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
  - 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
- Prepare you for the web page assignment

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

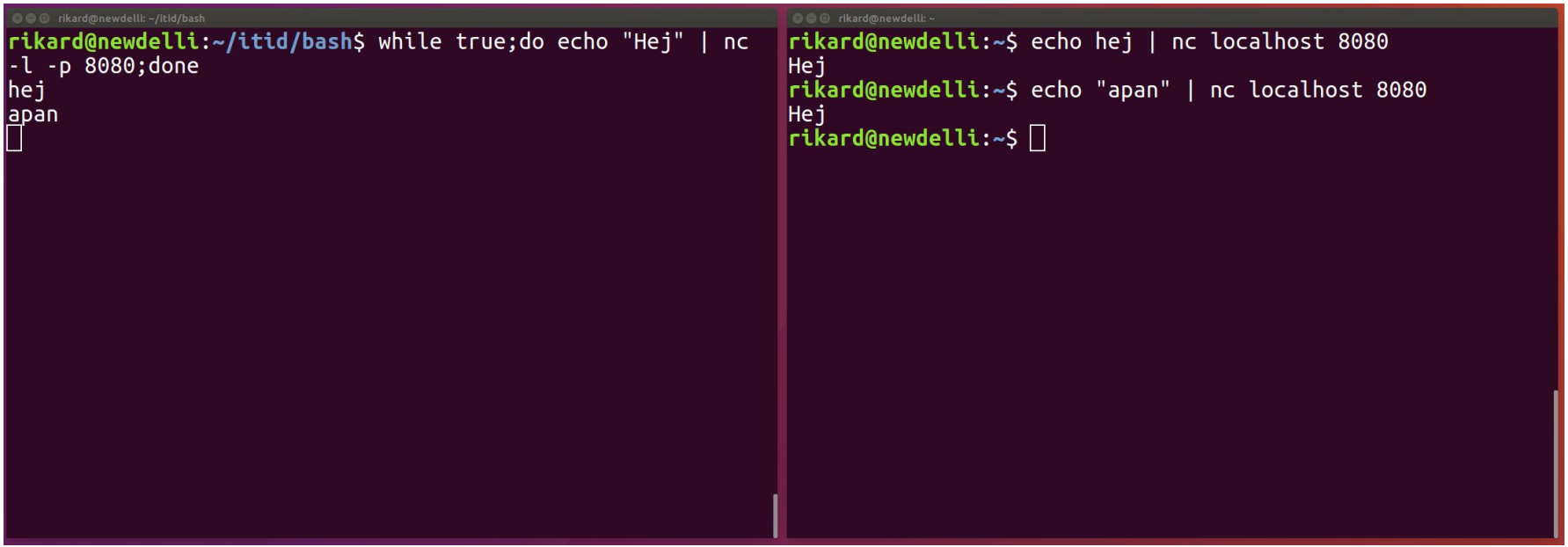
If you are running Ubuntu in a VirtualBox

- 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

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?

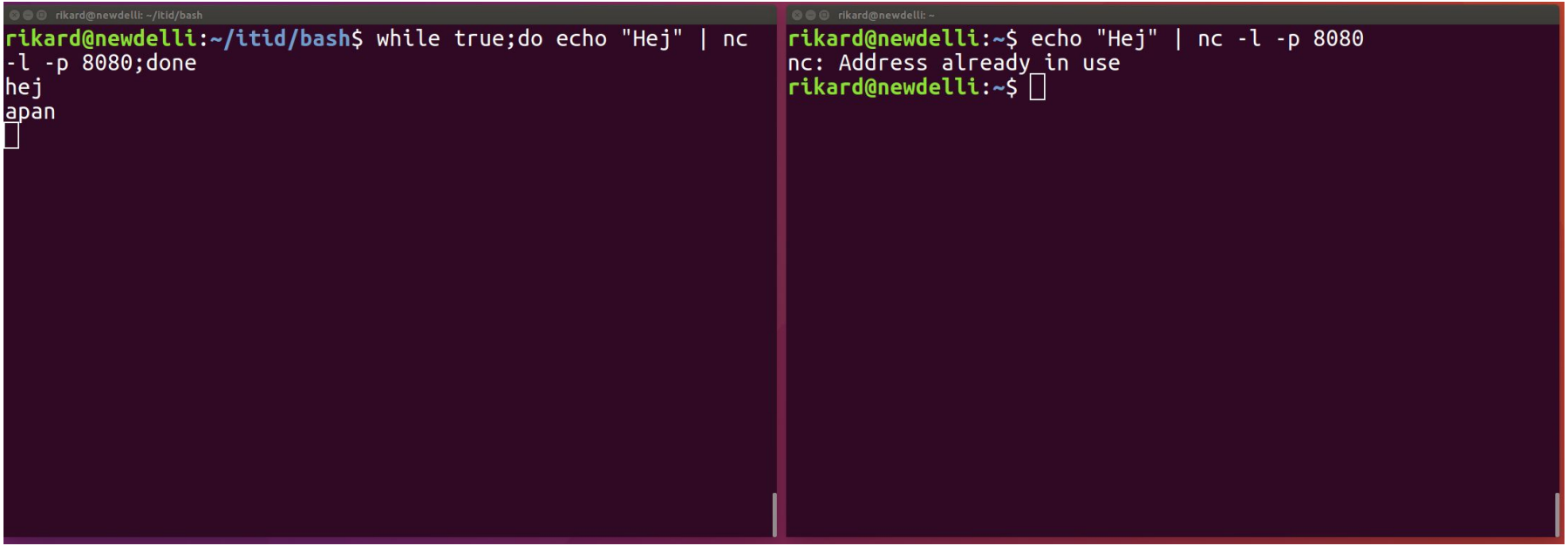
Start a simple server using nc (netcat)



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...



Using ssh

- Let’s log into another computer using ssh
  - You might need to install openssh-client - but it comes with Ubuntu by default
    ```
    $ sudo apt-get install openssh-client
    ```
- Get your username and password from the teachers for the lab computer
- Log in:
  ```
  $ ssh your-user-name@lab-server
  your-user-name@lab-server’s password:
  ```
- You should now be logged in
- Note: the teachers will provide your usernames, passwords and the address to the lab computer

Passwords are hard to remember

- It would be great if you could login to the lab-server without entering the password
- How could this be achieved in a secure way?
- Enter ssh-keys!

Generate a public-private key pair

```
ssh-keygen -t rsa -b 4096 -C "your_email@domain.com"
Enter file in which to save the key
(/home/yourusername/.ssh/id_rsa):                 just press Enter

Enter passphrase (empty for no passphrase): just press Enter
```

Copy the public key to the lab-server

```
$ ssh-copy-id your-user-name@lab-server
your-user-name@lab-server's password:          type your pw
```

Login to the lab-server and verify that it works

- You should now be able to login to the lab-server without entering your password
  - Do it!
- On the lab-server, do the following:
  `cat ~/.ssh/authorized_keys`
- What you see there, is your public key
- Logout again.

Write a small text file and copy it to the lab-server

- Create a small text file with some message of your choice and copy it to the lab-server (name the file about-yourname.txt - no spaces!)
- Use:

```
$ scp about-xx.txt your-user-name@lab-server:    # colon!
```

- How does the lab-server handle the fact that you are now uploading a heap of files?
  - In other words, how is it that you can upload files that won't interfere with your classmates' files?

Login and know your environment

- Login to the lab-server
- Find out what group you belong to
  hint: there's a command called `groups`
- Copy your text file to /var/www/your-group-name
- Try to copy your text file to /var/www/some-other-group-name
- Did it work?
  - Why/why not do you think it worked/didn't work?
- Download the text file to your own computer (open a new terminal) from http://lab-server/group-name/about-xx.txt

Create a file in `~/public_html`

- What text editors are available on the server?
- Use an editor to create the file `~/public_html/test.html`
- Put the following content in it:
  `<html><body>My name is Your Name</body></html>`
- Fire up a browser and navigate to http://lab-server/~username (your username)
- How do you think this is possible?
- List the contents of /var/www/
- Is your name there?
- If not, how do you think the web server handles this?

How do you buy and configure a domain?

- If there’s time - we’ll look at the jundeday loopia configuration
- Otherwise, that’s all, folks