Networking lab – Individual analysis using Wireshark – A.A. 2021/2022

1. Individual part

This part must the done by each student SEPARATELY, using the PC regularly connected to the Internet. You can work at home from any network, or from the LAIB, but **not from the Polito WiFi** network (due to the strict firewalling rules there). Clearly state it in you report, **describing which interface are you using (WiFi? Ethernet?), which connection are your using (ADSL? FTTH? 3G?), which provider, which Operating System, etc**.

Each student must upload the electronic version of the document on the didattica.polito.it server using his personal login, under the Network Measurement Lab upload area. The report shall not exceed 4 pages + 1 header page, plus the appendix.

The file name must be names "**Individual part - sXXXXXX.pdf**" where XXXXX must be the student ID (matricola) number. The report must be uploaded using **PDF FORMAT.** You must upload the **packet traces captured** by Wireshark during the test. Save the trace in .pcap format, using the menu File -> Save option. Name the file "nmap_sXXXXXX.pcap" (where XXXXXX is the student ID number). **Use the same file description in the upload panel.** Warning: do not use the "export" menu, but the "save as" menu entry!!!

2. PC configuration

1. Describe the network setup you are using. Where did you perform the analysis? When? From which hardware? Which network provider? Which Data Link layer? Using which operating system?
2. Bootstrap the PC leaving the network connected as normal and connect to the network using the automatic configuration. Try to disable/quit any applications that may be accessing the internet during the test to avoid interfering traffic being captured, e.g., quit skype, whatsapp, antivirus, etc.
3. Check the physical layer configuration of the PC, reporting in the table below. Use ethtool on linux, or the equivalent tool/GUI on Windows or MacOS.

The physical layer advertised by the linecard	
Current speed of ethX/wlanX	
Current Duplex status	
Link status	

4. Check the IP layer configuration of the PC, reporting

IP address	
netmask	
IP address of the default gateway	

3. Download and install nmap

`nmap` (Network Mapper and security auditing) is a tool designed to rapidly scan large networks, although it works fine against single hosts. It is available from https://nmap.org/download.html and it is available from almost any operating system. If it is not installed on your system, install it (via the package manager if available – `apt-get install nmap` – or by directly downloading and installing it from the web).

`Nmap` is a very complete and complicated tool, which can generate packets that abuse of the normal semantic of protocols, e.g., sending/forging a `icmp echo reply message` to a host directly. The goal of this lab is not to understand all possible usage of `nmap`, but rather to try to see how it is possible to abuse of protocols to infer information from a remote host.

NOTE: there are legal implications on trying to access and penetrate a remote server. Do not use `nmap` unless you know what you are doing and you have the right to do so.

4. Using nmap

The goal is to use nmap to scan a target and observe the packets being exchanged with the target. **Choose at least two hosts in your LAN** that are up and running. It could be the default gateway of your network, another host that you control, your smartphone which is connected to the same LAN, the smartTV, etc. Let `IP_ADDRESS` be your first target, then repeat the same on `IP_ADDRESS2, IP_ADDRESS3,...`

Using Wireshark, analyze the packet trace of the following commands:

1. Perform the scan of an active (if available) and inactive service on a remote host:

   ```
   nmap IP_ADDRESS -p 7, 22, 445

   sudo nmap IP_ADDRESS -p 7, 22, 445
   ```

2. What happens when run the above command as `root`, or as a `unprivileged` user? Observesee the sequence of packets sent. Compare the TCP `SYN` segments sent when running nmap as regular or root user. Check their length. Check the `TCP` source port choosen. Check the Initial Sequence Number in each SYN message. Do you notice anything wrong? Do they follow the standard behavior? Are those sent when using nmap as root regular `SYN` segments? Which entity is generating those segments? Comment and try to justify what you see. Hint, check the nmap man page and documentation.

3. For each target hosts you selected, run a complete scan, looking for which services are running in the ports from 1 to 150, using `TCP` and `UDP`:

   ```
   sudo nmap IP_ADDRESS -p 1-150[scan ports from 50 to 150 using TCP]

   sudo nmap IP_ADDRESS -p 1-150 -sU [scan ports from 50 to 150 using
   UDP]
   ```

4. Report and comment the plot that represent the checked port number versus time. Why TCP scan is much faster than UDP scan? How many times does nmap check a given port using TCP? And using UDP?
   Add the scripts you used to extract the data from the wireshark capture, and the eventual plot description for gnuplot in the appendix.