מטלה 5:

Packet Sniffing and Spoofing

מגישים:
211696521
328596978

תוכן עניינים:

חלק א

Sniffer

שאלה: למה צריך הרשאת השורש בשביל להפעיל תוכנית הSniffer ?
היכן תוכנית נכשלת במידה והיא מופעלת ללא ההרשאה הזו ?

תשובה: ה Sniffer זקוק להרשאה זו כדי לגשת להתקן הרשת ב promiscuous mode .
במצב זה התקן הרשת לוכד את כל החבילות המועברות ברשת , ללא קשר אם הן מיועדות ל host
או לא.
זה הכרחי כדי שהתוכנית ה sniffer תוכל ללכוד ולנתח את כל החבילות ברשת.

אם מופעלת תוכנית ה Sniffer ללא הרשאה זו, היא לא תוכל לגשת להתקן הרשת ב promiscuous mode.
כתוצאה מכך, התוכנית לא תוכל ללכוד את כל החבילות ברשת והפונקציונליות שלה תהיה מוגבלת.
ייתכן שהתוכנית תוכל ללכוד רק מנות המיועדות ל host, ולא את כל החבילות המועברות ברשת.

Sniffer

This program captures and analyzes TCP packets on a network. It uses the pcap library to capture packets from a network device.

The main function starts by finding all available network devices and printing them to the console.

The user is then prompted to select one of the devices for sniffing.

The selected device is then opened and a pcap_loop function is used to continuously capture packets and pass them to the processPacket function for processing.

The processPacket function checks the IP protocol of the captured packet, and if it is TCP, it calls the gotPacket function to process the packet.

The gotPacket function prints data from the packet to a text file and to the console.

The program uses several helper functions such as printIpHeader, printEthernetHeader and printData to extract and print relevant data from the packet.

The program continues to capture and process packets in a loop until the program is closed.

חלק ב

Spoofer

שאלה 1: האם אתה יכול להגדיר אורך שדה חבילת ה-ip לאורך שרירותי , ללא קשר לגודל החבילה בפועל?

תשובה: לא ניתן להגדיר את אורך ip לאורך שרירותי.
מכיוון שבשימוש פונקציית sendto השדה הזה נדרס כאשר מעבירים אליו את הגודל.
אבל אם נעביר אליו את השדה של ip length ואז מה שיקרה שהוא יקבע את מספר הבתים שישלחו אליו בפועל (הגודל צריך להיות מינימום 20 בתים כי עבור מתחת לזה יחשב כחבילה שגויה)
ואז הגרעין לא ידע לתרגם את החבילה שנשלחה אליו ואז פונקציית sendto תזרוק שגיאה.

שאלה 2: בשימוש ה-raw socket , האם היית צריך לחשב checksum עבור IP header ?

תשובה: לא היינו צריכים לחשב את הפונקציה הזו כי החישוב מתבצע באופן אוטמטי.
כאשר אנו מעבירים packet לפונקציית sendto , אנו יורדים לרמת הגרעין ושם החישוב מתבצע באופן עצמאי.

Spoofer

This program creates and sends a raw ICMP packet to a specified destination IP address with a specified source IP address.

The program takes two command-line arguments: the IP address of the fake sender and the IP address of the intended recipient.
It creates a raw socket and sets the IP_HDRINCL option to include the IP header in the packet.

Then it constructs both the IP header and the ICMP header and calculates the checksums for each.

Finally, it combines the headers into a single buffer and sends the packet using the sendto function.
To spoof other protocols, we will need to replace the IPPROTO_ICMP constant with the appropriate protocol number and replace the struct icmphdr type.

חלק ג

Docker

SniffAndSpoof

This program is a packet sniffer that captures ICMP packets, specifically ICMP echo request packets, and sends an ICMP echo reply packet in response.

The main function starts by finding all available network devices and printing them to the console. The user is then prompted to select one of the devices for sniffing.

The selected device is then opened and a filter is applied to capture only ICMP packets.

The pcap_loop function is then used to continuously capture packets and pass them to the processPacket function for processing.

The processPacket function checks if the captured packet is an ICMP echo request packet, and if it is, it prints a message to the console and creates an ICMP echo reply packet.

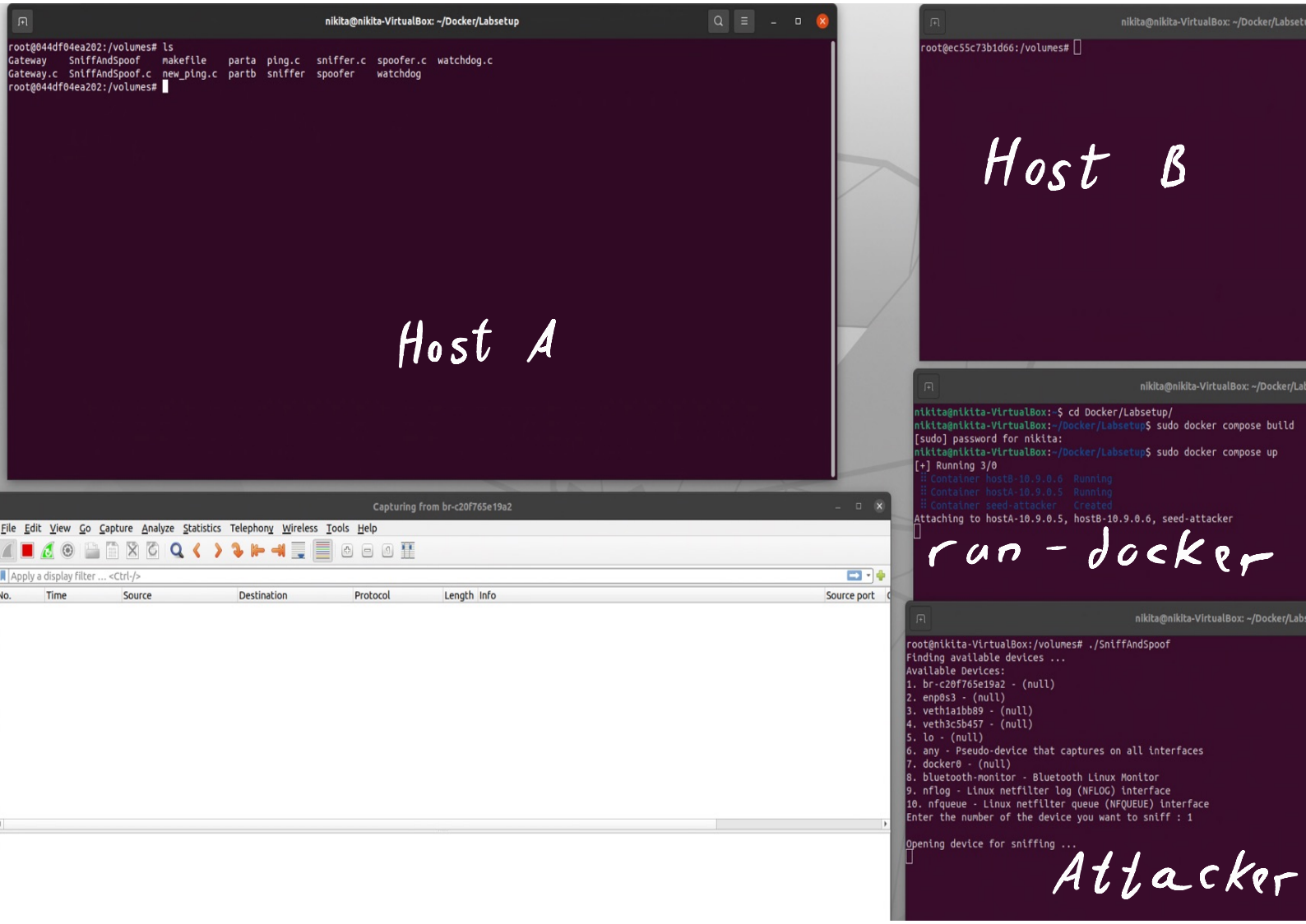
The reply packet is then sent using the sendSpoof function.

The program continues to capture and process packets in a loop until the program is closed.

יצירת הדוקר:

```
nikita@nikita-VirtualBox: ~/Docker/Labsetup
nikita@nikita-VirtualBox:~$ cd Docker/Labsetup/
nikita@nikita-VirtualBox:~/Docker/Labsetup$ sudo docker compose build
[sudo] password for nikita:
nikita@nikita-VirtualBox:~/Docker/Labsetup$ sudo docker compose up
[+] Running 3/0
 ⠿ Container hostB-10.9.0.6  Running                                    0.0s
 ⠿ Container hostA-10.9.0.5  Running                                    0.0s
 ⠿ Container seed-attacker   Created                                    0.0s
Attaching to hostA-10.9.0.5, hostB-10.9.0.6, seed-attacker
```

נכנסו לattacker ע״י : sudo docker exec -it 901672abce3f
את הid קיבלנו מיצירת הדוקר שנמצא בעמוד הקודם.
בקונטיינר הזה התקנו gcc כדי להשתמש בפקודות המייקפייל.
ואז הפעלנו את התוכנית ./SniffAndSpoof ורשמנו 1.
כדי שהוא יכול להסניף את המידע מ host a.
נוכל לראות את זה גם בקבצי pcap של ויירשארק שהוספנו בניפרד.
host a ip -10.9.0.5

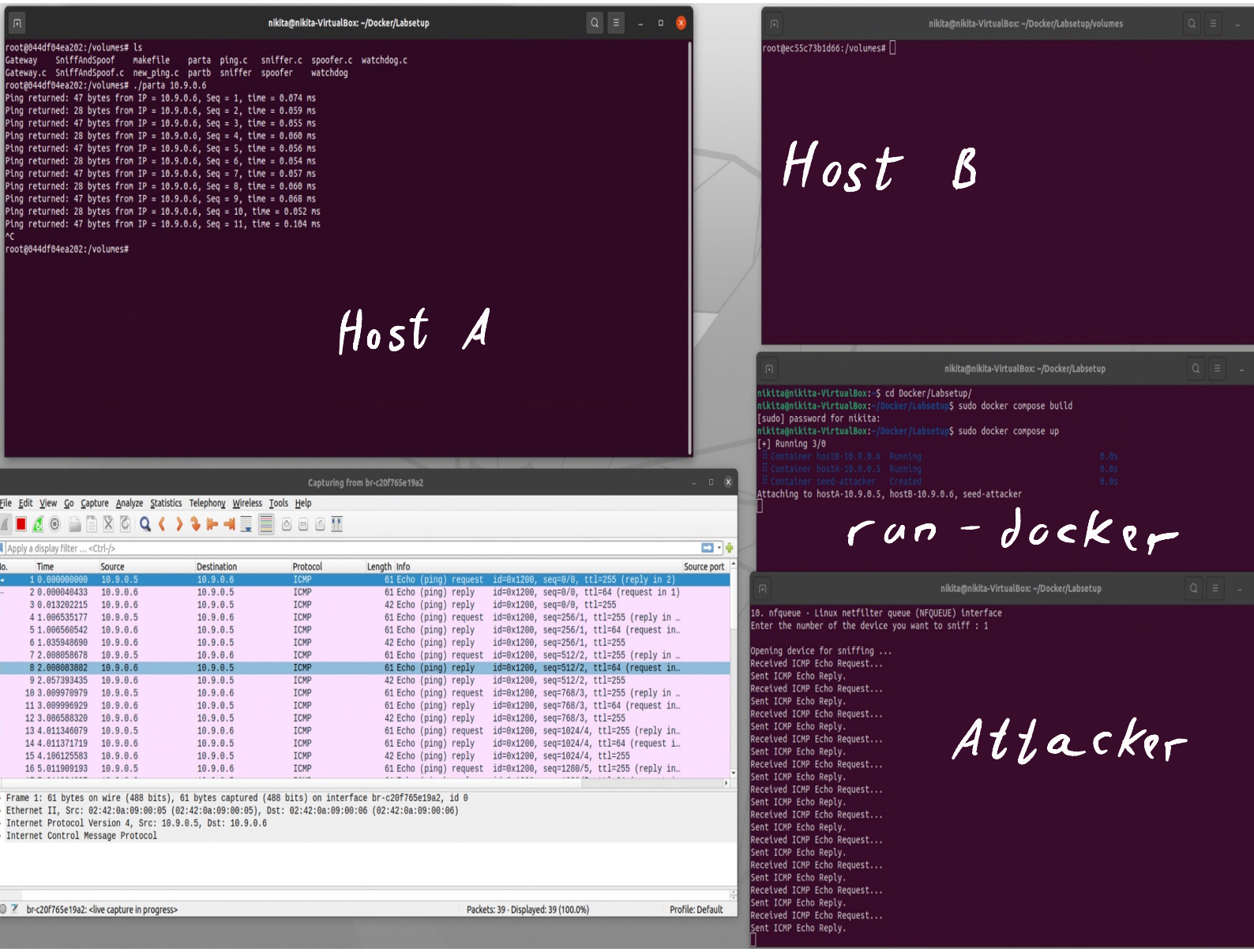


סעיף א:

בתמונה ניתן לראות את הטרמינל של host a (טרמינל מצד שמאל למעלה) שאנחנו מפעילים את תוכנית הפינג ממטלה 4 ע״י parta/. ונותנים לו את כתובת ה-ip של host b :

10.9.0.6

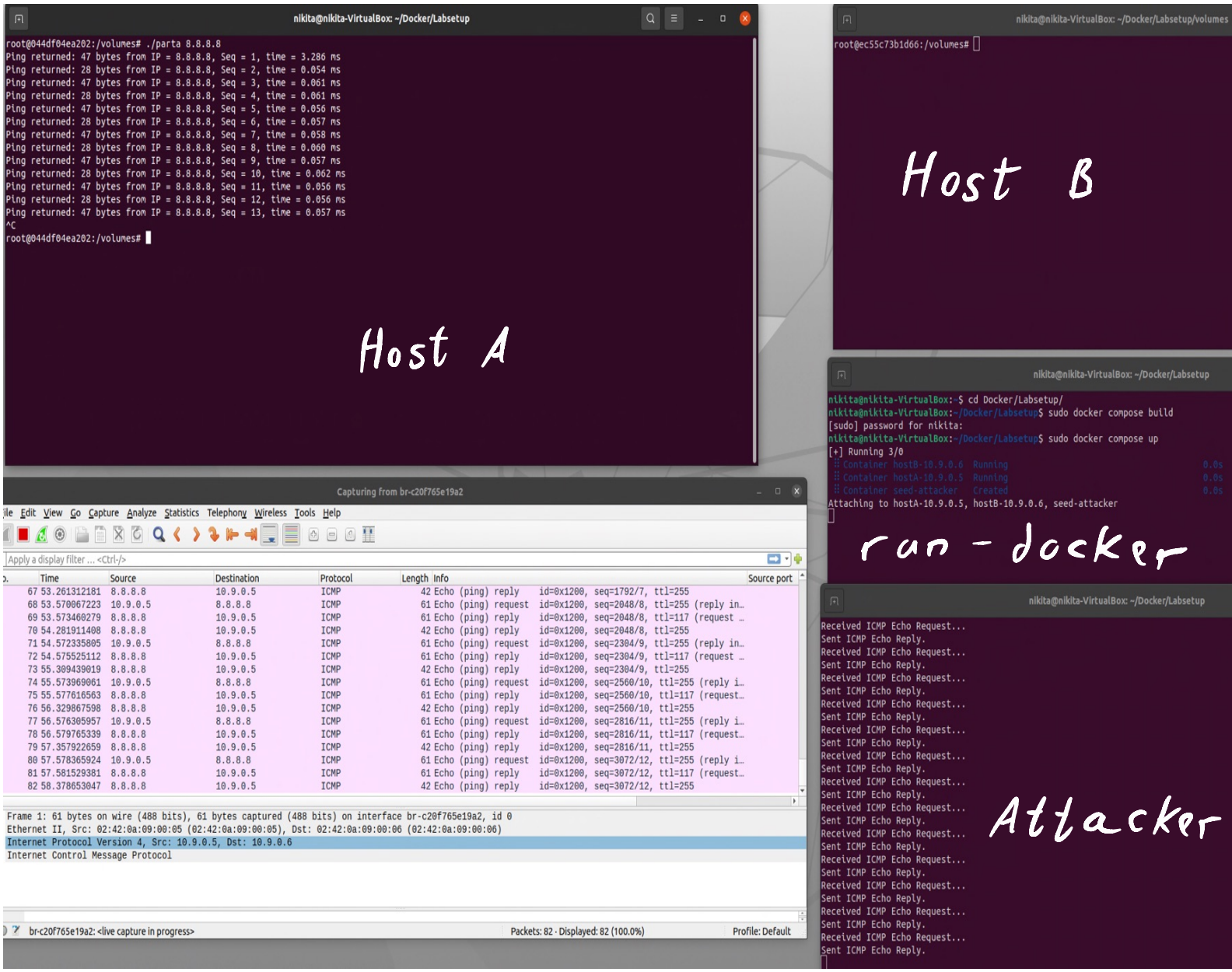
ניתן לראות בויירשארק ש-attacker אכן מצליח להסניף מה-host a את המידע כלומר ניתן לראות שהוא שולח פינג לכתובת הip ומקבל reply.



סעיף ב:

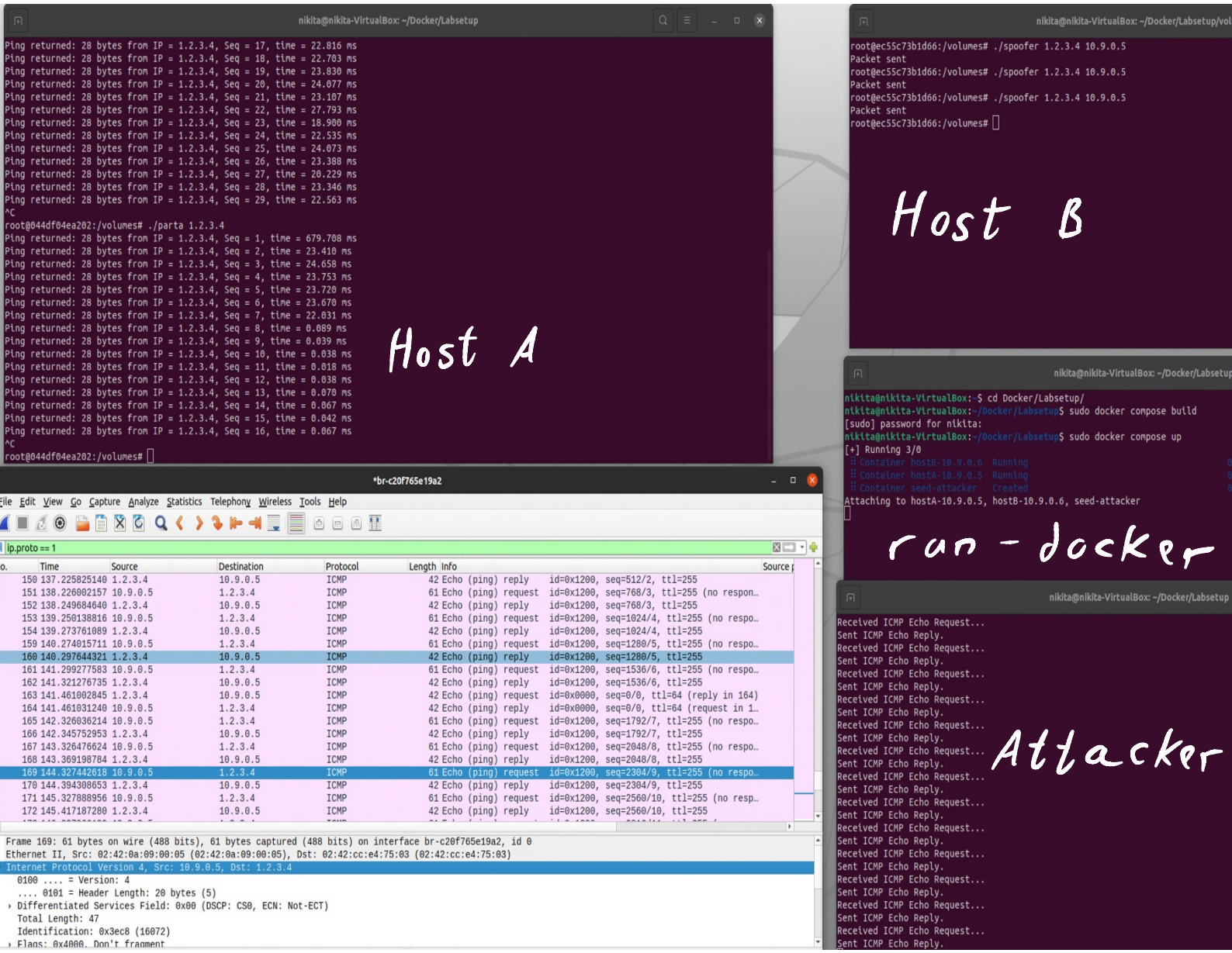
בתמונה ניתן לראות את הטרמינל של host a (טרמינל מצד שמאל למעלה) שאנחנו מפעילים את תוכנית הפינג ממטלה 4 ע״י parta/. ונותנים לו את כתובת ה-ip של גוגל 8.8.8.8

ניתן לראות בויירשארק ש-attacker אכן מצליח להסניף מה-host a את המידע כלומר ניתן לראות שהוא שולח פינג לכתובת הip ומקבל reply.



סעיף ג:

בתמונה ניתן לראות את הטרמינל של host a (טרמינל מצד שמאל למעלה) שאנחנו מפעילים את תוכנית הפינג ממטלה 4 ע״י ./parta ונותנים לו את כתובת ה-ip הלא קיימת לדוגמא 1.2.3.4
בהתחלה אנחנו לא מקבלים תגובה עד שלא מפעילים את תוכנית ה-spoofer בhost b ונותנים לו את הip של host a ולאן הוא רוצה לשלוח את הפינג 1.2.3.4 . התוכנית מצליחה לזייף את הכתובת וזה ניתן לראות בכך שhost a מקבל reply מכתובת 1.2.3.4 למרות שהיא לא קיימת.
ניתן לראות בויירשארק ש-attacker אכן מצליח להסניף מה-host a את המידע כלומר ניתן לראות שהוא שולח פינג לכתובת הip ומקבל reply.



חלק ד

Gateway

Gateway:

This program forwards incoming datagrams on a specific port (port P) to a specified host on another port (port P+1).

The program creates two sockets, one for receiving incoming datagrams on port P and another for sending out datagrams on port P+1.

It then enters an infinite loop where it waits to receive datagrams on the incoming socket. Once a datagram is received, it simulates an unreliable network by discarding the datagram with a 50% probability.

If the datagram is not discarded, it is then forwarded to the specified host on port P+1 using the outgoing socket.
The program exits when the program is closed.

How to run

A Makefile file is attached to the submission files. There are four targets, "sniffer", "spoofer", "SniffAndSpoof", and "Gateway".

To build the program, you can use the command "make all" which will build all the 4 targets
Then use the command "sudo ./<sniffer/SniffAndSpoof>" for the sniffer or the SniffAndSpoof program, or "sudo ./Gateway <host_ip>" for the Gateway, or "sudo ./spoofer <src_ip> <dst_ip>" for the spoofer.

To clean all the targets use command "make clean".