Networking Lab 4 – Analysis of the nmap command

1. nmap

nmap (Network Mapper and security auditing) is a tool that was designed to rapidly **scan large networks**, although it works fine against single hosts. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics.

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 an 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.

2. Using nmap

Configure H1, H2 and H3 so that they belong to the same subnet as usual, and activate the chargen and echo services on each of them. Using wireshark, analyze the packet trace of the following commands:

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

   ```
   nmap IP_ADDRESS -p 7, 99
   ```

   ```
   sudo nmap IP_ADDRESS -p 7, 99
   ```

2. what happens when run the above command as root, or as a unprivileged user? Try to see the sequence of packets sent. How are TCP source port chosen? Do they follow the normal behaviour? Compare the TCP segments sent with the SYN flag on. Which entity is generating those packets?
3. As above, but add the -O option (remote host Operating System identification)
   ```
   sudo nmap IP_ADDRESS -p 7, 99 -O
   ```
   Try to see the sequence of packets sent. How are TCP source port choosen? Do they follow the normal behaviour? How is it possible for and APPLICATION to generate those packets?
4. Run a complete scan of an host, looking for which services are running in the first 100 ports using the TCP and UDP ports:

   ```
   sudo nmap IP_ADDRESS -p 1-100
   ```

   ```
   sudo nmap IP_ADDRESS -p 1-100 -sU
   ```

5. Report and comment the plot that represent the number of port checked versus time. Why the TCP scan is much faster than the UDP scan? How many times nmap check a given port using TCP? And using UDP? Why it changes the behaviour?

Suggestions: use the -v option to increase verbosity if needed. Use the 'time' command to measure the system resource usage.