Name:_____

Screen

Screen makes it possible to preserve a terminal session even though the terminal connection is cut.

1. Create a new session named "hakku": *screen -S hakku*
2. Start a text editor in the created session and create a file hakku.txt
3. Cut the connection to the session: *CTRL-a d*
4. Create another session named "kuokka" and start aptitude: *screen -S kuokka aptitude*
5. List all sessions the you have in the system: *screen -ls*
6. Cut the connection to "kuokka" and connect to "hakku"
7. Create another window into session "hakku": *CTRL-a c*
8. Connect "kuokka" to the second window of "hakku": *screen -r kuokka*
9. Close aptitude inside session "kuokka". That instance of screen will terminate, since the last window inside it closes.

NOTE: Don't open the screen inside of itself. Loops will explode.
NOTE2: Tmux is another screen-like program.

SSH

SSH is an encrypted remote terminal program.

1. Create a SSH connection to a server. For example: *ssh username@kosh.aalto.fi*
2. Now you can run console commands on the server instead of the local computer.
3. SSH+Screen is an extremely powerful combination that can maintain terminal work sessions when moving from computer to computer.

SSH client programs exist on most platforms:

- putty for windows
- openssh for linux/unix/osx
- JuiceSSH for android

SSH is a primary tool for configuring network servers.

Transferring files

SSH includes helper programs to transfer files:

- putty has pscp/psftp, on linux scp/sftp. GUI software filezilla, winscp.
- Transfer a text file from your linux computer to another computer.
- If your Linux computer runs in a virtual machine, transfer a text file to the host computer. What do the contents of the file look when you open it on the host computer? (For example, notepad on windows)

Tunneling

SSH can also "tunnel" graphical applications. This requires an X11 server on your computer. Linux has X11 built in, MacOSX needs to have it installed, and Windows needs some third-party software.

1. Make a ssh connection using parameter "-X". For example *ssh -X username@brute.aalto.fi*
2. Start a graphical application: *xclock*
3. The program will run on the computer you made the connection to, but the graphical user interface is rendered on your computer at the other end of the SSH tunnel.
4. This works for more complex applications also: *matlab*
5. The network connection does cause a lot of overhead, so complex or more graphically intense applications don't perform well.

SSH can also tunnel generic TCP connections. There are three types of TCP tunnels: local, remote and SOCKS5.

- Local tunnels mean that SSH listens to a local TCP port and relays the connections to a named target at the remote end.
- Remote tunnels work the opposite; SSH listens to a TCP port on remote end and relays the connections a named target at local end.
- SOCKS5 tunnels work so that local SOCKS5-compatible programs can initiate connections so that the remote end is the sourc of the connection.

One practical example for SSH tunneling is to avert firewall restrictions. An example of this kind of activity would be to use Aalto University resources from outside of Aalto networks by using a SSH/SOCKS5 tunnel to Kosh.aalto.fi.

1. Then the tunnel would be created as follows:
   ssh username@kosh.aalto.fi -D8080
2. Now set this as your web browsers proxy: *SOCKS5 = localhost:8080*
   (firefox: preferences/advanced/network/settings)
3. Check that it works using `http://whatismyipaddress.com`

Public key authentication

Basic SSH connections are authenticated using a password. The ownership of the password is proof that the user is the owner of the user account. Public key authentication is based on so called public and private keys. These are created as a pair and ownership of the private key is the proof that user is the owner of that public key.

1. Run command: *ssh-keygen -f testi*
2. This creates private key named "testi' and the corresponding public key "testi.pub". Public key can be freely copied to a server (knowing the public key does not grant you any power. The private key is the important one.)
3. You can copy your key to a server by manually copying the contents of "testi.pub" to your home directory on the server in the file ".ssh/authorized_keys". Or you can use a helper command: *ssh-copy-id -i testi username@server*
4. Now you can log into the server without password: *ssh -i testi username@server*

If you don't use parameter "-f" the public key is created as the default key of your profile and it will be used automatically (even without parameter "-i").
Public key authentication is often used automate administrative task using ssh connections; the admin software can use the public key to make connections without having a human present to enter the passwords.

Network diagnostics

The fundamental model of Ethernet and Wireless LAN edge networks is that "neighbors" (meaning hosts in the same network segment and IP address subnet) are directly reachable. Otherwise a host called "gateway" is used to reach the target. Gateway is a router that will communicate with other routers to deliver the messages.

Ping command

1. Investigate *ip addr*, what is your own IP address? Does this address respond to ping?
2. Investigate *ip route*, what is your gateway? Does it respond to ping?
3. Figure out "www.aalto.fi" server address using *nslookup* command.
4. Does "www.aalto.fi" respond to ping? What is the average round-trip time (rtt avg)? Is your result the same as with others?

Having hard time with IP addresses and subnets? There's a neat tool called *ipcalc*. Install it: (*apt-get install ipcalc*). Pick your IPv4 address from *ip addr* (after inet: "x.x.x.x/x") and enter it: *ipcalc x.x.x.x/x*.

Tracing network paths

1. What kind of route do you see with: *traceroute www.google.fi*? How about *traceroute www.aalto.fi*?
2. Test command *mtr* to same targets.

Capturing traffic

Next commands will require additional privileges so run tem as root or using sudo command.

1. While ping is running in one terminal, use another terminal to run *sudo tcpdump -i device icmp* (where device is the network device you're using)
2. Run *sudo tcpdump -i device tcp port 80* and load a page that uses plain http, for example: `http://www.aalto.fi/` (https uses different port, 443).

Wireshark is a graphical tool for network traffic capture and protocol analysis. You will probably be using these tools later on if you attend Comnet courses.