



Network commands and tools

Workshop



Requirements

You should have read and watched:

- Kernigan
 - Communications (pp 117-122)
 - Networking (pp 123-134)
 - The Internet (pp 135-160)
 - The World Wide Web (pp 161-186)

- Wiki:

[http://wiki.juneday.se/mediawiki/index.php/ITIC:Network tools and commands](http://wiki.juneday.se/mediawiki/index.php/ITIC:Network_tools_and_commands)

- Read the text and seen the video lectures
- Done the exercises (as far as you could)

Relevance

- Give you hands-on experience from using network commands and tools
- Prepare you for your assignments (personal homepage, bash script)
- A few basic network tools is something everyone studying or working with computers should know
- Even in your personal life, you might have use for this when dealing with network problems, setting up your home network etc

VirtualBox/Ubuntu versus your own OS

- If you are running Ubuntu on a VirtualBox instance, please try to find out both how your actual computer is networked and how Ubuntu on VirtualBox is networked
- You will have to use a search engine to figure out how e.g. Windows or macOS equivalents of the tools in this workshop work (and what they are)

Warm-up: Find out about your network config.

- Work in groups
- Figure out
 - how you are connected to the network
 - what the network name is
 - what the default gw is
 - what your IP address is
 - what your external IP address is
 - what DNS you are using
- If you cannot do it in VirtualBox/Ubuntu, do it on your computer
- Use a search engine to figure out the commands for you OS

Some basic commands

- ping
- traceroute, mtr
- host, nslookup, dig
- ip, ifconfig
- netstat, ss
- nc, telnet, ssh
- scp, rsync, wget, curl

If you are running Ubuntu in a VirtualBox

- Try to find out the same information in a terminal in your native OS (your default OS, the host)
- The following should work:
 - ping
 - traceroute (Windows, perhaps tracert)
 - macOS: ifconfig
 - windows ipconfig
 - netstat
 - You may need to install ssh, telnet, curl, wget

traceroute

```
$ traceroute www.sunet.se
```

```
traceroute to www.sunet.se (192.36.171.231), 30 hops max, 60 byte packets
 1  gw-9-129.chl.chalmers.se (129.16.219.129)  0.397 ms  0.352 ms  0.411 ms
 2  core2-chl-gw.chalmers.se (129.16.2.89)    0.702 ms  0.779 ms  0.859 ms
 3  cth-r2.sunet.se (130.242.6.10)  0.562 ms  0.548 ms  0.569 ms
 4  goteborg-gbg7-r2.sunet.se (130.242.4.176)  0.700 ms  0.721 ms  0.705 ms
 5  trollhattan-trh-r1.sunet.se (130.242.4.41)  1.628 ms  1.614 ms  1.631 ms
 6  karlstad-karl-r1.sunet.se (130.242.4.38)  3.842 ms  3.841 ms  3.764 ms
 7  orebro-lba-r1.sunet.se (130.242.4.29)  5.173 ms  5.121 ms  5.060 ms
 8  vasteras-fsn2-r1.sunet.se (130.242.4.30)  6.330 ms  6.304 ms  6.276 ms
 9  stockholm-tug-r1.sunet.se (130.242.4.35)  7.567 ms  7.546 ms  7.530 ms
10  stockholm-tug-r2.sunet.se (130.242.5.47)  7.551 ms  7.536 ms  7.522 ms
11  * * *
12  * * *
13  * * *
```


Start a simple server using nc (netcat)

- Netcat - nc - let's you start a simple server on your computer.
- In a terminal, do:

```
while true; do echo "Hej" | nc -l -p 8080; done
```
- Open a new terminal (without closing the first one)
- Do:

```
echo hej | nc localhost 8080
```
- What happens?

Start a simple server using nc (netcat)

```
rikard@newdelli:~/itid/bash$ while true;do echo "Hej" | nc -l -p 8080;done
Hej
apan
[]

rikard@newdelli:~$ echo hej | nc localhost 8080
Hej
rikard@newdelli:~$ echo "apan" | nc localhost 8080
Hej
rikard@newdelli:~$ []
```

Address already in use...

- Next, we'll see what happens if we try to open a new server on the same port, at the same time
- Without closing the server in the first terminal, do the following in the second terminal:

```
echo "Hej" | nc -l -p 8080
```
- What happens?

Address already in use...

```
rikard@newdelli:~/itid/bash$ while true;do echo "Hej" | nc  
-l -p 8080;done  
hej  
apan  
█
```

```
rikard@newdelli:~$ echo "Hej" | nc -l -p 8080  
nc: Address already in use  
rikard@newdelli:~$ █
```