“Analysing Network Traffic with Wireshark”

4th Edition. November 2023

1 Introduction

This practice will be done using a virtual machine where all tools needed for this laboratory are already installed. You can find this virtual machine in your computer at the Telematics department laboratory. You can also find the image in the following URL (using your UC3M account):

https://drive.google.com/file/d/10m6vm_prA2fIrssuHyeH0eIPgKKxOqcr/view?usp=drive_link

Alternative for ARM Mac users (M1, M2 o M3)
https://drive.google.com/drive/folders/1N4XcrSxp1geBwCPh0brlllNczZrkPx9x?usp=drive_link

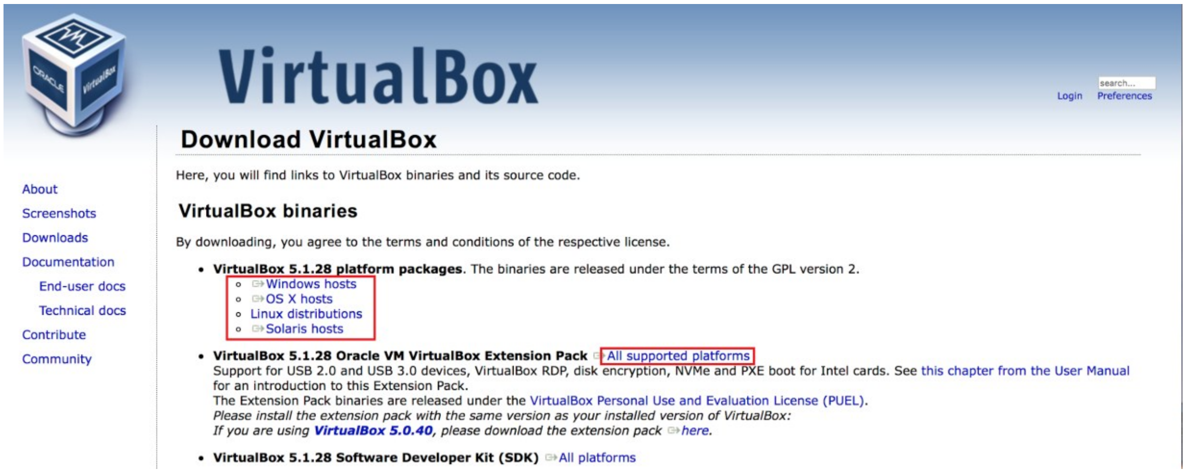
You can install the above image in your own computer. It is necessary to have Virtualbox (https://www.virtualbox.org/wiki/Downloads) already installed in your computer.

There is an assignment in Aula Global where you have to answer some questions related with the questions you will find below, so pay special attention to the paragraphs highlighted in grey colour. You can open the assignment at any time to check all questions.

2 Guidelines to start the virtual machine from your PC (recommended)

Step 1: Download VirtualBox.
VirtualBox from Oracle is available for free from the developer's website at **https://www.virtualbox.org/wiki/Downloads**.

- Choose the latest version for your operating system to download. (Mac Users choose OS X hosts)
- Also download the VirtualBox Extension Pack - All Supported Platforms



Download VirtualBox

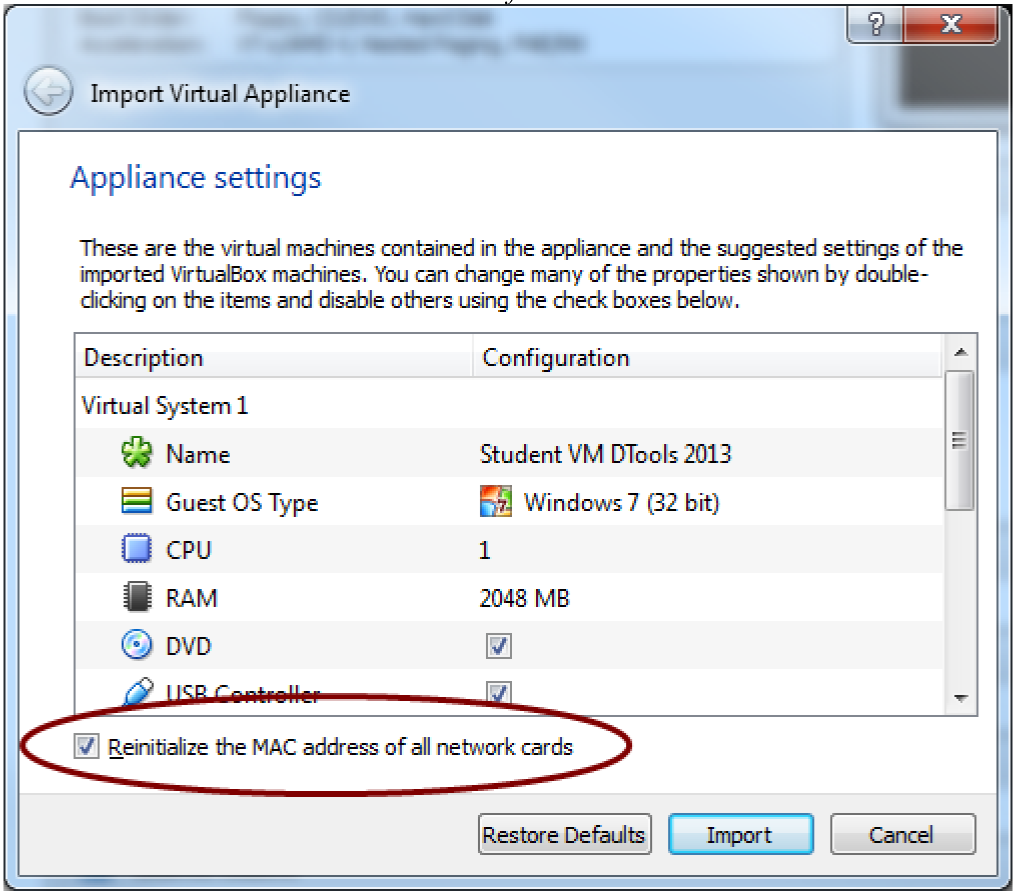
Here, you will find links to VirtualBox binaries and its source code.

VirtualBox binaries

By downloading, you agree to the terms and conditions of the respective license.

- **VirtualBox 5.1.28 platform packages**. The binaries are released under the terms of the GPL version 2.
  - Windows hosts
  - OS X hosts
  - Linux distributions
  - Solaris hosts
- **VirtualBox 5.1.28 Oracle VM VirtualBox Extension Pack** All supported platforms
  Support for USB 2.0 and USB 3.0 devices, VirtualBox RDP, disk encryption, NVMe and PXE boot for Intel cards. See this chapter from the User Manual for an introduction to this Extension Pack.
  The Extension Pack binaries are released under the VirtualBox Personal Use and Evaluation License (PUEL).
  Please install the extension pack with the same version as your installed version of VirtualBox:
  *If you are using **VirtualBox 5.0.40**, please download the extension pack here.*
- **VirtualBox 5.1.28 Software Developer Kit (SDK)** All platforms

Step 2: Install the VirtualBox program.
Step 3: Install the Extension Pack.
Install the Extension Pack by running the installer you downloaded.
Step 4: Launch virtualbox
Once downloaded, double click the OVA file to open it in VirtualBox. This will open a window similar to the one below that initiates an import process. Check the box marked *"Reinitialize the Mac address of all network cards"*.



Import Virtual Appliance

Appliance settings

These are the virtual machines contained in the appliance and the suggested settings of the imported VirtualBox machines. You can change many of the properties shown by double-clicking on the items and disable others using the check boxes below.

Description	Configuration
Virtual System 1	
Name	Student VM DTools 2013
Guest OS Type	Windows 7 (32 bit)
CPU	1
RAM	2048 MB
DVD	☑
USB Controller	☑

☑ Reinitialize the MAC address of all network cards

You should not need to change any configuration settings for the VM to work, but feel free to change any that you would like such as the name of the VM or where you want to save it on your hard drive. If you choose to adjust the RAM settings, it is strongly recommended you do not allocate any more than 50% of the RAM that your computer has.

Click *Import* to start importing the OVA file into your library. This process may take several minutes, depending on the size of your OVA file.

You can delete the OVA file you downloaded. The virtual machine has been extracted to your user folder.

Step 5: Start the VM
Once VirtualBox has completed importing the OVA file you should be able to see the virtual machine in your list. If desired, you can change the amount of RAM and CPU cores the VM will use when running. In VirtualBox, go to Settings, System, and increase the RAM and/or CPU cores if you experience slowness in the VM. Please be aware that the more you give the VM, the less the main OS has to work with. When the VM is shut down, all hardware resources go back to the main OS. We recommend at least 1 CPU core and 2 GB of RAM but not more than 50% of the total for either. Select **Start** (or double-click the VM object in the list) to launch the VM.

The VM Will start (and you can move to section 4)

3 Guidelines to start the virtual machine from the virtual labs

If you are using the virtual labs of the Telematics department, follow these steps to start the IMUNES virtual machine. Once you are connected to any virtual lab machine[1], run a terminal application like LXTerminal, selecting Aplicaciones (top-left), then “Herramientas del Sistema” and finally LXTerminal. Now, run the following command (be sure to copy it entirely):

```
/usr/dist/src/imunes/imunes_virt.sh
```

After 20-30 seconds, the virtual machine starts in a different window, similar to this screenshot:

[1] https://aulavirtual.lab.it.uc3m.es/

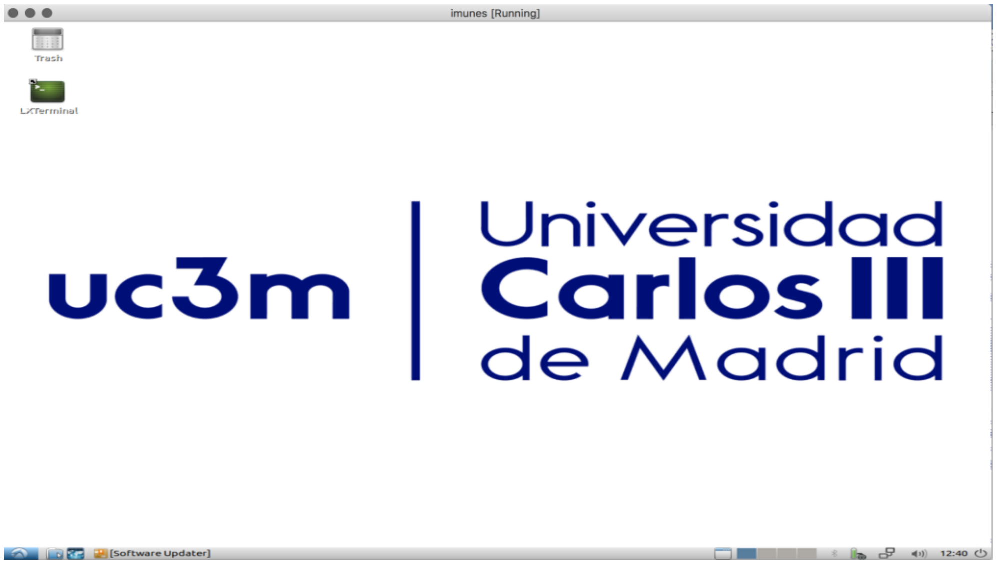


Figure 1. IMUNES VM main screen

4 Using IMUNES

It is important to notice that this is a Linux machine, more precisely a Lubuntu 22.04 distribution. Please, do not update the software, so it is safe to close the window entitled “Software Updater”.

In this laboratory you will use IMUNES (Integrated Multiprotocol Network Emulator/Simulator)[2]. This is a software used to test and analyse computer networks in a safe environment.

To start IMUNES, double-click the LXTerminal icon at the top-left of your window. A new window inside your virtual machine will show a terminal. In the terminal, run imunes as root with the following command:

```
sudo imunes
```

The password is *imunes*

The imunes GUI should start, showing an environment like the next figure:

[2] https://github.com/imunes/imunes

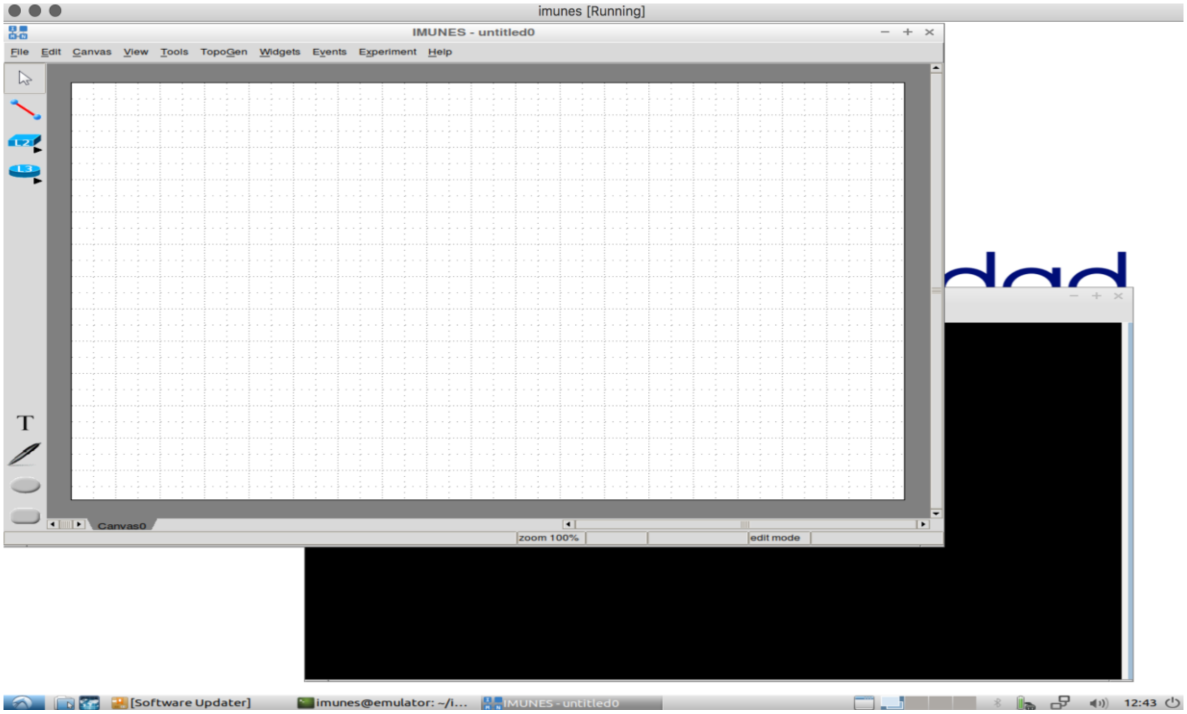


Figure 2. IMUNES main screen

Right now, you are in the "Edit mode", so you can add new switches, routers and/or PCs using the elements in the left bar. After that, you can connect elements using the "Link tool" (a diagonal segment with two dots at both ends) also in the left bar. Please notice that IMUNES automatically assigns MAC (Level 2) addresses to all network interface cards (NICs) and IP (Level 3) addresses to those NICs.

When you are happy with your network, you can start the emulation by clicking on the *Experiment* element in the top navigation bar (the almost last element) and then clicking on *Execute*. In the "running mode" you cannot modify any element. Wait until all elements are instantiated and configured.

Please notice that in this new mode, the left bar has changed. You can stop the emulation to go back to the "Edit mode" by clicking again in *Experiment* and then in *Terminate.*

One of the best characteristics of IMUNES is that you can connect to all computers and routers you have in your network. If you double-click on a computer or router, a new terminal window will be executed, where you can run commands that will be executed **in the selected computer/router**. Please, do not confuse the terminal in Lubuntu and the terminals executed inside IMUNES. We will represent the former using a dollar symbol ($) and the latter with a *hash* (#).

You do not need much more details about IMUNES, but you can always find the IMUNES user guide here: http://imunes.net/dl/imunes_user_guide.pdf

5 A simple local area network

In order to load the simple scenario, select File → Open. Open the Cloud_Networks_Lab folder, and then select *simple_topology.imn*

You can always drag-and-drop the correspondent components and create your own topology, as the one below.

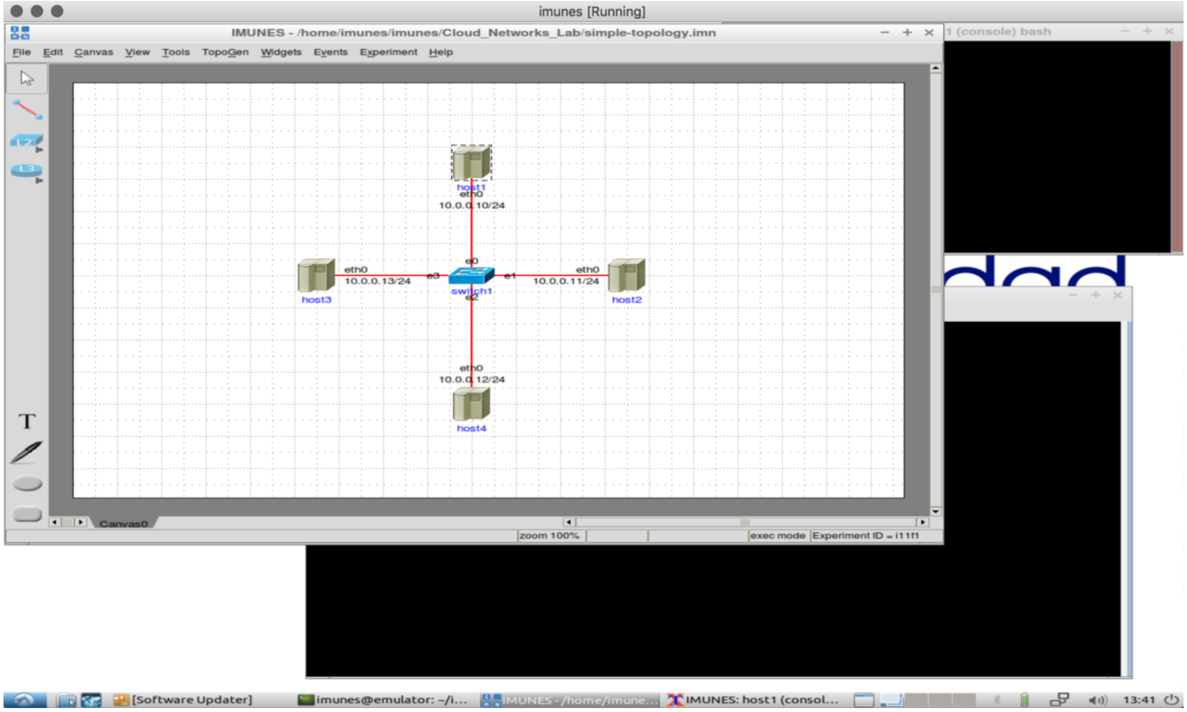


Figure 3. Simple topology

Now, execute the emulation (ExperimentExecute) and wait until all elements are running. Please, check Annex A at the end of this document to learn more about Wireshark. After the emulation is running, proceed with the following steps:

- Right-click on host2. A window will show all services available at host2. Select Wireshark and then interface eth0 (10.0.0.11). This will open Wireshark capturing packets in the eth0 interface.

- Double-click on host1 to open a terminal in that host. Then ping 10.0.0.11 to check the connectivity between those two hosts:

    - host1# ping 10.0.0.11

- In the Wireshark application analysing traffic received by host2, filter out other packets to show only those packets generated by the ping command.

Question 1: write down the most important information shown in each frame: source and destination MAC addresses, type of the network protocol and source and destination IP addresses (you will find this information in different headers!)

- Now, ping host3 from host1 while you are analysing the traffic received by host2 with Wireshark.

Question 2: What type of traffic do you receive?

6 Two Local Area Networks

Stop the previous simulation (Experiment → Terminate) and edit the previous network by connecting a router to the switch, and a host to the router, like the figure below:

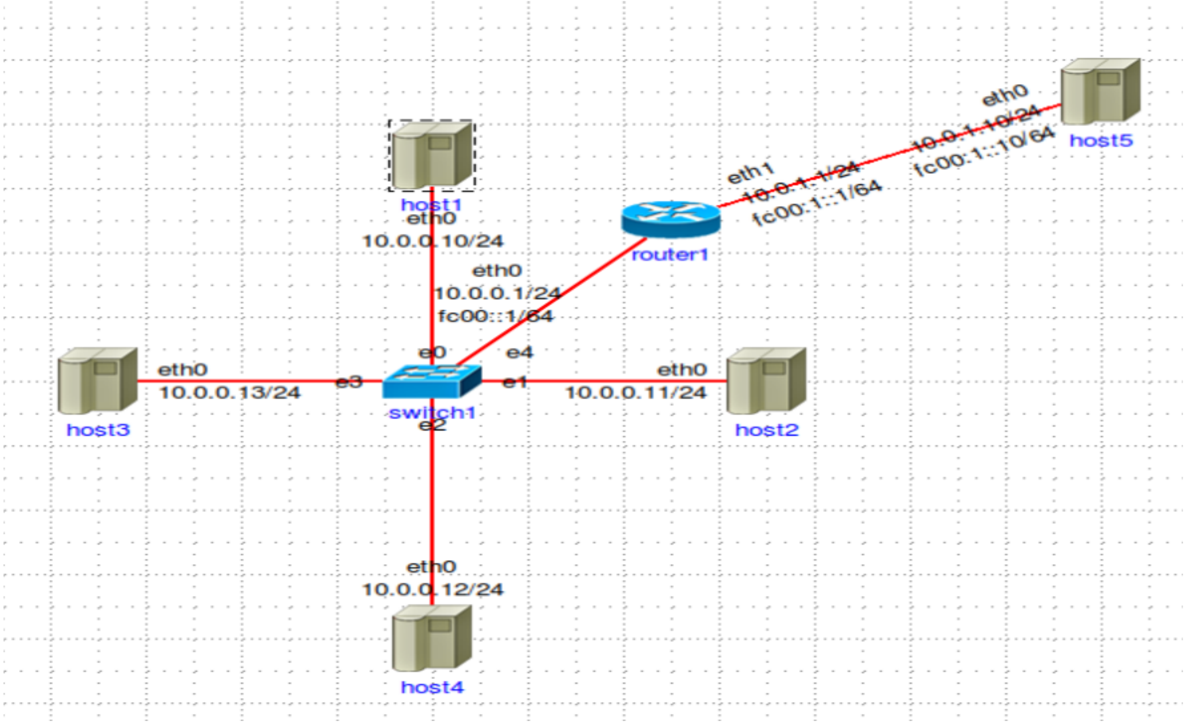


Figure 4. Two local area networks

- Run the Wireshark application, now at eth0 of host5.
- From host1, ping host5

Question 3: Analyze packets received at host5 from host1. Check source and destination MAC address and understand what is going on.

7 A hacker is sniffing your traffic

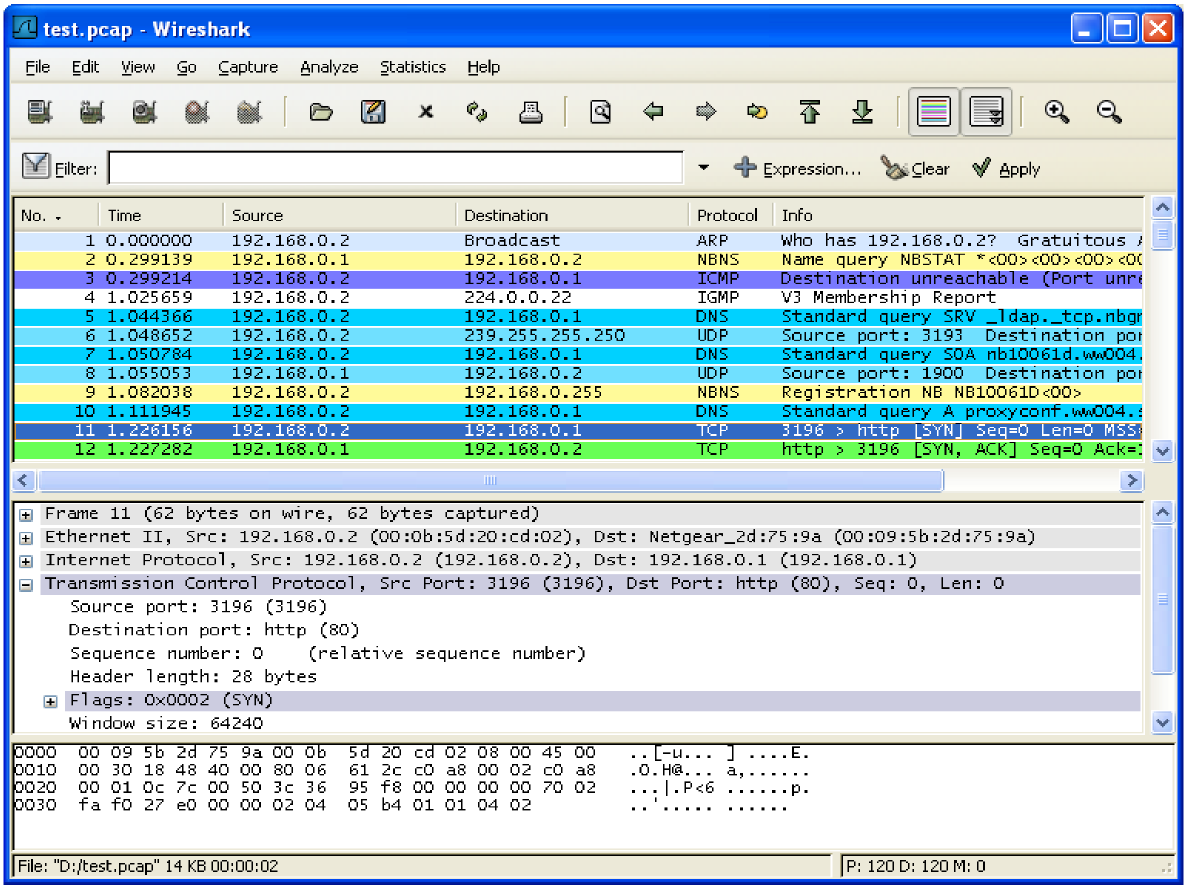
- With the previous network, now right-click host5 and select Services → telnet → Start
- While you are receiving traffic in host5, do the following in a terminal running in host1:
  - *telnet 10.0.1.10*
  - host5 login: *mylogin*
  - Password: *mypassword*
- Where you have to type everything that is in italic above. Please notice that while typing your password you cannot see it, because of security reasons.
- Telnet is an insecure protocol to remotely connect to a server. You will notice that the telnet has failed, but both the login and password has been exchanged between the client and the server.

Question 4: You are a hacker analyzing the traffic at host5. Can you get the plain text of both the login and password exchanged by these two devices? How?

Annex A - WireShark

WireShark is an open source application used to capture all traffic received at the host. In order to show all packets received, and not only the packets intended for the local host, it is important to run the application with root rights in Linux/OSX or as an administrator in Windows. This way, WireShark will run in promiscuous mode.

You can find the full documentation of WireShark in https://www.wireshark.org/docs/. You can skip the instructions to install the application, and go directly to the instructions to run it.



```
No. .  Time       Source         Destination        Protocol  Info
     1 0.000000   192.168.0.2    Broadcast          ARP       Who has 192.168.0.2?  Gratuitous A
     2 0.299139   192.168.0.1    192.168.0.2        NBNS      Name query NBSTAT *<00><00><00><00
     3 0.299214   192.168.0.2    192.168.0.1        ICMP      Destination unreachable (Port unre
     4 1.025659   192.168.0.2    224.0.0.22         IGMP      V3 Membership Report
     5 1.044366   192.168.0.2    192.168.0.1        DNS       Standard query SRV _ldap._tcp.nbgr
     6 1.048652   192.168.0.2    239.255.255.250    UDP       Source port: 3193  Destination por
     7 1.050784   192.168.0.2    192.168.0.1        DNS       Standard query SOA nb10061d.ww004.
     8 1.055053   192.168.0.1    192.168.0.2        UDP       Source port: 1900  Destination por
     9 1.082038   192.168.0.2    192.168.0.255      NBNS      Registration NB NB10061D<00>
    10 1.111945   192.168.0.2    192.168.0.1        DNS       Standard query A proxyconf.ww004.s
    11 1.226156   192.168.0.2    192.168.0.1        TCP       3196 > http [SYN] Seq=0 Len=0 MSS=
    12 1.227282   192.168.0.1    192.168.0.2        TCP       http > 3196 [SYN, ACK] Seq=0 Ack=1
```

```
⊞ Frame 11 (62 bytes on wire, 62 bytes captured)
⊞ Ethernet II, Src: 192.168.0.2 (00:0b:5d:20:cd:02), Dst: Netgear_2d:75:9a (00:09:5b:2d:75:9a)
⊞ Internet Protocol, Src: 192.168.0.2 (192.168.0.2), Dst: 192.168.0.1 (192.168.0.1)
⊟ Transmission Control Protocol, Src Port: 3196 (3196), Dst Port: http (80), Seq: 0, Len: 0
    Source port: 3196 (3196)
    Destination port: http (80)
    Sequence number: 0    (relative sequence number)
    Header length: 28 bytes
  ⊞ Flags: 0x0002 (SYN)
    Window size: 64240
```

```
0000  00 09 5b 2d 75 9a 00 0b  5d 20 cd 02 08 00 45 00   ..[-u... ] ....E.
0010  00 30 18 48 40 00 80 06  61 2c c0 a8 00 02 c0 a8   .0.H@... a,......
0020  00 01 0c 7c 00 50 3c 36  95 f8 00 00 00 00 70 02   ...|.P<6 ......p.
0030  fa f0 27 e0 00 00 02 04  05 b4 01 01 04 02         ..'..... ......
```

File: "D:/test.pcap" 14 KB 00:00:02 | P: 120 D: 120 M: 0

First of all, let’s try to capture some frames. To do so, use the Capture menu, selecting the promiscuous mode in the network card (eth0 or eth1 interface) and click Start. After capturing some packets, click stop.

The previous figure shows an example of the application window. Below the menu icons there are three windows: on the top, you can select any packet captured in the previous step. After selecting a packet, in the middle window you can inspect all fields/layers of the selected packet. You can click on the ‘+’ icons to expand the corresponding field/layer. Try to identify the fields of the Ethernet protocol. At the bottom window you can see the same information as in the previous window, but in hexadecimal representation this time.

One important feature of WireShark is the ability to filter captured packets using powerful filters, which follow the same syntax used in the **tcpdump** application. You can use as an example one of the proposed in:

http://wiki.wireshark.org/CaptureFilters

In order to filter captured frames to reduce the number of displayed frames, you can use the Display Filters. Please find some examples in the following link:

http://wiki.wireshark.org/DisplayFilters

The easiest way to filter the captured packets is by using the option "Filter:" in the box displayed in the main window of WireShark, where you can introduce the filter string (e.g. arp) or clicking the button near that box. In the latter case, a new window emerges where you can create several filters. In the right button named "Expression…" you can create your own filter. Try to create a new filter using the filters inside the Ethernet protocol. For example: *eth.padding "*is present".

It is possible to create other filters for other protocols (IP, ARP, UDP, etc.) and for addresses (see Help Display Filters). For example, try with the following filters: (1) to show only packets coming from a given IP address (example: ip.src == 10.0.0.10), (2) all packets going to a given IP address (ip.dst), etc. It is also possible to use the logical operators *and (&&)* and *or (||)* to combine expressions. For example, ip.src == 10.0.0.10 && ip.dst = 10.0.0.11 only shows frames generated by 10.0.0.10 addressed to 10.0.0.11.

Try different filters, following the rules you think are interesting (i.e. all packets exchanged between your machine and the one next to you, and only *telnet* packets)