CSC

# Assignment 1
# Implement and analyse a virtual computer network

## Goals

- Implement and test a virtual computer network;
- Perform a TCP/IP packet analysis.

## 1 Introduction

Throughout this course, lab assignments make use of a computer network connecting several virtual machines. Virtualization is done through VirtualBox, while the virtual machines run Linux Lubuntu.

This first assignment requires you to recall some basic networking concepts. The goal is to install the aforementioned computer network and monitor its TCP/IP activity.

## 2 Installing a virtual machine

You should start by installing VirtualBox from www.virtualbox.org on your own machine. Should the one appropriated to your operating system. Then you should download from the course site the virtual machine "machine 1.ova", which is a preconfigured virtual machine with Linux Lubuntu operatings system. These sections details the required steps to instantiate a virtual machine based on the given virtual hard disk image.

### *2.1 The host system*

The host is the physical machine running the VirtualBox software, i.e. your own machine.

### *2.2 Creating a virtual machine*

Start VirtualBox and create the first machine by choosing "File->Import Appliance" and then choosing the file "machine 1.ova" that you have downloaded from the course homepage.

VERY IMPORTANT: When prompted choose to reinitialize MAC addresses

Start the machine by selecting it and pressing the start button. Log in with the user *user* and the password *inseguro*;

Currently the keyboard is set to model pc105 Language Portuguese. Your computer might have a different keyboard. To change the keyboard, press the Portuguese Flag in the bottom right corner with the right bottom of the mouse. Choose keyboard layout handler, and then change the keyboard model

and layout according with your requirements.

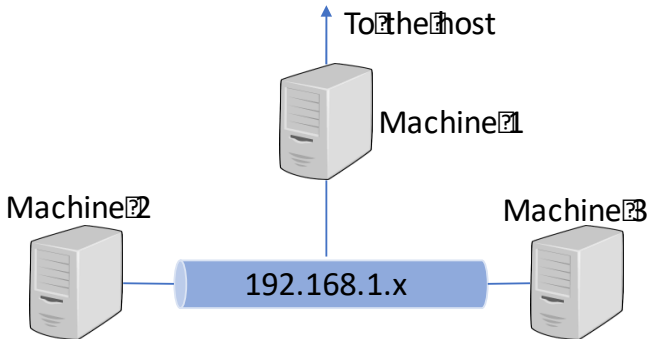
You may want to install VirtualBox Guest Additions for your operating system (this will allow you to copy and paste between the host and virtual machines.

- Open a terminal (press the green rectangle in the bottom of the window);
- cd /media/user/VBOXADDITIONS_XX (where XX is your specific version)
- sudo ./VBoxLinuxAdditions.run

Test the network by opening a terminal (press the green rectangle in the bottom of the window) and using the ping command (e.g. ping 8.8.8.8)

# 3 Create a virtual network

Implement the following virtual network by creating two additional virtual machines




To create each of the other machines:

1. Shutdown Machine 1
2. Select "Machine 1" and press "Machine->Clone;
3. Set the name to Machine 2 or Machine 3
4. Choose Full Clone if you have enough disk space (each machine takes around 8,5GB) or choose Link Clone to save space.

## *3.1 Configuring machine 1*

Notice that machine 1 has two network interfaces, one connected to the Host and another one connected to the subnet 192.168.1.X. You will need to shut down this virtual machine in order to add a second Network Adapter through the Settings panel in VirtualBox.

1. Shutdown Machine 1
2. Select Machine 1 and choose settings, then Network and Adapter 2
3. Check "Enable Network Adapter" and choose "Internal Network"
4. Under Advanced
    a. set "Promiscuous Mode" to "Allow All"
    b. refresh the MAC address by pressing the button at the end of the line

c. Write down this address. This will be the address of the Ethernet card connected to the internal network.

5. Start “Machine 1”
6. Configure the net by writing in a terminal

sudo nmcli con mod "Wired connection 2" connection.interface-name enp0s8 connection.autoconnect yes ipv4.addresses "192.168.1.1/24" ipv4.method "manual”

sudo nmcli con mod "Wired connection 1" connection.interface-name enp0s3 connection.autoconnect yes ipv4.method "auto”

With ifconfig check that enp0s8 is associated with the MAC address of the internal network (cf. step 4c). Confer that the other device name is named enp0s3.

This is setting the IP address of the second network connection to 192.168.1.1 with a netmask 255.255.255.0. The same may be achieved by pressing the network button (two rectangles linked by a line) on the bottom bar of the Linux Machine, and choose edit connections.

Confirm manually that changes were made with

ifconfig

Since machine 1 will be the default gateway for machines 2 and 3, *ip* forwarding must be enabled. This will allow machine 2 and 3 to communicate with machines outside the subnet 192.168.1.X.

- Open the file */etc/sysctl.d/sysctl.conf using*
  *sudo nano /etc/sysctl.d/99-sysctl.conf*
- and uncomment the line
  #net.ipv4.i_forward=1;
- Load the configurations into the *kernel*:
  sudo *service procps restart*
- Confirm that the flag value was updated to **1**:
  *sudo sysctl net.ipv4.ip_forward*

## *3.2 Configuring machines 2 and 3*

Select Machine 2, go to settings and disable adapter 2. For adapter 1 choose “Internal Network”.

Start the machine. Repeat the process for the other machine. Start a terminal on each machine and run:

sudo nmcli general hostname machine2

This will change your machine hostname.

Check that your network connection name is “Wired connection 1” or “Wired connection 2” by issuing the command

sudo nmcli

then issue accordingly

sudo nmcli con mod "Wired connection 1" ipv4.addresses 192.168.1.2/24 ipv4.method manual gw4 192.168.1.1 ipv4.dns 8.8.8.8

Confirm that changes were made using the commands

Ifconfig

ip route

Repeat the process for machine 3, replacing the underline 2 by 3. Moreover, under Settings-Network-Advanced, set "Promiscuous Mode" to "Allow All".

Use the ping command to test the connectivity between **all virtual machines**, as well as **with the host system** and Internet (e.g. you may ping google dns using ping 8.8.8.8)

### *3.3 Configuring machine 1 (NAT)*

During the previous step, the connectivity test between machines 2 or 3 with the host system and the Internet failed. Since NAT was not enabled in machine 1, the host sent the reply to its gateway, which eventually dropped the packet. Use the **iptables** command (man **iptables**) in machine 1 to correct this behaviour.

```
sudo iptables –P FORWARD ACCEPT
sudo iptables –F FORWARD

sudo iptables –t nat -F
sudo iptables –t nat –A POSTROUTING  –o enp0s3 –j MASQUERADE
```

To make this rules persistent do:

```
sudo sed -i "s/zesty/xenial/g" /etc/apt/sources.list
sudo apt update
sudo apt install iptables-persistent
sudo bash –c "iptables-save > /etc/iptables.rules"
```

## 4 Monitor network traffic

For this part of the lab we need to start a telnet service at machine 2. For that, you need to:

1) Execute the commands

```
sudo sed -i "s/zesty/xenial/g" /etc/apt/sources.list

sudo apt update –y

sudo apt install xinetd telnetd
```

2) Create a file named telnet in directory /etc/xinetd.d with the following content

```
# default: on
# description: The telnet server serves telnet sessions; it uses
```

```
# unencrypted username/password pairs for authentication.
service telnet
{
disable = no
flags = REUSE
socket_type = stream
wait = no
user = root
server = /usr/sbin/in.telnetd
log_on_failure += USERID
}
```

3) Execute the command

sudo /etc/init.d/xinetd restart

Use machine 3 to run **tcpdump** and capture all network traffic. Make sure you can detect ICMP packets originating at machine 1 and destined to machine 2 (using ping). Use tcpdump with options –X and –XX and identify the ip addresses, mac addresses and protocol in a given packet.

While still running sudo tcpdump –i enp0s3 –vvv –x –X (check the device name with ifconfig or nmcli), open a telnet connection between machines 1 and 2 using user *user* and password "*inseguro*". Verify that you can capture both the username and password with tcpdump. Mind that username and password characters appear in different packets. Try also the same thing using sudo wireshark.