

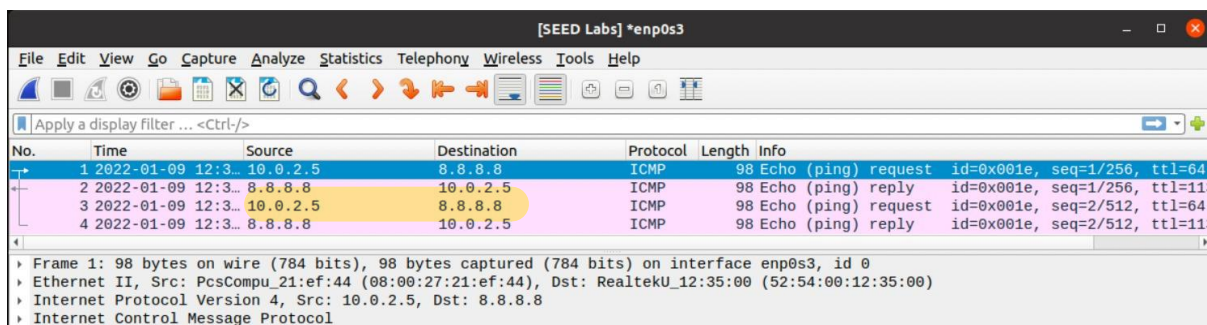
תקשורת ומחשוב - Ex6
ברק עמרם - 209369289
לירוי - 209366970

Task 1: Using Scapy to sniff and Spoof Packets

Task1.1: Sniffing Packets

Task1.1A:

```
[01/09/22] seed@VM:~/.../PartA_py$ sudo python3 1.1a_sniffer.py  
###[ Ethernet ]###  
dst      = 52:54:00:12:35:00  
src      = 08:00:27:21:ef:44  
type     = IPv4  
###[ IP ]###  
version  = 4  
ihl      = 5  
tos      = 0x0  
len      = 84  
id       = 20087  
flags    = DF  
frag     = 0  
ttl      = 64  
proto    = icmp  
chksum   = 0xd01d  
src      = 10.0.2.5  
dst      = 8.8.8.8  
\options \  
###[ ICMP ]###  
type     = echo-request  
code     = 0  
chksum   = 0x2dd4  
id       = 0x1e  
seq      = 0x1  
###[ Raw ]###  
load     = '\xe9\x1d\xdba\x00\x00\x00\x00F\xba\x00\x00\x00\x00\x00\x;  
  
###[ Ethernet ]###  
dst      = 08:00:27:21:ef:44  
src      = 52:54:00:12:35:00  
type     = IPv4  
###[ IP ]###  
version  = 4  
ihl      = 5  
tos      = 0x0  
len      = 84  
id       = 0  
flags    =  
frag     = 0  
ttl      = 113  
proto    = icmp  
chksum   = 0x2d95  
src      = 8.8.8.8  
dst      = 10.0.2.5  
\options \  
###[ ICMP ]###  
type     = echo-reply  
code     = 0  
chksum   = 0x35d4
```



Task 1.1B:

```
[01/09/22]seed@VM:~/.../PartA_py$ sudo python3 1.1b_sniffer_ICMP.py
#### Ethernet ]###
    dst      = 52:54:00:12:35:00
    src      = 08:00:27:21:ef:44
    type     = IPv4
#### IP ]###
    version  = 4
    ihl      = 5
    tos      = 0x0
    len      = 84
    id       = 55469
    flags    = DF
    frag     = 0
    ttl      = 64
    proto    = icmp
    checksum = 0x45e7
    src      = 10.0.2.5
    dst      = 8.8.8.8
    \options \
#### ICMP ]###
    type     = echo-request
    code     = 0
    checksum = 0xbfc2
    id       = 0x20
    seq      = 0x1
#### Raw ]###
    load     = '\x83\x1f\xdba\x00\x00\x00\x00\x17\xc8\x03\x00\x00\x00\x00\x00\x'
#### Ethernet ]###
    dst      = 08:00:27:21:ef:44
    src      = 52:54:00:12:35:00
    type     = IPv4
#### IP ]###
    version  = 4
    ihl      = 5
    tos      = 0x0
    len      = 84
    id       = 0
    flags    = 
    frag     = 0
    ttl      = 113
    proto    = icmp
    checksum = 0x2d95
    src      = 8.8.8.8
    dst      = 10.0.2.5
    \options \
#### ICMP ]###
    type     = echo-reply
    code     = 0
    checksum = 0xc7c2
```

The image shows a Wireshark capture of ICMP Echo (ping) packets. The packet list shows four packets: a request from 10.0.2.5 to 8.8.8.8, and three replies from 8.8.8.8 to 10.0.2.5. The packet details pane shows the structure of the ICMP Echo request and reply, including the ID, sequence number, and TTL.

No.	Time	Source	Destination	Protocol	Length	Info
1	2022-01-09 12:4...	10.0.2.5	8.8.8.8	ICMP	98	Echo (ping) request id=0x0020, seq=1/256, ttl=64
2	2022-01-09 12:4...	8.8.8.8	10.0.2.5	ICMP	98	Echo (ping) reply id=0x0020, seq=1/256, ttl=113
3	2022-01-09 12:4...	10.0.2.5	8.8.8.8	ICMP	98	Echo (ping) request id=0x0020, seq=2/512, ttl=64
4	2022-01-09 12:4...	8.8.8.8	10.0.2.5	ICMP	98	Echo (ping) reply id=0x0020, seq=2/512, ttl=113

בחלק זה המרחרר שלנו לוחד רק פקטות מסוג ICMP, ביצענו פינג אל כתובת 8.8.8.8 ואפשר לראות במרחרר שלנו את הtype של ה - ICMP message (request = 8,reply = 0).

Capture any TCP packet that comes from a particular IP and with a destination port number 23:

The image shows a Wireshark capture of various network traffic. The packet list shows a mix of TCP and DNS packets. A Telnet connection attempt is visible, with a SYN packet from 10.9.0.1 to 10.9.0.5 on port 23, followed by several retransmissions and a final ACK. The packet details pane shows the structure of the Telnet data.

No.	Time	Source	Destination	Protocol	Length	Info
1	2022-01-09...	10.9.0.1	10.9.0.5	TCP	76	34048 → 23 [SYN] Seq=3938825293 win=64240 L...
2	2022-01-09...	10.9.0.1	10.9.0.5	TCP	76	[TCP Out-Of-Order] 34048 → 23 [SYN] Seq=393...
3	2022-01-09...	10.9.0.5	10.9.0.1	TCP	76	23 → 34048 [SYN, ACK] Seq=1142034836 Ack=39...
4	2022-01-09...	10.9.0.5	10.9.0.1	TCP	76	[TCP Out-Of-Order] 23 → 34048 [SYN, ACK] Se...
5	2022-01-09...	10.9.0.1	10.9.0.5	TCP	68	34048 → 23 [ACK] Seq=3938825294 Ack=1142034...
6	2022-01-09...	10.9.0.1	10.9.0.5	TCP	68	[TCP Dup ACK 5#1] 34048 → 23 [ACK] Seq=3938...
7	2022-01-09...	10.9.0.1	10.9.0.5	TELN...	95	Telnet Data ...
8	2022-01-09...	10.9.0.1	10.9.0.5	TCP	95	[TCP Retransmission] 34048 → 23 [PSH, ACK] ...
9	2022-01-09...	10.9.0.5	10.9.0.1	TCP	68	23 → 34048 [ACK] Seq=1142034837 Ack=3938825...
10	2022-01-09...	10.9.0.5	10.9.0.1	TCP	68	[TCP Dup ACK 9#1] 23 → 34048 [ACK] Seq=1142...
11	2022-01-09...	10.9.0.5	8.8.3.3	DNS	83	Standard query 0x35ae PTR 1.0.9.10.in-addr...
12	2022-01-09...	10.9.0.5	8.8.3.3	DNS	83	Standard query 0x35ae PTR 1.0.9.10.in-addr...
13	2022-01-09...	10.0.2.5	8.8.3.3	DNS	83	Standard query 0x35ae PTR 1.0.9.10.in-addr...
14	2022-01-09...	10.0.2.5	104.18.123.25	TCP	56	50726 → 443 [ACK] Seq=2236218718 Ack=151853...
15	2022-01-09...	104.18.123.25	10.0.2.5	TCP	62	[TCP ACKed unseen segment] 443 → 50726 [ACK...
16	2022-01-09...	10.9.0.5	8.8.8.8	DNS	83	Standard query 0x35ae PTR 1.0.9.10.in-addr...
17	2022-01-09...	10.9.0.5	8.8.8.8	DNS	83	Standard query 0x35ae PTR 1.0.9.10.in-addr...
18	2022-01-09...	10.0.2.5	8.8.8.8	DNS	83	Standard query 0x35ae PTR 1.0.9.10.in-addr...
19	2022-01-09...	8.8.8.8	10.0.2.5	DNS	83	Standard query response 0x35ae No such name...
20	2022-01-09...	8.8.8.8	10.9.0.5	DNS	83	Standard query response 0x35ae No such name...
21	2022-01-09...	8.8.8.8	10.9.0.5	DNS	83	Standard query response 0x35ae No such name...
22	2022-01-09...	10.9.0.5	10.9.0.1	TELN...	80	Telnet Data ...

```
seed@VM: ~/.../PartA_py
[01/09/22]seed@VM:~/.../PartA_py$ telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.

seed@VM: ~/.../PartA_py
[01/09/22]seed@VM:~/.../PartA_py$ sudo python3 1.1b_sniffer_TCP_port23.py
#### Ethernet ####
dst      = 02:42:0a:09:00:05
src      = 02:42:88:30:de:7c
type     = IPv4
#### IP ####
version  = 4
ihl      = 5
tos      = 0x10
len      = 60
id       = 63488
flags    = DF
frag     = 0
ttl      = 64
proto    = tcp
chksum   = 0x2e94
src      = 10.9.0.1
dst      = 10.9.0.5
\options \
#### TCP ####
sport    = 34048
dport    = telnet
seq      = 3938825293
ack      = 0
dataofs  = 10
reserved = 0
flags    = S
window   = 64240
chksum   = 0x1446
urgptr   = 0
options  = [('MSS', 1460), ('SAckOK', b''), ('Timestamp', (3427990219,
0)), ('NOP', None), ('WScale', 7)]
#### Ethernet ####
dst      = 02:42:0a:09:00:05
src      = 02:42:88:30:de:7c
type     = IPv4
#### IP ####
version  = 4
ihl      = 5
tos      = 0x10
len      = 52
id       = 63489
flags    = DF
frag     = 0
ttl      = 64
```

שלחנו הודעת telnet להוסט אחר (hostA) עם הכתובת 10.9.0.5, השתמשנו בtelnet כדי ליצור חיבור TCP בפורט 23.

Capture packets comes from or to go to a particular subnet:

Internet Protocol Version 4, Src: 10.0.2.5, Dst: 128.230.0.2

Internet Control Message Protocol

Type: 8 (Echo (ping) request)

Code: 0

Checksum: 0xa2fb [correct]

[Checksum Status: Good]

Identifier (BE): 34 (0x0022)

Identifier (LE): 8704 (0x2200)

Sequence number (BE): 1 (0x0001)

Sequence number (LE): 1 (0x0001)

proto = icmp

chksum = 0xba69

src = 10.0.2.5

dst = 128.230.0.2

\options \

###[ICMP]###

type = echo-request

code = 0

chksum = 0xa2fb

id = 0x22

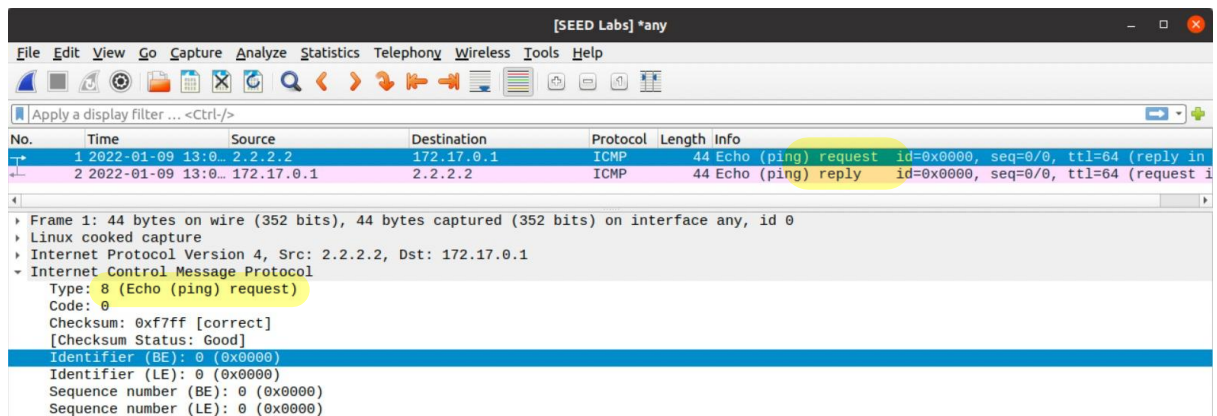
seq = 0x1

###[Raw]###

load = '\xea\xdb\xa0\xa0\xa0\xa0\xc4\x8b\x0c\xa0\xa0\xa0\xa0'

בחלק זה אנו מריצים תוכנית פייתון אשר מסניפה פקטות שנמצאים ספציפית ב-subnet 16\128.230.0.0 ביצענו פינג אל כתובת 128.230.0.2 אשר נמצאת ב-subnet ה"ל".

Task 1.2: Spoofing ICMP Packets



ניתן לראות כי ביצענו spoofing לפקטת icmp, מסוג 8 (request echo), את כתובת ה ip של הפקטה שיצרנו הגדרנו בתור 2.2.2.2 שזוהי אינה כתובת קו אמיתית. ניתן לראות ב Wireshark כי נשלחה פקטה מ-2.2.2.2 ל-172.17.0.1 שזו היא הכתובת של אחד מה- hosts שעל הוירטואל משין שלנו. בנוסף נשלחה reply echo מכתובת 172.17.0.1 אל 2.2.2.2

Task 1.3: Traceroute

```
seed@VM: ~/.../PartA_py
[01/09/22]seed@VM:~/.../PartA_py$ sudo python3 1.3_traceroute.py
1 - curr addr: 10.0.2.1
2 - curr addr: 10.0.0.138
3 - curr addr: 212.179.37.1
4 - curr addr: 10.250.44.58
5 - curr addr: 212.25.77.10
6 - curr addr: 62.219.189.2
7 - curr addr: 10.250.99.2
8 - curr addr: 212.25.70.69
9 - curr addr: 108.170.225.59
10 - curr addr: 142.251.78.87
dest addr: 142.251.37.238 is in length of: 10
[01/09/22]seed@VM:~/.../PartA_py$
```

כתבנו תוכנית פיתון שמגלה את אורך ה-traceroute, שלחנו icmp echo request עם ttl שקבע בהתחלה ל-1, בכל פעם שנקבל תגובה שהיא לא תשובה echo reply כלומר (time to live exceeded), נקבל 1 ל-ttl

הישן ונשלח שוב את ה-tcp, מתי שנקבל echo reply אז הגענו ליעד ועכשיו אנחנו יודעים את אורך המסלול - כמה זמן קיבלנו time to live exceeded

No.	Time	Source	Destination	Protocol	Length	Info
7	2022-01-09 13:1...	10.0.2.5	142.251.37.238	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=1 (no response f...
8	2022-01-09 13:1...	10.0.2.1	10.0.2.5	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)
9	2022-01-09 13:1...	127.0.0.1	127.0.0.53	DNS	72	Standard query 0xdfb0 A google.com
10	2022-01-09 13:1...	127.0.0.53	127.0.0.1	DNS	88	Standard query response 0xdfb0 A google.com A 142.251.37.238
11	2022-01-09 13:1...	10.0.2.5	142.251.37.238	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=2 (no response f...
12	2022-01-09 13:1...	10.0.0.138	10.0.2.5	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)
13	2022-01-09 13:1...	127.0.0.1	127.0.0.53	DNS	72	Standard query 0xc5ec A google.com
14	2022-01-09 13:1...	127.0.0.53	127.0.0.1	DNS	88	Standard query response 0xc5ec A google.com A 142.251.37.238
15	2022-01-09 13:1...	10.0.2.5	142.251.37.238	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=3 (no response f...
16	2022-01-09 13:1...	212.179.37.1	10.0.2.5	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)
17	2022-01-09 13:1...	127.0.0.1	127.0.0.53	DNS	72	Standard query 0xffdd A google.com
18	2022-01-09 13:1...	127.0.0.53	127.0.0.1	DNS	88	Standard query response 0xffdd A google.com A 142.251.37.238
19	2022-01-09 13:1...	10.0.2.5	142.251.37.238	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=4 (no response f...

▶ Frame 8: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface any, id 0
 ▶ Linux cooked capture
 ▶ Internet Protocol Version 4, Src: 10.0.2.1, Dst: 10.0.2.5
 ▶ Internet Control Message Protocol
 Type: 11 (Time-to-live exceeded)
 Code: 0 (Time to live exceeded in transit)
 Checksum: 0xf4ff [correct]
 [Checksum Status: Good]
 Unused: 00000000
 ▶ Internet Protocol Version 4, Src: 10.0.2.5, Dst: 142.251.37.238

45	2022-01-09 13:1...	127.0.0.1	127.0.0.53	DNS	72	Standard query 0x4d23
46	2022-01-09 13:1...	127.0.0.53	127.0.0.1	DNS	88	Standard query respons
47	2022-01-09 13:1...	10.0.2.5	142.251.37.238	ICMP	44	Echo (ping) request i
48	2022-01-09 13:1...	142.251.37.238	10.0.2.5	ICMP	62	Echo (ping) reply i

▶ Frame 48: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface any, id 0
 ▶ Linux cooked capture
 ▶ Internet Protocol Version 4, Src: 142.251.37.238, Dst: 10.0.2.5
 ▶ Internet Control Message Protocol
 Type: 0 (Echo (ping) reply)
 Code: 0
 Checksum: 0x0000 incorrect, should be 0xffff
 [Checksum Status: Bad]
 Identifier (BE): 0 (0x0000)
 Identifier (LE): 0 (0x0000)
 Sequence number (BE): 0 (0x0000)
 Sequence number (LE): 0 (0x0000)
 [Request frame: 47]

Task 1.4: Sniffing and-then Spoofing:

למשימה זו כתבנו מרחרח שיש לו מסנן של פרוטוקול ARP ו-ICMP. כאשר הסניפר שלי מקבל בקשת ICMP או ARP אז הוא שולח את ה-tcp האלה לפונקציה שמחליטה אם הtcp הוא מסוג ARP או מסוג ICMP. אם זה ARP אז הtcp נשלח ל- spoof_arp_pkt ששולח ARP מזויף כנל לגבי אם זה בקשת tcp ICMP אז זה שלח אותה ל- spoof_icmp_pkt ששולח תשובת ICMP מזויפת

ביצענו ping לכתובת 2.2.2.2 אשר לא קיימת ברשת

```

seed@VM: ~/.../PartA_py
[01/09/22]seed@VM:~/.../PartA_py$ sudo python3 1.4_sniff_and_spoof.py

Original Packet.....
Source IP : 10.0.2.5
Destination IP : 2.2.2.2

Spoofed Packet...
Source IP : 2.2.2.2
Destination IP : 10.0.2.5

Original Packet.....
Source IP : 10.0.2.5
Destination IP : 2.2.2.2
  
```

ביצענו ping לכתובת 10.9.0.60 אשר לא קיימת ברשת הפנימית

```
seed@9fb7703834c3:~$ ping 10.9.0.60
PING 10.9.0.60 (10.9.0.60) 56(84) bytes of data.
64 bytes from 10.9.0.60: icmp_seq=1 ttl=64 time=120 ms
From 10.9.0.1: icmp_seq=2 Redirect Host(New nexthop: 10.9.0.60)
64 bytes from 10.9.0.60: icmp_seq=2 ttl=64 time=22.8 ms
From 10.9.0.1: icmp_seq=3 Redirect Host(New nexthop: 10.9.0.60)
64 bytes from 10.9.0.60: icmp_seq=3 ttl=64 time=28.0 ms

[01/09/22]seed@VM: ~/.../PartA_py$ sudo python3 1.4_sniff_and_spoof.py
spoofer packet information:
.
Sent 1 packets.
send arp replay
spoofer packet information:
.
Sent 1 packets.
send arp replay

Original Packet.....
Source IP : 10.9.0.6
Destination IP : 10.9.0.60

Spoofed Packet...
Source IP : 10.9.0.60
Destination IP : 10.9.0.6
spoofer packet information:
.

21 2022-01-09... 10.9.0.6 10.9.0.60 ICMP 100 Echo (ping) request id=0x0067, seq=1/256, ...
22 2022-01-09... 10.9.0.6 10.9.0.60 ICMP 100 Echo (ping) request id=0x0067, seq=1/256, ...
23 2022-01-09... 02:42:88:30:d... ARP 44 who has 10.9.0.60? Tell 10.9.0.1
24 2022-01-09... 02:42:88:30:d... ARP 44 who has 10.9.0.60? Tell 10.9.0.1
25 2022-01-09... 02:42:88:30:d... ARP 44 who has 10.9.0.60? Tell 10.9.0.1
26 2022-01-09... 00:00:00_00:0... ARP 44 10.9.0.6 is at 00:00:00:00:00:00
27 2022-01-09... 10.9.0.60 10.9.0.6 ICMP 100 Echo (ping) reply id=0x0067, seq=1/256, ...
28 2022-01-09... 10.9.0.60 10.9.0.6 ICMP 100 Echo (ping) reply id=0x0067, seq=1/256, ...
```


בצענו ping לכתובת 8.8.8.8 אשר קיימת ברשת

```
seed@9fb7703834c3:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=64 time=47.6 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=112 time=50.0 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=2 ttl=64 time=19.7 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=112 time=49.7 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=3 ttl=64 time=28.4 ms

[01/09/22] seed@VM: ~/.../PartA_py$ sudo python3 1.4_sniff_and_spoof.py

Original Packet.....
Source IP : 10.9.0.6
Destination IP : 8.8.8.8

Spoofed Packet...
Source IP : 8.8.8.8
Destination IP : 10.9.0.6

Original Packet.....
Source IP : 10.0.2.5
Destination IP : 8.8.8.8

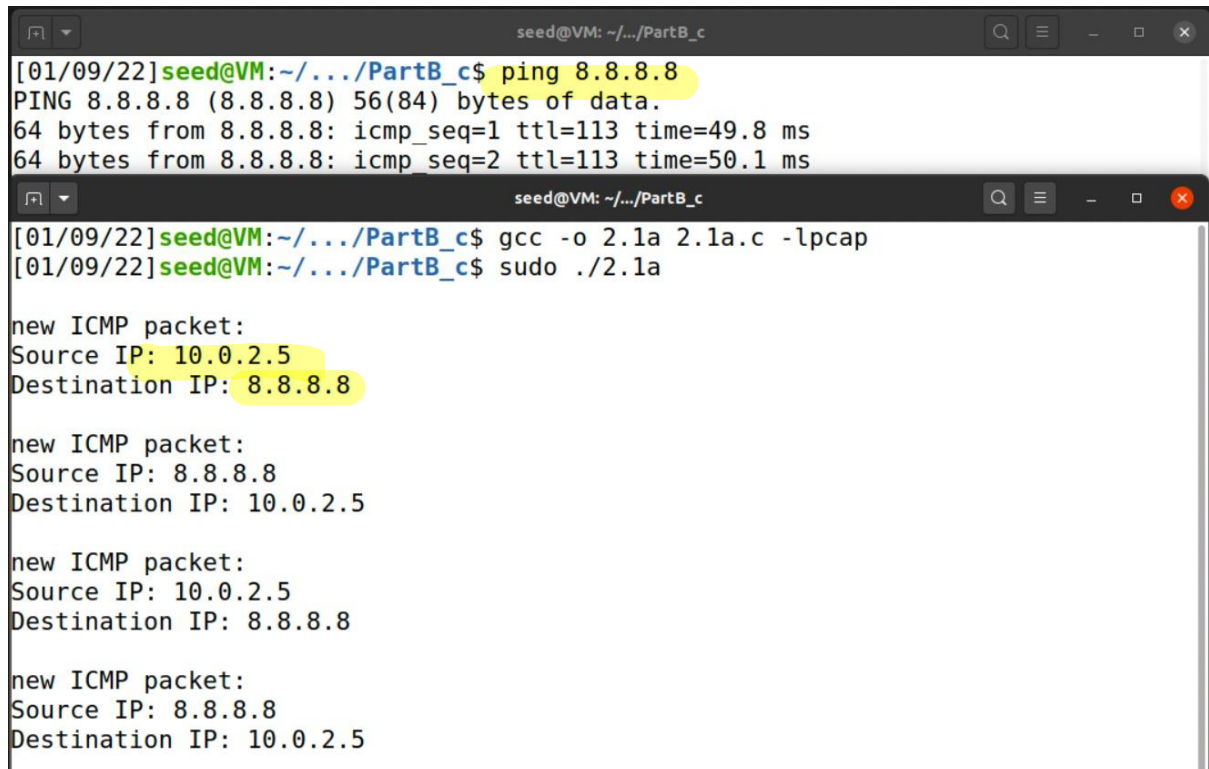
Spoofed Packet...
Source IP : 8.8.8.8
Destination IP : 10.0.2.5
spoof packet information:
```

7	2022-01-09...	10.9.0.6	8.8.8.8	ICMP	100 Echo (ping) request	id=0x0069, seq=1/256, ...
10	2022-01-09...	10.9.0.6	8.8.8.8	ICMP	100 Echo (ping) request	id=0x0069, seq=1/256, ...
13	2022-01-09...	10.0.2.5	8.8.8.8	ICMP	100 Echo (ping) request	id=0x0069, seq=1/256, ...
19	2022-01-09...	8.8.8.8	10.9.0.6	ICMP	100 Echo (ping) reply	id=0x0069, seq=1/256, ...
20	2022-01-09...	8.8.8.8	10.9.0.6	ICMP	100 Echo (ping) reply	id=0x0069, seq=1/256, ...
25	2022-01-09...	8.8.8.8	10.0.2.5	ICMP	100 Echo (ping) reply	id=0x0069, seq=1/256, ...

Lab Task Set 2: Writing Programs to Sniff and Spoof Packets:

Task 2.1: Writing Packet Sniffing Program:

Task 2.1A: Understanding How a Sniffer Works



```
seed@VM: ~/.../PartB_c
[01/09/22] seed@VM:~/.../PartB_c$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=113 time=49.8 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=113 time=50.1 ms

seed@VM: ~/.../PartB_c
[01/09/22] seed@VM:~/.../PartB_c$ gcc -o 2.1a 2.1a.c -lpcap
[01/09/22] seed@VM:~/.../PartB_c$ sudo ./2.1a

new ICMP packet:
Source IP: 10.0.2.5
Destination IP: 8.8.8.8

new ICMP packet:
Source IP: 8.8.8.8
Destination IP: 10.0.2.5

new ICMP packet:
Source IP: 10.0.2.5
Destination IP: 8.8.8.8

new ICMP packet:
Source IP: 8.8.8.8
Destination IP: 10.0.2.5
```

שאלה 1

ראשית עלינו לבחור באיזה ממשק אנו רוצים לרחרח את החבילות.
לאחר מכן, נשתמש בספריית Pcap ונשתמש בפונקציה "pcap open live".
לאחר מכן עלינו לשנות את המסנן עבור איזו חבילה אנו רוצים לרחרח. ובסופו של דבר נשתמש בפונקציית pcap_loop על מנת ללכוד את החבילות.

שאלה 2

אנחנו צריכים את root privilege כדי להפעיל תוכנית סניפר מכיוון שללא root privilege לא נוכל לקבל גישה ל-NIC. על מנת לקבל גישה ל-NIC אנו זקוקים להרשאת מנהל.

שאלה 3

אם Promiscuous mode 'מופעל' אז נוכל לרחרח אחרי כל חבילה שעוברת דרך הרשת שלנו. עם זאת, אם זה במצב 'כבוי', אנו יכולים לראות רק את החבילות שנשלחו ישירות אלינו וממנו.
אז כדי לרחרח או לזייף את כל החבילות ברשת שלנו, עלינו להפעיל את Promiscuous mode.

[SEED Labs] Capturing from any

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

icmp

No.	Time	Source	Destination	Protocol	Length	Info
1	2022-01-09...	10.0.2.5	8.8.8.8	ICMP	100	Echo (ping) request id=0x002a, seq=1/256, ...
2	2022-01-09...	8.8.8.8	10.0.2.5	ICMP	100	Echo (ping) reply id=0x002a, seq=1/256, ...
5	2022-01-09...	10.0.2.5	8.8.8.8	ICMP	100	Echo (ping) request id=0x002a, seq=2/512, ...
6	2022-01-09...	8.8.8.8	10.0.2.5	ICMP	100	Echo (ping) reply id=0x002a, seq=2/512, ...
7	2022-01-09...	10.0.2.5	8.8.8.8	ICMP	100	Echo (ping) request id=0x002a, seq=3/768, ...
8	2022-01-09...	8.8.8.8	10.0.2.5	ICMP	100	Echo (ping) reply id=0x002a, seq=3/768, ...

Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0xfc08 [correct]
[checksum status: Good]

Task 2.1B: Writing Filters

Capture the ICMP packets between two specific host

```
seed@VM: ~/.../PartB_c
[01/09/22]seed@VM:~/.../PartB_c$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=113 time=50.3 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=113 time=49.9 ms

seed@VM: ~/.../PartB_c
[01/09/22]seed@VM:~/.../PartB_c$ gcc -o 2.1b_ICMP 2.1b_ICMP.c -lpcap
[01/09/22]seed@VM:~/.../PartB_c$ sudo ./2.1b_ICMP

new packet:
Source IP: 10.0.2.5
Destination IP: 8.8.8.8

new packet:
Source IP: 8.8.8.8
Destination IP: 10.0.2.5

new packet:
Source IP: 10.0.2.5
Destination IP: 8.8.8.8

new packet:
Source IP: 8.8.8.8
Destination IP: 10.0.2.5
```

[SEED Labs] Capturing from any

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

icmp

No.	Time	Source	Destination	Protocol	Length	Info
7	2022-01-09...	10.0.2.5	8.8.8.8	ICMP	100	Echo (ping) request id=0x002c, seq=1/256, ...
8	2022-01-09...	8.8.8.8	10.0.2.5	ICMP	100	Echo (ping) reply id=0x002c, seq=1/256, ...
9	2022-01-09...	10.0.2.5	8.8.8.8	ICMP	100	Echo (ping) request id=0x002c, seq=2/512, ...
10	2022-01-09...	8.8.8.8	10.0.2.5	ICMP	100	Echo (ping) reply id=0x002c, seq=2/512, ...

בחלק זה ביצענו פינג לכתובת 8.8.8.8 והסניפר שלנו קובע את הפילטר שלנו. ניתן לראות בסניפר שלנו כי הוא הוא תופס את פקטות הICMP.

Capture the TCP packets with a destination port number in the range from 10 to 100

```
seed@VM: ~/.../PartB_c
[01/09/22]seed@VM:~/.../PartB_c$ telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.

seed@VM: ~/.../PartB_c
[01/09/22]seed@VM:~/.../PartB_c$ gcc -o 2.1b_TCP 2.1b_TCP.c -lpcap
[01/09/22]seed@VM:~/.../PartB_c$ sudo ./2.1b_TCP

new packet:
Source IP: 10.9.0.1
Destination IP: 10.9.0.5

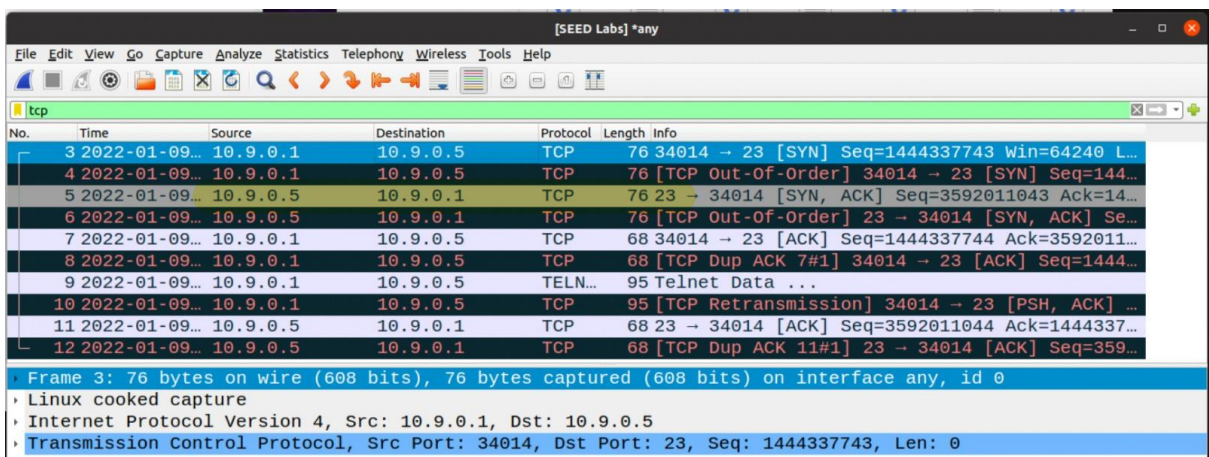
new packet:
Source IP: 10.9.0.1
Destination IP: 10.9.0.5

new packet:
Source IP: 10.9.0.1
Destination IP: 10.9.0.5

new packet:
Source IP: 10.9.0.1
Destination IP: 10.9.0.5

new packet:
Source IP: 10.9.0.1
Destination IP: 10.9.0.5

new packet:
```



No.	Time	Source	Destination	Protocol	Length	Info
3	2022-01-09...	10.9.0.1	10.9.0.5	TCP	76	34014 → 23 [SYN] Seq=1444337743 Win=64240 L...
4	2022-01-09...	10.9.0.1	10.9.0.5	TCP	76	[TCP Out-Of-Order] 34014 → 23 [SYN] Seq=144...
5	2022-01-09...	10.9.0.5	10.9.0.1	TCP	76	23 → 34014 [SYN, ACK] Seq=3592011043 Ack=14...
6	2022-01-09...	10.9.0.5	10.9.0.1	TCP	76	[TCP Out-Of-Order] 23 → 34014 [SYN, ACK] Se...
7	2022-01-09...	10.9.0.1	10.9.0.5	TCP	68	34014 → 23 [ACK] Seq=1444337744 Ack=3592011...
8	2022-01-09...	10.9.0.1	10.9.0.5	TCP	68	[TCP Dup ACK 7#1] 34014 → 23 [ACK] Seq=1444...
9	2022-01-09...	10.9.0.1	10.9.0.5	TELN...	95	Telnet Data ...
10	2022-01-09...	10.9.0.1	10.9.0.5	TCP	95	[TCP Retransmission] 34014 → 23 [PSH, ACK] ...
11	2022-01-09...	10.9.0.5	10.9.0.1	TCP	68	23 → 34014 [ACK] Seq=3592011044 Ack=1444337...
12	2022-01-09...	10.9.0.5	10.9.0.1	TCP	68	[TCP Dup ACK 11#1] 23 → 34014 [ACK] Seq=359...

Frame 3: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface any, id 0

- Linux cooked capture
- Internet Protocol Version 4, Src: 10.9.0.1, Dst: 10.9.0.5
- Transmission Control Protocol, Src Port: 34014, Dst Port: 23, Seq: 1444337743, Len: 0

בחלק זה אנו משנים את הפילטר ל TCP ומשתמשים בtelnet אשר משתמש ב TCP בפורט 23, שלחנו telnet לכתובת 10.9.0.5 שזו אחת מכתובות ה VM והסניפר מדפיס את כל הנתונים של פקטות TCP

Task 2.1C: Sniffing Passwords

```
seed@VM: ~/.../PartB_c
[01/09/22]seed@VM:~/.../PartB_c$ telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
0b101611f6cd login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

seed@VM: ~/.../PartB_c
[01/09/22]seed@VM:~/.../PartB_c$ gcc -o 2.1c 2.1c.c -lpcap
[01/09/22]seed@VM:~/.../PartB_c$ sudo ./2.1c

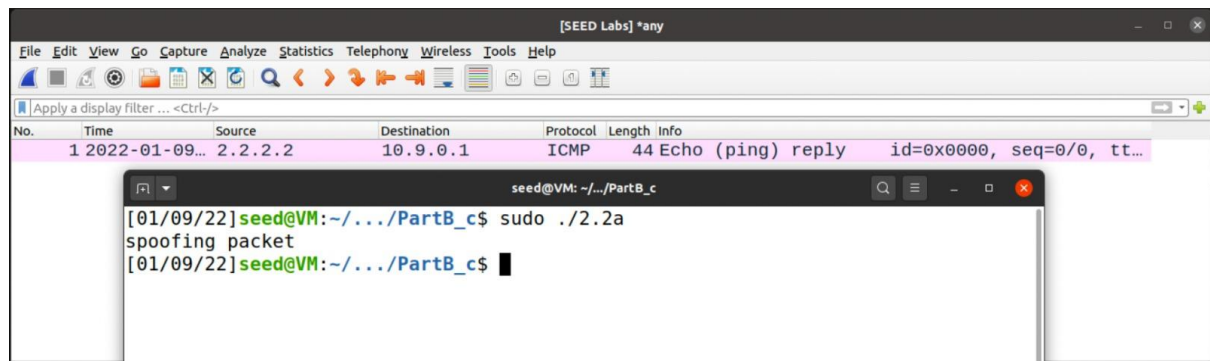
Source port: 33986
Destination port: 23
Source ip: 10.9.0.1
Destination ip: 10.9.0.5
Packet data:      #                8C_a*_b_J_J      Z_h

Source port: 33986
Destination port: 23
Source ip: 10.9.0.1
Destination ip: 10.9.0.5
Packet data:      # /                8C_aJ$e_]_]      Z_h

Source port: 33986
Destination port: 23
Source ip: 10.9.0.1
Destination ip: 10.9.0.5
Packet data:      # /                _!_"_'_#      8C_a_Ge_B_B      Z_h
```

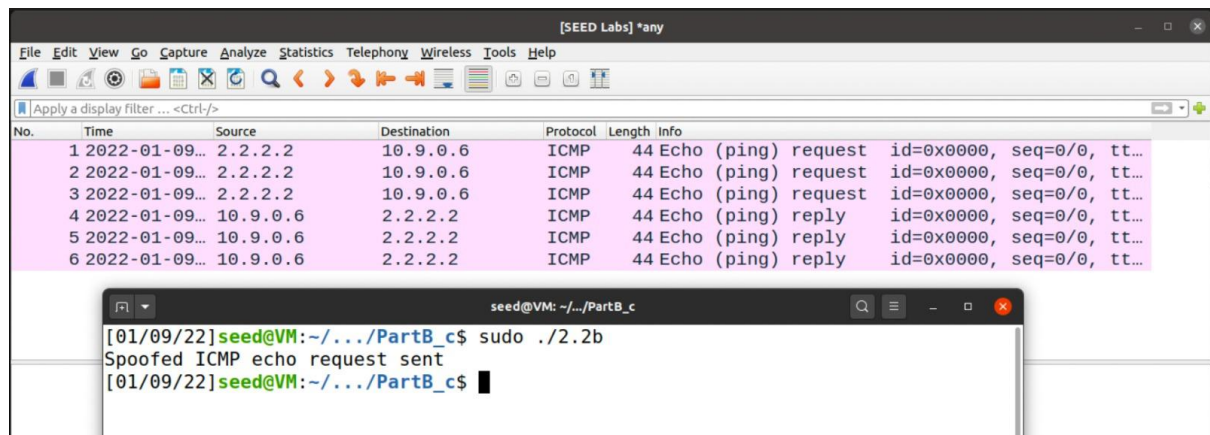
בחלק זה התחברנו כמו בסעיף הקודם ב telnet 10.9.0.5, רחרחנו אחר כל הפקטות TCP והדפסנו את הנתונים שלהם וסימא חלקית.

Task 2.2A: Write a spoofing program



בחלק זה שלחנו פקטה מזוייפת לכתובת IP של 2.2.2.2.

Task 2.2B: Spoof an ICMP Echo Request



בחלק זה נבצע spoofing לפקטת ICMP אוני ניצור פקטת בקשה ICMP אשר נערוך את כתובת ה SRC שלה לכתובת מפורקת שאינה בהכרח קיימת. נשלח פקטה זו ונרצה לראות כי אכן התקבלה פקטת תגובה לאותה הכתובת.

שאלה 4

לא ניתן לשנות את אורך ה IP לאורך שרירותי, נקבל הודעת שגיאה ב Wireshark שלא לנתח את תוכן הפקטה בגלל ההבדל בין אורך הפקטה שהוזן לאורך הפקטה שנשלחה.

שאלה 5

לא חייב לחשב את ה - checksum של הכתובת IP מכיוון שמערכת ההפעלה מחשבת אותו באופן אוטומטי

שאלה 6

לא שימוש ב root privilege לא נוכל לפתוח socket raw פעולה זו מוגנת עי הרשאות root כדי למנוע ממשתמשים להגיע לחמרה הפיזית של המחשב ככלל כאשר אנו מחוברים כרוט ישנה אפשרות לעשות טעויות חמורות כמו למשל למחוק קבצים חשובים לכן פעולות מסוימות מוגנות על ידי הרשאות רוט כדי למנוע מטעויות שכאלו לקרות .

Task 2.3: Sniff and then Spoof

The screenshot displays a Wireshark network traffic capture and two terminal windows. The Wireshark interface shows a list of ICMP Echo (ping) requests and replies between 10.0.2.5 and 2.2.2.2. The terminal windows show the execution of ping commands from a VM named 'seed' to a host named 'PartB_c'.

Wireshark Packet List:

No.	Time	Source	Destination	Protocol	Length	Info
1	2022-01-09...	10.0.2.5	2.2.2.2	ICMP	100	Echo (ping) request id=6
2	2022-01-09...	2.2.2.2	10.0.2.5	ICMP	100	Echo (ping) reply id=6
3	2022-01-09...	10.0.2.5	2.2.2.2	ICMP	100	Echo (ping) request id=6
4	2022-01-09...	2.2.2.2	10.0.2.5	ICMP	100	Echo (ping) reply id=6
5	2022-01-09...	10.0.2.5	2.2.2.2	ICMP	100	Echo (ping) request id=6
6	2022-01-09...	10.0.2.5	8.8.8.8	ICMP	100	Echo (ping) request id=6
7	2022-01-09...	8.8.8.8	10.0.2.5	ICMP	100	Echo (ping) reply id=6
8	2022-01-09...	2.2.2.2	10.0.2.5	ICMP	100	Echo (ping) reply id=6
9	2022-01-09...	10.0.2.5	2.2.2.2	ICMP	100	Echo (ping) request id=6
10	2022-01-09...	10.0.2.5	8.8.8.8	ICMP	100	Echo (ping) request id=6
11	2022-01-09...	8.8.8.8	10.0.2.5	ICMP	100	Echo (ping) reply id=6
12	2022-01-09...	8.8.8.8	10.0.2.5	ICMP	100	Echo (ping) reply id=6
13	2022-01-09...	10.0.2.5	2.2.2.2	ICMP	100	Echo (ping) request id=6
14	2022-01-09...	10.0.2.5	8.8.8.8	ICMP	100	Echo (ping) request id=6
15	2022-01-09...	8.8.8.8	10.0.2.5	ICMP	100	Echo (ping) reply id=6
16	2022-01-09...	2.2.2.2	10.0.2.5	ICMP	100	Echo (ping) reply id=6
17	2022-01-09...	10.0.2.5	2.2.2.2	ICMP	100	Echo (ping) request id=6
18	2022-01-09...	10.0.2.5	8.8.8.8	ICMP	100	Echo (ping) request id=6
19	2022-01-09...	8.8.8.8	10.0.2.5	ICMP	100	Echo (ping) reply id=6
20	2022-01-09...	8.8.8.8	10.0.2.5	ICMP	100	Echo (ping) reply id=6
21	2022-01-09...	10.0.2.5	2.2.2.2	ICMP	100	Echo (ping) request id=6

Terminal 1 (Left):

```
[01/09/22]seed@VM: ~/.../PartB_c$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=113 time=61.4 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=113 time=48.5 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=44 time=1463 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=3 ttl=113 time=48.3 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=113 time=48.4 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=44 time=2479 ms (DUP!)
```

Terminal 2 (Right):

```
[01/09/22]seed@VM: ~/.../PartB_c$ ping 2.2.2.2
PING 2.2.2.2 (2.2.2.2) 56(84) bytes of data.
64 bytes from 2.2.2.2: icmp_seq=1 ttl=44 time=601 ms
64 bytes from 2.2.2.2: icmp_seq=2 ttl=44 time=625 ms
64 bytes from 2.2.2.2: icmp_seq=3 ttl=44 time=662 ms
64 bytes from 2.2.2.2: icmp_seq=4 ttl=44 time=1672 ms
64 bytes from 2.2.2.2: icmp_seq=5 ttl=44 time=2669 ms
^Z
```

בחלק זה אנו מרחירים פקטות icmp ולאחר מכן שולחים פקטה מזויפת שמחזירה תשובה של icmp.