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

Simone Galota – s290198

1. PC configuration

1. The analysis has been performed at home, during the night in order to reduce the interference of other host's activities in the network. The house is equipped with an FTTH connection provided by FiberCop (a TIM company). The network configuration is the following one:

    - A router provided by ISP (eth switch and wi-fi integrated), firewall disabled.
    - A PC (192.168.1.16) running Ubuntu Live through USB, used as target host. This PC is connected to the router through an ethernet cable cat. 5e.
    - A MAC (192.168.1.220) computer from which Nmap command are launched, connected to the router using an ethernet cable cat. 5e through an USB network adapter supporting gigabit connection.

2. Check the physical layer configuration of the PC, reporting in the table below. Use ethtool on linux, or the equivalent tool on windows or MAC.

The physical layer advertised by the linecard	1000baseT/Full
Current speed of ethX/wlanX	1000 Mb/s
Current Duplex status	Full
Link status	active

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

IP address	192.168.1.220
netmask	255.255.255.0
IP address of the default gateway	192.168.1.1
Number of PCs that belong to your subnet	$2^8 - 3 = 253$ possible hosts. 2 PCs connected.

2. Using nmap

1. Launching command with "-p" option means that port scanning is performed only on specified ports:

```
nmap 192.168.1.16 -p 7,22,445
```

```
(base) Simones-MBP:~ simonegalota$ nmap 192.168.1.16 -p 7,22,445
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-04 03:26 CET
Nmap scan report for laboratorio.homenet.telecomitalia.it (192.168.1.16)
Host is up (0.00084s latency).

PORT    STATE  SERVICE
7/tcp   closed echo
22/tcp  open   ssh
445/tcp closed microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 0.18 seconds
```

```
sudo nmap 192.168.1.16 -p 7,22,445
```

```
(base) Simones-MBP:~ simonegalota$ sudo nmap 192.168.1.16 -p 7,22,445
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-04 03:31 CET
Nmap scan report for laboratorio.homenet.telecomitalia.it (192.168.1.16)
Host is up (0.00056s latency).

PORT    STATE  SERVICE
7/tcp   closed echo
22/tcp  open   ssh
445/tcp closed microsoft-ds
MAC Address: 10:C3:7B:20:19:5B (Asustek Computer)

Nmap done: 1 IP address (1 host up) scanned in 0.22 seconds
```

The simplest feature of nmap is specifying which is the state of the ports scanned. There are 4 possible states:

- **Open**: a service on the target host is LISTENING on that port.
- **Closed:** a closed port is accessible (it receives and replies to nmap), but there is no app listening
- **Filtered:** nmap cannot determine if the port is open because a packet filter could prevent packets reaching the port.
- **Unfiltered:** this state means that a port is accessible, but Nmap cannot determine if it is open or closed.

2. **What happens when run the above command as `root`, or as an `unprivileged` user? Try to see the sequence of packets sent. How are `TCP` source port chosen? Do they follow the normal behavior? Compare the `TCP` segments sent with the `SYN` flag on. Which entity is generating those segments? Are those normal `SYN` segments?**

As root

```
 1 0.000000      RealtekS_7… Broadcast   ARP    42 Who has 192.168.1.16? Tell 192.168.1.220
 2 0.000531      ASUSTekC_2… RealtekS_… ARP    60 192.168.1.16 is at 10:c3:7b:20:19:5b
```

First, Nmap checks if the target host is up and reachable before sending probing packets. It is a useful practice especially when we need to scan many hosts in a large network, because in this way the application avoids scanning a host if it is down, reducing the network load and speeding the scanning. Time is essential, less time we spend in scanning, less time the network has for understanding what is ongoing. This check is made by sending an ARP request to broadcast and waiting for hosts' answers. It is Nmap that, using some raw sockets, creates ARP packets and sends them directly to layer 2. A raw socket is a special socket that bypass the normal TCP/IP processing. It means that the packets travel from the application layer to the Ethernet layer directly, without using services provided by intermediate layers.

In this case TCP source port is chosen randomly but it is unique (35786), the same one for all probing activity. Also, the order in which destination ports are probed is chosen randomly too, for adding unpredictability and stealthiness to port scanning.

If target port is closed (like port 7 and 445), the target host replies with a [RST, ACK]. If a target port is open (like port 22 - ssh) the target host replies with [SYN, ACK].
The packets do not follow the normal behavior, because they are forged packets with forged headers. It is the application that generates them, formatting them as they were part of the protocol and using raw sockets as explained before instead of usual TCP socket (and system call tcp.connect()). In this way it is possible to abuse of the TCP protocol to gain information about target network, without respecting its constraints.
We have to say that, after that the probe packets are sent, the rest of the conversations keep on the normal tcp socket. Indeed, in the only case of open port, after [SYN, ACK] packet the target host receives a [RST] packet instead of [RST, ACK] because normal tcp socket has never opened a connection.

Furthermore, the length of these SYN segments is 20 bytes less than the regular ones (58 vs 78), given that there are 20 bytes of options less.

```
  5 0.048662      192.168.1.…  192.168.1.…  TCP    58 35786 → 445 [SYN] Seq=0 Win=1024 Len=0 MSS=146
  7 0.007… 192.168.1.2… 192.168.1.… TCP    78 52786 → 22 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=1326106988 TSecr=0 SACK_PERM=1
```

As unprivileged user

Not having root permission, the application uses normal layer 4 protocol to know if the target hosts are up and reachable. Thus, Nmap executes its activity opening and closing connection with regular handshakes.

```
1 0.000000    192.168.1.220  192.168.1.16   TCP    78 52784 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=2573034738 TSecr=0 SACK_PERM=1
2 0.000067    192.168.1.220  192.168.1.16   TCP    78 52785 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=3358001313 TSecr=0 SACK_PERM=1
3 0.000570    192.168.1.16   192.168.1.220  TCP    60 80 → 52784 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
4 0.000571    192.168.1.16   192.168.1.220  TCP    60 443 → 52785 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
```

To check if the target host is up, nmap chose to use SYN packets over TCP because UDP is not connection oriented and ICMP is blocked by firewall. Among all the possible ports the application chose the ones not usually blocked by the O.S. firewall. Usually port 80 (http) and 443 (http over TLS). Once the host is declared up, the port scanning begins. The source ports are chosen randomly and differently for each stream.
If target port is closed (like port 7 and 445), the target host replies with a [RST, ACK]. If a target port is open (like port 22 - ssh) the target host replies with [SYN, ACK].
In case of open port, the handshake is closed with the remaining [ACK]. But, since the host running nmap does not want to establish a connection, it is also sent a [RST, ACK].

In both cases a reverse-DNS resolution is performed (by default for all responsive hosts), even if you can omit it by adding "-n" option. The reason is, as stated in the nmap manual, that the Domain Name System is a great source of information, because the internal systems of an organization are often named according to what is their task. For instance, firewalls are named "fw" or "fw-1" etc.

```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-04 03:31 CET
Nmap scan report for laboratorio.homenet.telecomitalia.it (192.168.1.16)
```

3. **Run a complete host scan, looking for which services are running in the ports from 50 to 150, on `TCP` and `UDP`:**

```
sudo nmap 192.168.1.16 -p 50-150
```

```
(base) Simones-MBP:~ simonegalota$ sudo nmap 192.168.1.16 -p 50-150
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-04 03:40 CET
Nmap scan report for laboratorio.homenet.telecomitalia.it (192.168.1.16)
Host is up (0.00038s latency).
All 101 scanned ports on laboratorio.homenet.telecomitalia.it (192.168.1.16) are in ignored states.
Not shown: 101 closed tcp ports (reset)
MAC Address: 10:C3:7B:20:19:5B (Asustek Computer)

Nmap done: 1 IP address (1 host up) scanned in 0.25 seconds
```

```
sudo nmap 192.168.1.16 -p 50-150 -sU
```

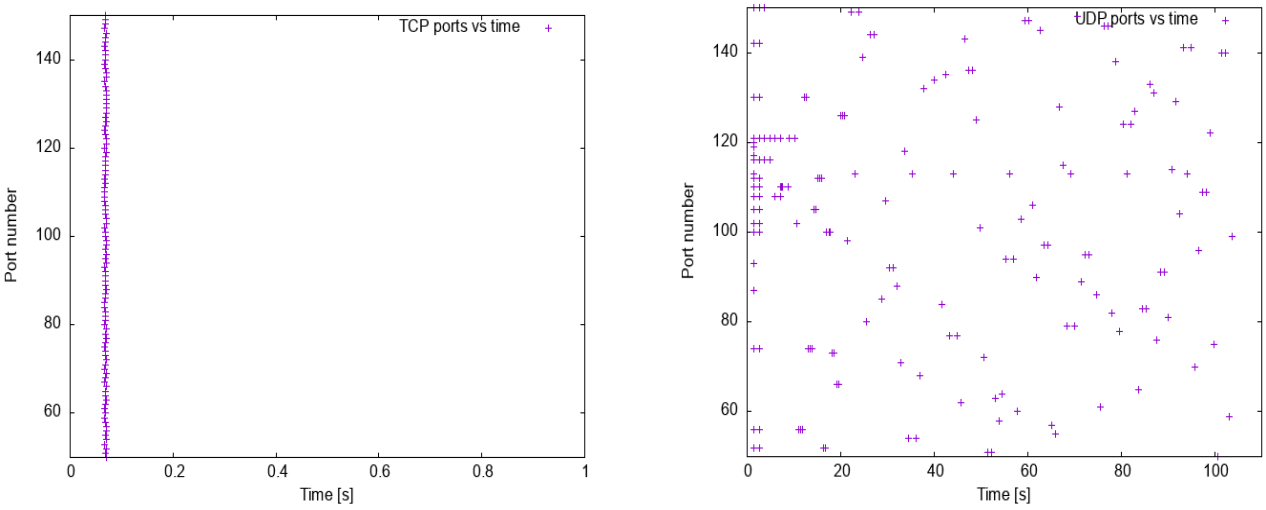
```
(base) Simones-MBP:~ simonegalota$ sudo nmap 192.168.1.16 -p 50-150 -sU
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-04 03:43 CET
Nmap scan report for laboratorio.homenet.telecomitalia.it (192.168.1.16)
Host is up (0.00075s latency).
All 101 scanned ports on laboratorio.homenet.telecomitalia.it (192.168.1.16) are in ignored states.
Not shown: 101 closed udp ports (port-unreach)
MAC Address: 10:C3:7B:20:19:5B (Asustek Computer)

Nmap done: 1 IP address (1 host up) scanned in 102.51 seconds
```

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 nmap check a given port using TCP? And using UDP?**

command used for extracting data: `cat udp.txt | tr -s ' '| cut -d ' ' -f 3,10 > input_udp.txt`

script used for plotting:

```
set terminal png
set out 'UDP-plot.png'
set style data point
set xlabel "Time [s]"
set ylabel "Port number"
set xrange [0:110]
set yrange [50:150]
plot "input_udp.txt" using 1:2 title "UDP ports vs time"
```



The ports result all closed for both protocols. It is noticed that there is a substantial difference in the duration of the two scans. 0,25s for TCP against 102,51s for UDP.

For TCP scanning, previous considerations can be applied. From the capture we see that TCP SYN have always the same source port because there is no real will of opening a connection ("half-open scanning"). Also, from the plot on the left we can see that one probe packet is sent to each port and they are sent all at the same time. So, the duration is very short.

On the other hand, UDP scanning lasted about 100 s, much more than TCP case. This is due to the protocol's nature, that is connectionless and best-effort. Thus, packets could be lost (with consequent retransmission) and replies by the target hosts is not assured. It must be said that UDP port scanning is available only with root privileges, because access to ICMP messages is require. Indeed, open ports should reply with a UDP packet, closed ports with an ICMP Port Unreachable and filtered ports with other ICMP errors.

Moreover, the principal reason of slowness of UDP scanning is that Linux systems limit by default the number of ICMP Port Unreachable packets sent in the time unit (reasonably a countermeasure to port scanning). Therefore, Nmap is forced to slow down the execution because packet could be ignored by the target host.

Lastly, except for well-known ports, in both kind of scanning the order of ports to scan is randomly chosen, in order to reduce the possibility of being detected by the target network.

5. **Choose an open and a closed port on your target. Run nmap with the –O option.**
   Try to see the sequence of packets sent. How are TCP source ports chosen? Do they follow the normal behavior? How is it possible for an APPLICATION to generate those packets?

```
sudo nmap 192.168.1.16 -p 7,22 -O
```

```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-09 12:44 CET
Nmap scan report for laboratorio.homenet.telecomitalia.it (192.168.1.16)
Host is up (0.00075s latency).

PORT   STATE  SERVICE
7/tcp  closed echo
22/tcp open   ssh
MAC Address: 10:C3:7B:20:19:5B (Asustek Computer)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.17 seconds
```

It is possible (not always) to know which operating system is running on the victim host. Some fingerprints mechanism is used: the application sends some bogus TCP messages interacting directly with the Ethernet layer (creating ad hoc packet as explained previously). Then, based on the answer received, application looks up in a database, where are saved the typical replies of each OS to a specific message.
Looking at the sequence of packets captured, we see a lot of packets sent by the application. The scanning can be divided in 2 phases. The first one in which application wants to understand which port is open and which closed. It is equal to the previous ones considered, so can be applied the considerations made for TCP scanning.

In the second one we can see several probe packets sent to perform OS detection:

- Six TCP SYN packets sent to an open port. These packets are called sequence generation and they vary in the TCP options they use and TCP window field value.
- Two ICMP Echo requests to the target.
- A UDP packet sent to a random closed port.
- A SYN packet which also has ECN, CWR, Reserved flags set. This test is called TCP Explicit Congestion Notification
- Six TCP packet with the following format:
  - Three packets sent to a detected open port (22): [None], [FIN,SYN,PSH,URG], [ACK]
  - Three packets sent to a detected closed port (7): [SYN], [ACK], [FIN,PSH,URG]

It is also possible use -A option to understand what version of Linux is running on target host. By looking at the version of the protocol running on the open port. In our case, SSH.

```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-09 12:50 CET
Nmap scan report for laboratorio.homenet.telecomitalia.it (192.168.1.16)
Host is up (0.00070s latency).

PORT   STATE  SERVICE VERSION
7/tcp  closed echo
22/tcp open   ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 9e:45:dc:bf:d5:3d:18:a4:f2:a7:68:8f:de:35:b6:e2 (RSA)
|   256 9f:cb:16:7a:77:94:f9:64:7f:0b:18:37:a0:0a:67:51 (ECDSA)
|_  256 26:ac:41:d0:59:f3:dd:a5:5b:57:42:56:8f:cc:f0:23 (ED25519)
MAC Address: 10:C3:7B:20:19:5B (Asustek Computer)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.70 ms laboratorio.homenet.telecomitalia.it (192.168.1.16)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.92 seconds
```

```
sudo nmap 192.168.1.16 -p 7,22 -A
```