

• **Sniff your network and explain what you can see there. Gain as much information about it while remaining passive.**

- I created two docker containers out of rastasheep/ubuntu-sshd image (just because it was left from one of the previous lab)
- connected them to the same docker network
- Their ip addresses are 172.18.0.2 and 172.18.0.3.
- I already had wireshark installed, but it is pretty easy to do in Ubuntu via \$sudo apt install wireshark
- File with sniffed traffic can be found there:
https://github.com/BananaAndBread/SNA/blob/master/sniffed_things.pcapng

Traffic was made firstly by:

```
$nc -lvp 5000
```

```
$cat 11-0.txt |nc 172.18.0.2 5000
```

where 11-0.txt is Alice in the Wonderland

Then by:

```
$nc -lvp 5000
```

```
$cat /dev/random |nc 172.18.0.2 5000
```

That is the end of tcp connection:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	172.18.0.3	172.18.0.2	TCP	66	39956 → 5000 [FIN, ACK] Seq=1 Win=229 Len=0 TSval=1482010857 TSecr=879230455
2	0.000166226	172.18.0.2	172.18.0.3	TCP	66	5000 → 39956 [FIN, ACK] Seq=1 Ack=2 Win=1174 Len=0 TSval=879282533 TSecr=1482010857

ARP protocol finds physical machine address according to the IP address in the message in a local area network:

ARP	42	Who has 172.18.0.2? Tell 172.18.0.3
ARP	42	172.18.0.2 is at 02:42:ac:12:00:02
ARP	42	Who has 172.18.0.3? Tell 172.18.0.2
ARP	42	172.18.0.3 is at 02:42:ac:12:00:03

Beginning of the tcp connection:

8	5.237591110	172.18.0.3	172.18.0.2	TCP	74	39972 → 5000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1482016095 TSecr=0 WS=128
9	5.237701971	172.18.0.2	172.18.0.3	TCP	74	5000 → 39972 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=879287771 TSecr=1482016095

Messages themselves, IPA" is the ip.access "GSM over IP" protocol:

The image shows a Wireshark interface with the file 'sniffed_things.pcapng' open. The main packet list pane shows several packets, with packet 13 selected. The packet details pane shows the selected packet's structure: Internet Protocol Version 4, Src: 172.18.0.3, Dst: 172.18.0.2; Transmission Control Protocol, Src Port: 39972, Dst Port: 5000, Seq: 7241, Ack: 1, Len: 952; and IPA protocol ip.access, type: unknown 0x65. The packet bytes pane shows the raw data of the selected packet, with a hex dump and ASCII representation. The ASCII representation shows the beginning of a message: '...ore, a nd...beh1', 'nd it wa s a litt', 'le door about fi', 'fteen in ches hig', 'h: she t ried the', '...little golden', 'key in t he lock,', 'and to her grea'.

He does not get what type of message it is

```
> IPA protocol ip.access, type: unknown 0x65
```

It possible to see parts from the Alice in the Wonderland:

00b0	20 61 6e 64 20 74 6f 20 68 65 72 20 67 72 65 61	and to her grea
00c0	74 20 64 65 6c 69 67 68 74 20 69 74 20 66 69 74	t delight t it fit
00d0	74 65 64 21 0d 0a 0d 0a 41 6c 69 63 65 20 6f 70	ted!... Alice op
00e0	65 6e 65 64 20 74 68 65 20 64 6f 6f 72 20 61 6e	ened the door an
00f0	64 20 66 6f 75 6e 64 20 74 68 61 74 20 69 74 20	d found that it
0100	6c 65 64 20 69 6e 74 6f 20 61 20 73 6d 61 6c 6c	led into a small
0110	20 70 61 73 73 61 67 65 2c 20 6e 6f 74 0d 0a 6d	passage , not m
0120	75 63 68 20 6c 61 72 67 65 72 20 74 68 61 6e 20	uch larg er than

After the last part sent connection finishes

52	5.239999415	172.18.0.3	172.18.0.2	IPA	10/73 unknown 0x74
53	5.240681452	172.18.0.2	172.18.0.3	TCP	66 5000 → 39972 [ACK] Seq=1 Ack=173596
54	10.666479022	172.18.0.3	172.18.0.2	TCP	66 39972 → 5000 [FIN, ACK] Seq=173596
55	10.666649704	172.18.0.2	172.18.0.3	TCP	66 5000 → 39972 [FIN, ACK] Seq=1 Ack=1
56	10.666709545	172.18.0.3	172.18.0.2	TCP	66 39972 → 5000 [ACK] Seq=173597 Ack=2

That is 172.18.0.3 trying to knock to the locked port (locked, because server was stopped):

56	10.666709545	172.18.0.3	172.18.0.2	TCP	66 39972 → 5000 [ACK] Seq=173597 Ack=2
57	36.538988168	172.18.0.3	172.18.0.2	TCP	74 40014 → 5000 [SYN] Seq=0 Win=29200
58	36.539077412	172.18.0.2	172.18.0.3	TCP	54 5000 → 40014 [RST, ACK] Seq=1 Ack=1

New tcp connection:

59	40.436543649	172.18.0.3	172.18.0.2	TCP	74 40028 → 5000 [SYN] Seq=0 Win=29200
60	40.436652720	172.18.0.2	172.18.0.3	TCP	74 5000 → 40028 [SYN, ACK] Seq=0 Ack=1

Now the random bytes are sent

Here's ARP (in the middle of TCP connection) helps 172.18.0.2 to find MAC address of a client:

96	45.627339688	02:42:ac:12:00:02	02:42:ac:12:00:03	ARP	42 Who has 172.18.0.3? Tell 172.18.0.2
97	45.627434216	02:42:ac:12:00:03	02:42:ac:12:00:02	ARP	42 172.18.0.3 is at 02:42:ac:12:00:03

Nothing more interesting here.

2)

- **Scan your network without attempting any attack. Just proceed with a gentle network discovery and report your analysis. This should not be illegal esp. on an internal network.**

This is the result of the nmap:

```
root@96677e73e57f:~# nmap -sP 172.18.0.0/16

Starting Nmap 7.60 ( https://nmap.org ) at 2019-10-15 17:56 UTC
Nmap scan report for 172.18.0.1
Host is up (0.0000090s latency).
MAC Address: 02:42:50:D0:C5:B7 (Unknown)
Nmap scan report for machine_one.some_network (172.18.0.2)
Host is up (-0.011s latency).
MAC Address: 02:42:AC:12:00:02 (Unknown)
Nmap scan report for 96677e73e57f (172.18.0.3)
Host is up.
```