth0 ipv4.dns "8.8.8.8,8.8.4.4"`
- Set routes: `nmcli connection modify eth0 ipv4.routes "192.168.0.0/24 10.0.0.1, 192.168.1.0/24 10.0.0.1"`

3.4. DHCP (Dynamic Host Configuration Protocol)

- Configure device: `dhclient -v eth0`
- Release device configuration: `dhclient -r`
- DHCP client data: `/var/lib/dhclient/dhclient.leases`

4. Network socket of processes

- List active connections: `netstat -plunt`; `lsof -i`; `ss -tua`
- List process communication on port: `lsof -i :22` / `lsof -i :ssh`
- Check PID binded on local port: `ss -lt`; `fuser -n tcp 22`
- Monitor net. communication of single process: `strace -f -e trace=network -s 10000 -p` *`PID`*

5. ICMP (Internet Control Message Protocol)

- For IPv6 use: `ping6`, `tracepath6`, `traceroute6`
- Ping n-times: `ping -c` *`n`* `IP`
- Broadcast: `ping -b 10.0.0.255`
- Use different interface: `ping -I` *`eth1`*
- Trace route: `traceroute` *`host`*; `mtr -c 1 -r` *`host`*;
- Use TCP instead: `tcptraceroute`, `tcping` *`host port`*

6. Ethernet Bridge Manipulation

- Shows all current instances of the ethernet bridge: `brctl show`
- Create bridge *br0*: `brctl addbr br0`, `nmcli con add type bridge ifname br0`

7. ARP (Address Resolution Protocol)

- Show ARP table: `arp`; `ip neighbor list`; `cat /proc/net/arp`
- Clean ARP table: `ip -s neigh flush all`
- Add an entry in your ARP table:
  - `arp -i eth0 -s 192.168.0.1 00:11:22:33:44:55`
  - `ip neigh add 192.168.0.1 lladdr 00:11:22:33:44:55 nud permanent dev eth0`
- Switch ARP resolution off on one device: `ifconfig -arp eth0`; `ip link set dev eth0 arp off`
- Delete entry in interface: `arp -i eth1 -d 10.0.0.1`
- arpping -I interface -c count destination

8. Routing

- Display routes: `ip route show`, `ip route list`, `netstat -rn`
- Set default gateway: `ip route add default via 192.168.1.1`, `route add default gw 192.168.1.1`
- Print host interfaces and routes: `nmap --iflist`
- Route IP range through eth0: `ip route add 192.168.1.0/24 dev eth0`
- Delete route: `ip route delete 192.168.1.0/24 dev eth0`
- Enable IP forwarding:
  - `echo "1" > /proc/sys/net/ipv4/ip_forward`
  - Save in `/etc/sysctl.conf` option `net.ipv4.ip_forward = 1`
- Static route configuration: `/etc/sysconfig/network-scripts/route-eth0`:
  - `default via 10.254.0.1 dev eth0`
  - `172.31.0.0/16 via 10.254.0.1 dev eth0`

9. Firewall

9.1. IPv4/IPv6 packet filtering and NAT – `iptables`

- For IPv6 use: `ip6tables`
- Print all rules: `iptables -S` iptables [tabulka] [akce] [řetězec] [pravidla] [cíl] iptables -L – vypiš pravidla iptables -L FORWARD – iptables -A input -p tcp -dport N -j ACCEPT iptables -A input -p tcp -dport N -s IP/mask -j ACCEPT
- Enable SSH: `iptables -A INPUT -m tcp -p tcp --dport 22 -j ACCEPT`
- Enable SSH, HTTP, HTTPS: `iptables -A INPUT -p tcp -m state --state NEW -m multiport --dports ssh,http,https -j ACCEPT`
- Save iptables: `iptables-save > /etc/sysconfig/iptables`
- Network Address Translation (NAT) / Masquarage: `iptables -t nat -A POSTROUTING -s 10.200.0.0/24 -o eth0 -j MASQUERADE`

9.2. Dynamic Firewall Manager – `firewalld`

- Check status: `firewall-cmd --state`, `systemctl status firewalld`
- Print all rules: `firewall-cmd --list-all`
- List zones: `firewall-cmd --get-active-zones`, `firewall-cmd --get-zones`
- Get or set default zone: `firewall-cmd --get-default-zone`, `--set-default-zone=`*`ZONE`*
- Set default zone: `firewall-cmd --set-default-zone=`*`ZONE`*
- Without `--permament` option any changes will not be available after restart.
- Open TCP port in zone: `firewall-cmd --permanent --zone=`*`ZONE`* `--add-port=8080/tcp`
- Enable services: `firewall-cmd --permanent --add-service=http,https`
- Activate changes in configuration: `firewall-cmd --reload`
- Disable: `--remove-port=`*`port/protocol`*, `--remove-service=`*`service`*, `--remove-source=`*`X.X.X.X/Y`*
- firewall-cmd –zone=external –add-masquerade
- Forward packets to different IP and port: `firewall-cmd --zone=external --add-forward-port=port=22:proto=tcp:toport=2055:toaddr=192.0.2.55`
- Network Address Translation (NAT) / Masquarage: `iptables -t nat -A POSTROUTING -s 10.200.0.0/24 -o eth0 -j MASQUERADE`
- Rich language examples:
  - `firewall-cmd --permanent --add-rich-rule='rule family=ipv4 source address=172.25.X.10/32 service name="http" log level=notice prefix="NEW HTTP " limit value="3/s" accept '`
  - `firewall-cmd --permanent --add-rich-rule 'rule family=ipv4 source address=10.0.0.1/32 forward-port port=443 protocol=tcp to-port=22'`

10. Traffic monitoring

10.1. tcpdump

- Display communication with HTTP: `tcpdump -i eth0 'tcp port 80'`
- Communication with HTTP, print all ASCII, truncate packet content to 1024 bytes: `tcpdump -vvv -s 1024 -l -A 'tcp port http'`
- Display all communication except SSH: `tcpdump -i eth0 'not port ssh'`
- Display frames at the data link layer: `tcpdump -e`
- Don't convert host addresses / ports to name: `tcpdump -n / -nn`
- Hexdump headers and data of each packet: `-X`, and header `-XX`
- Monitor source: `tcpdump -i eth0 src 192.168.10.1`
- Monitor destination: `tcpdump -i eth0 dst 192.168.10.1`
- Monitor network: `tcpdump -i eth0 net 192.168.10.1/24`
- DNS packets: `tcpdump udp and src port 53`
- Capture communication on `eth1` to file: `tcpdump -ni eth1 -w file.cap`
- Capture telnet and ssh: `tcpdump -n portrange 22-23`
- `tcpdump -nnvvS src 10.0.0.5 and dst port 3389`
- Check packet filter syntax: `man pcap-filter`

11. Remote shells

11.1. Secure SHell (SSH)

- Connect: `ssh -l `*`login`* `-p `*`port hostname`*, `ssh `*`login@hostname`*
- Escape character sequences, press Enter, then '`~`' followed by a command:
  - `?` – Display a list of escape characters.
  - `.` – Terminate connection.
  - `Ctrl-z` – suspend ssh process, use `fg` to enable it again.
  - `B` – send a BREAK to the remote system.
  - `C` – open a command line (use `help`) for port forwarding options.
- Local port transfer – remote port will be available locally
  - `ssh -L localport:remoteIP:remoteport host`
  - `ssh -L localIP:localport:remoteIP:remoteport host`
- Remote port transfer – local port will be available on remote
  - `ssh -R remoteport:localIP:localport host`
  - `ssh -R remoteIP:remoteport:localIP:localport host`
- Dynamic port transfer – creation of SOCKS proxy:
  - `ssh -D [LocalAddress:]LocalPort host`
  - Use LocalAddress:LocaPort as SOCKS proxy and all request will be forwarded throught host.
  - `curl --user-agent "Mozilla" --socks4 localhost:1080 http://www.whatsmyip.org/`
- Remote filesystem: `sshfs -o allow_other,defer_permissions,IdentityFile=/.ssh/id_rsa user@xxx.xxx.xxx.xxx:/ /mnt/droplet`
- Copy remote stdout to your X11 buffer: `ssh user@host 'cat /path/to/some/file' | xclip`

11.1.1. SSH key handling

- Genereate 4096bit key with comment: `ssh-keygen -t rsa -b 4096 -C "Top secret key"`
- Generate public key from private: `ssh-keygen -y -f private.pem > public.pub`
- Permissions: `chmod 700 /.ssh`; `chmod 600 /.ssh/authorized_keys`
- Copy key to host and updates `~/.authorized_keys`: `ssh-copy-id `*`user@host`*
- Holds SSH keys in memory for 8 hours: `ssh-agent -t $((8*3600))`
- Add key to agent: `ssh-add  /.ssh/id_rsa` (will ask for passphrase once in time life)
- Forward SSH agent: `ssh -A hostname`
- Connect to SSH *host* via *server*: `ssh -At `*`server`* `'ssh `*`host`*`'`

12. Remote desktop

- X11 SSH tunnel: `ssh -X `*`host`*, `ssh -Y `*`host`* (trusted)
- X11 redirection:
  - on remote, redirect display: `export DISPLAY=`*`YOUR_IP`*`:0.0`
  - on local, enable connection: `xhost +`*`REMOTE_IP`*
- Windows remote desktop: `rdesktop -u `*`USER`* `-d `*`DOMAIN`* `-g 1024x768 -r disk:local=  `*`hostname`*
- Other options: X2Go, VNC, NoMachine NX.

12.1. TELNET

- Connect: `telnet hostname port`
- Set login name: `telnet -l login hostname`
- Enter command mode: `Ctlr-]`

13. Remote file systems

13.1. Common Internet Filesystem (CIFS/SaMBa)

- `mount -t cifs '\\dc1\devel' /mnt/dc1 -o user=DOMAIN/USER`
- smbclient
- smbget
- smbmount smbumount
- smbclient -L localhost -N
- smbclient //localhost/<share> -U <uživatel>
- smbstatus

13.2. Network File System (NFS)

- User mush have same UID and GID on server and localhost.
- Server configuration stored in `/etc/export`:
  - Share directory with client IP: `/mnt/share 192.168.0.100(rw,sync,no_root_squash)`
  - `ro` read-only, `rw` read-write, `sync` , `no_root_squash` allow root, `no_subtree_check`
- List connected clients: `netstat | grep nfs`
- Remote check: `rpcinfo -s bee | grep -E 'nfs|mountd'`
- Show server's export list: `showmount -e`
- Mount remote directory: `mount -t nfs 192.168.0.99:/mnt/share /mnt/local`

14. File transfer

14.1. File transfer protocol (FTP)

14.2. Batch FTP transfer

14.3. rsync

- rsync
- Tunnel through SSH: `rsync -avHPS --rsh="ssh -p 2222" source user@host:/destination`
- -r recursive
- -l synchronize symlinks
- -p preserve permissions
- -t preserve timestamp
- -g, -o preserve group, owner
- -D synchronize device files

14.4. SCP/SFTP

- scp -P port
- scp user@host:file .
- scp file user@host:dir
- scp user1@host1:file user2@host2:dir
- -i identity file
- server.example.com#2000: – port
- scp -rp ./adresář host1:

15. Port scanning

- nmap -sV -n -sP -p -sT -sU – UDP port scan -sO – IP Protocol scan -sV – Detekuj služby a aplikace -F -O
- Scan IP range for open port, grepable output to stdout: nmap -p80 10.0.0.0/24 -oG -

16. netcat – Concatenate and redirect sockets

- netcat host port
- netcat -l -p port

17. Domain Name Service (DNS)

- Local names definition: `/etc/hosts`
- Source of name resolution: `/etc/nsswitch.conf`
- Resolver configuration file – `/etc/resolv.conf`:

```
nameserver 8.8.8.8
nameserver 8.8.4.4
```

`search`

- `host` *`name`* – look up the IP address nslookup getent – get entries from Name Service Switch libraries `getent hosts` *`hostname`* – test resolution with `/etc/hosts` hostnamectl status
- Return hostname for IP: `dig -x 10.32.1.10 +short`
- Return IP for *hostname*: `dig` *`hostname`* `+short`
- dig -t record type
  - A / AAAA – return 32/128 bit address for host
  - CNAME – aliases of hostname, can point to A
  - MX – mail exchanger record
  - NS – specify authoritative nameserver for domain
  - PTR – pointer records for reverse lookup (addr->host)
  - SOA – Start of Authority, name of the server that supplied the data for the zone
- User given DNS server: `dig @8.8.8.8 hostname`

dig @a.root-servers.net example.com gtld-servers.net dig +dnssec +multi @a.iana-servers.net example.com +cdflag

18. WHOIS service

- whois

19. HTTP(S) (Hypertext Transfer Protocol [SECURE])

- URL format: `http://user:password@domain:port/path?query#fragment_id`
- wget –referer –user-agent
- Mirror site: `wget -e robots=off -r -L http://`*`URL`*
- Display HTTP header: `curl -I`, `wget -S`
- Donwload file: curl -O URL
- Download URL and display it in stdout: `curl URL`
- Enable HTTP proxy in shell: `export http_proxy=http://foo:bar@202.54.1.1:3128/`
- Use the same for HTTPS: `export https_proxy=$http_proxy`
- Convert page to text: `elinks -dump` *`URL`*

20. OpenSSL

- Generate random sequences: `openssl rand -base64 8`
- Display server certificate: `openssl s_client -showcerts -connect google.com:443`

21. Network Time Protocol (NTP)

- ntpq
- ntpd
- ntpdate
- ntpdate -s time.nist.gov –
- NTP servers: `tik.cesnet.cz`, `tak.cesnet.cz`

22. Remote Procedure Call (RPC)

rpcinfo -p localhost

23. Internet daemon – inetd, xinetd

/etc/hosts.allow vsftpd: /etc/myftp.hosts
/etc/myftp.hosts 192.168.0.0/255.255.255.0
/etc/hosts.deny

24. Security Enhanced Linux (SELinux)

- List port mapping: `semanage port -l`
- Use 8000 for http: `semanage port -a -t http_port_t -p tcp 8000`
- Check status: `getenforce`
- Disable SELinux temporalily: `setenforce 0`
- Set directory accessible by httpd: `chcon -R -t httpd_sys_content_t ./directory`

25. Show/manipulate traffic control settings

tc

26. Virtual Private Network (OpenVPN)

- TUN device for IP traffic, TAP device for ethernet frames
- Enable UDP port 1194: `iptables -A INPUT -i eth0 -m state --state NEW -p udp --dport 1194 -j ACCEPT`, `firewall-cmd --permanent --add-service openvpn`
- Basic server: `openvpn --ifconfig 10.200.0.1 10.200.0.2 --dev tun`
- Basic client: `openvpn --ifconfig 10.200.0.2 10.200.0.1 --dev tun --remote your.openvpnserver.net`
- Use TCP protocol: `--proto tcp-server` (server), `--proto tcp-client` (client)
- Create/use static key: `openvpn --genkey --secret secret.key` and use `--secret secret.key` on client/server.