

Task 1.1A:

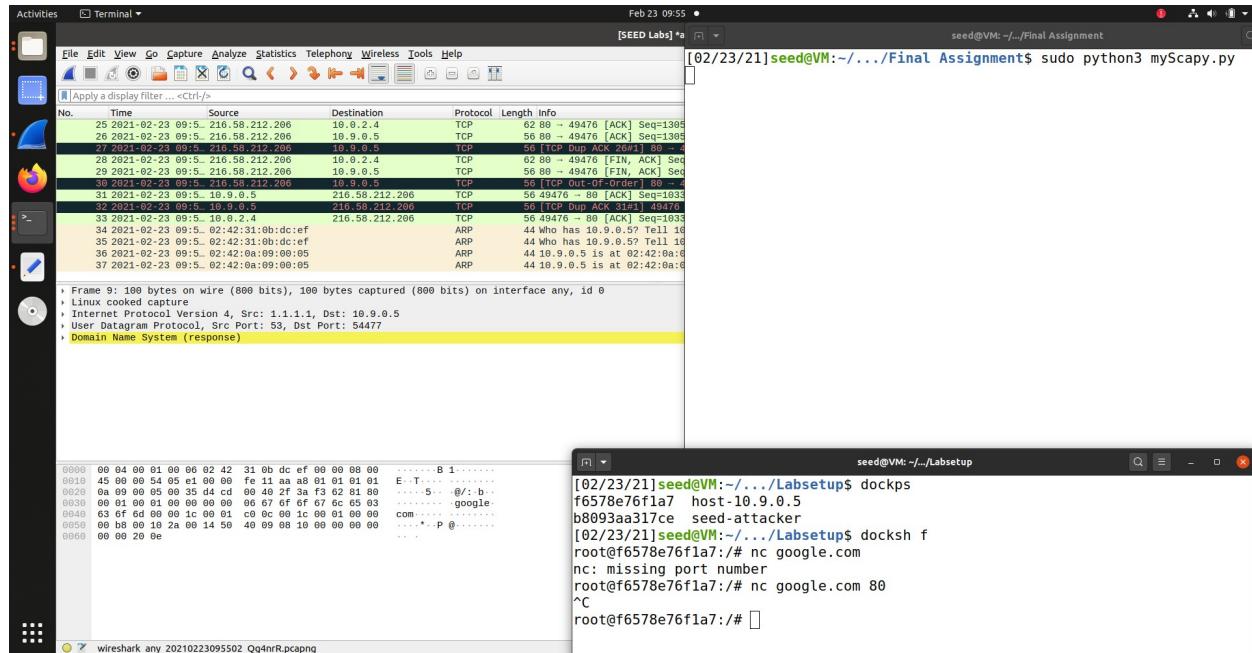
התקשו להריץ SCAPYicum עם הרשאות מנהל (SUDO) ופעם נספת ביל', לאחר השימוש בהראות מנהל אנחנו הצלחנו לתפוס את פקודות ה PING שנשלחו לאוגל וגם את התשובות באמצעות SCAPY, את התוצאות ניתן לראות מטה.

כארר ניסינו להריץ SCAPY בLİ SUDO קיבלונו שגיאיה. הסיבה לכך הייתה השימוש של YOCTO בRAW SOCKET ובדי לבצע זאת צריך גישה בעקיפין למערכת הפעלה לכרטיס הרשות וזה ניתן להשיג רק על ידי שימוש בהרשאות חובה.

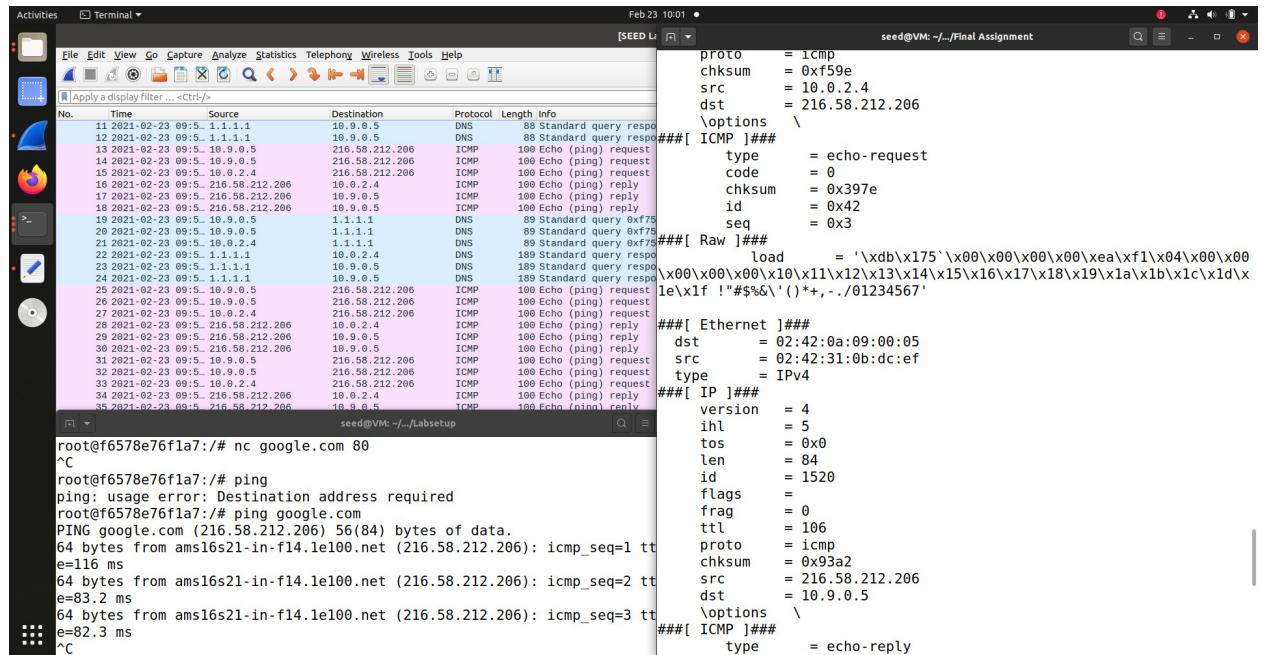
כלומר, כאשר השתמשנו בSOUDS יכולנו להסניף את פקודות ה-ICMP בלי לקבל אף שגיאה. בתמונה מטה אפשר לראות את ניסיון השימוש ב-SCAPY בלי שימוש בהרשאות אדמין וכתוצאה לכך קיבלנו שגיאה.

Task 1.1B:

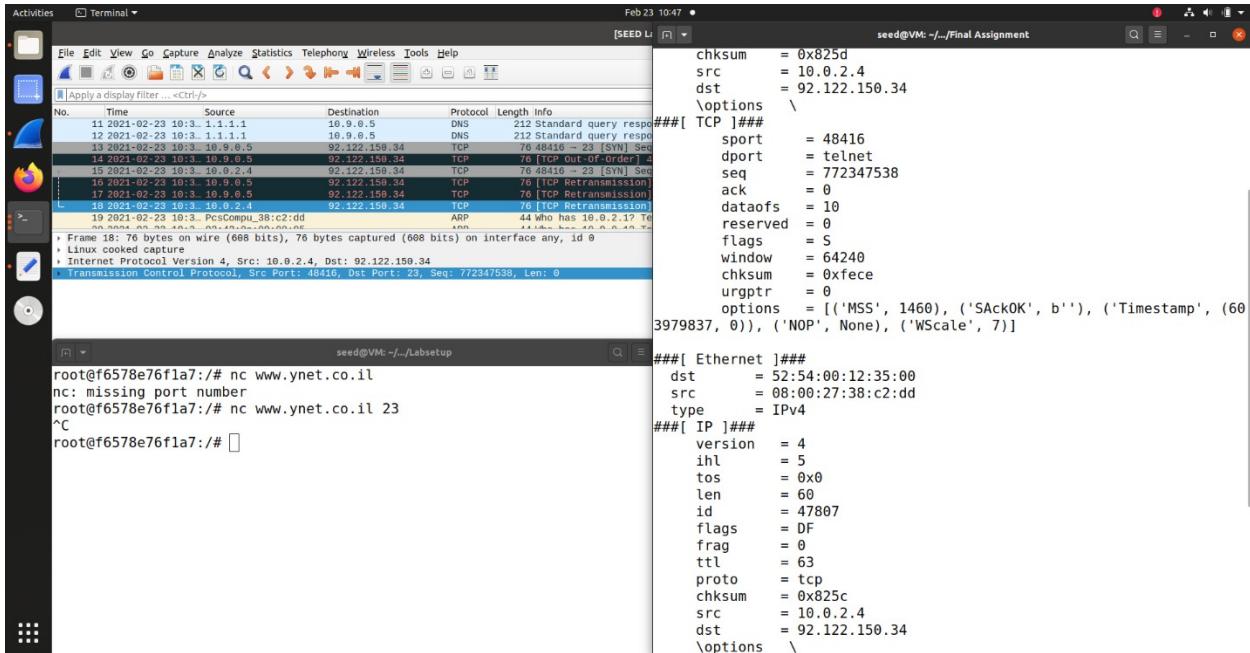
כאן הוכיחנו להשתמש בפילטר כדי לתפוס:
רף פקודות ICMP – כפי שניתן לראות, לאחר הריצת SCAPY ושימוש בNETCAT כדי להשיג את האתר של גולി שימוש בPING, לא הצליחו למצאו אף פקטה עם SCAPY.



כשר השתמשנו בPING הצלחנו לתפוס את פקודות ICMP (אם הביקשות - קוד 9 וטאגובות קוד 0) באמצעות SCAPY.

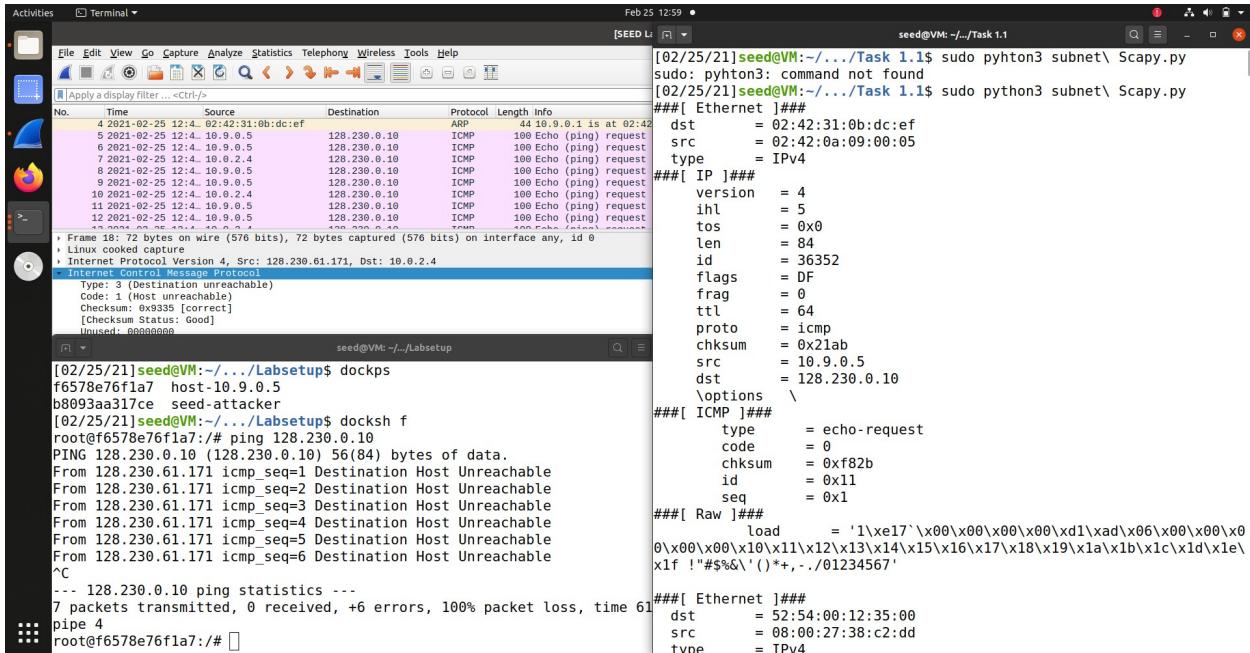


כל פקחת TCP מ IP מקור – כפי שניתן לראות מטה, לאחרת האדרט הפליטר לחיפוש אחר IP בפורט 23, הצלחנו לתפוס את שתי הפקודות אשר נשלחו אל 10.0.CO.YN (אחת מהן מסומנת והשנייה מעלייה ב痼ע אפור).



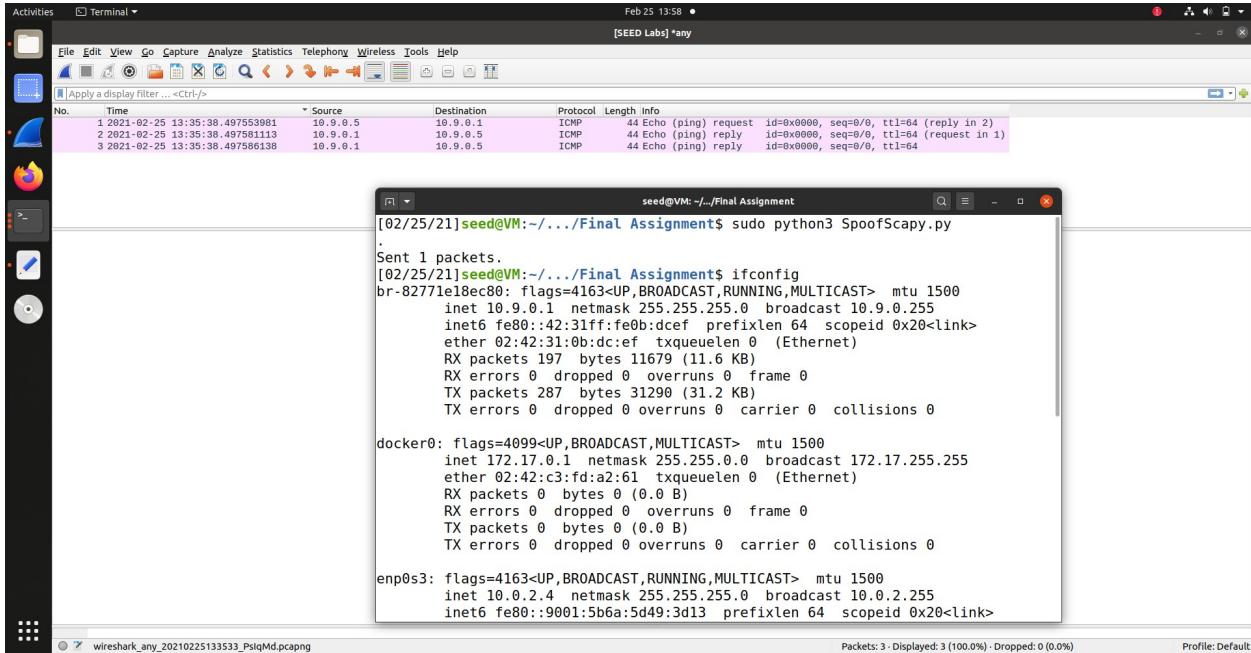
פקטוֹת שמגִיאוֹת מֵאוֹ הַולְכָות לְSUBNET סְפִיצִי – החלטנו לנסוט לטפוס פקטות שנשלחות לHOST 128.230.0.10, אל SUBNET 128.230.0.0.

כפי שניתן לראות בתמונה מטה, SCAPY תופס את הפקודות שהולכות לHOST ב-SUBNET הרצוי.



Task 1.2 :

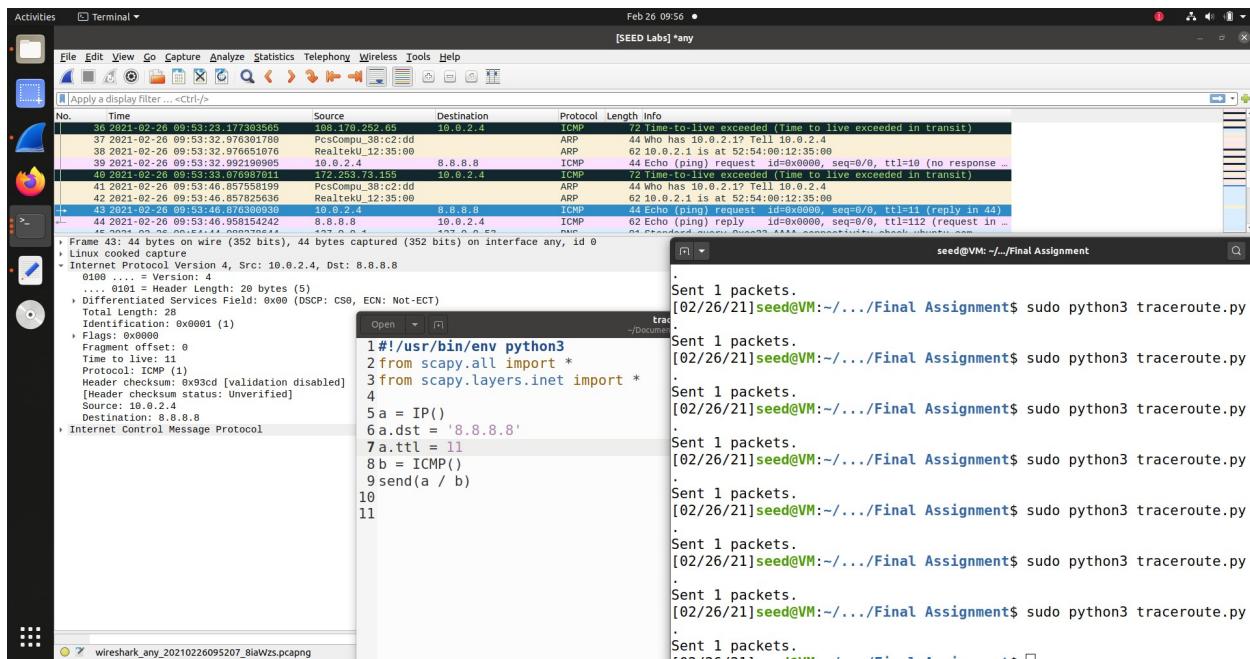
שלחנו את הפקטה מהמכונה הווירטואלית (IP: 10.0.2.4) תוך זיהוי כתובת מקור ל 10.9.0.5 והזיהוי מטר הHOST 10.9.0.1 וניתן לראות את בתמונה מטה שהפקטה התקבלה והודעתה REPLY יוצאה מטר הHOST 10.9.0.1 בכתובת 10.9.0.5 ונכנסת לCONTAINER של התוקף בכתובת 10.9.0.1 CONTAINER .



Task 1.3 :

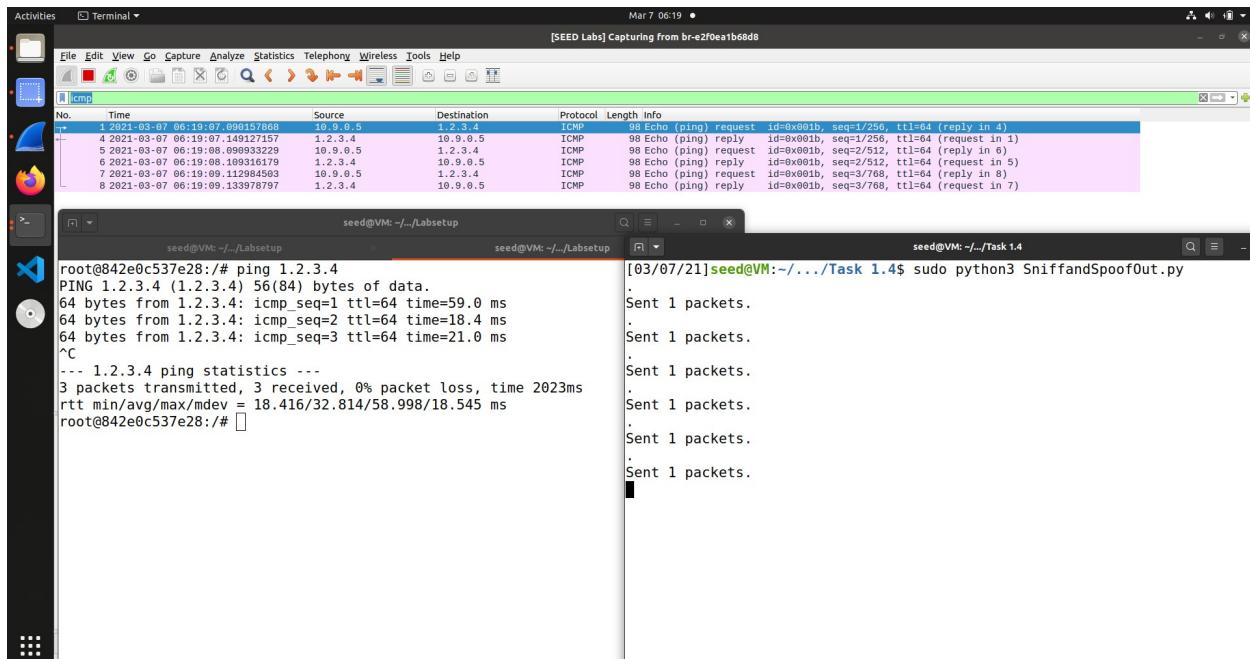
. TRACEROUTE התקבשנו ליזור

אנחנו עונים על השאלה הזאת ידנית כי לא ממש הצלחנו להבין איך לתפוס ICMP REPLY . כמובן שגם בצוירה ידנית זה עובד (PCAP עם 9 TTL בוצע וצורף).

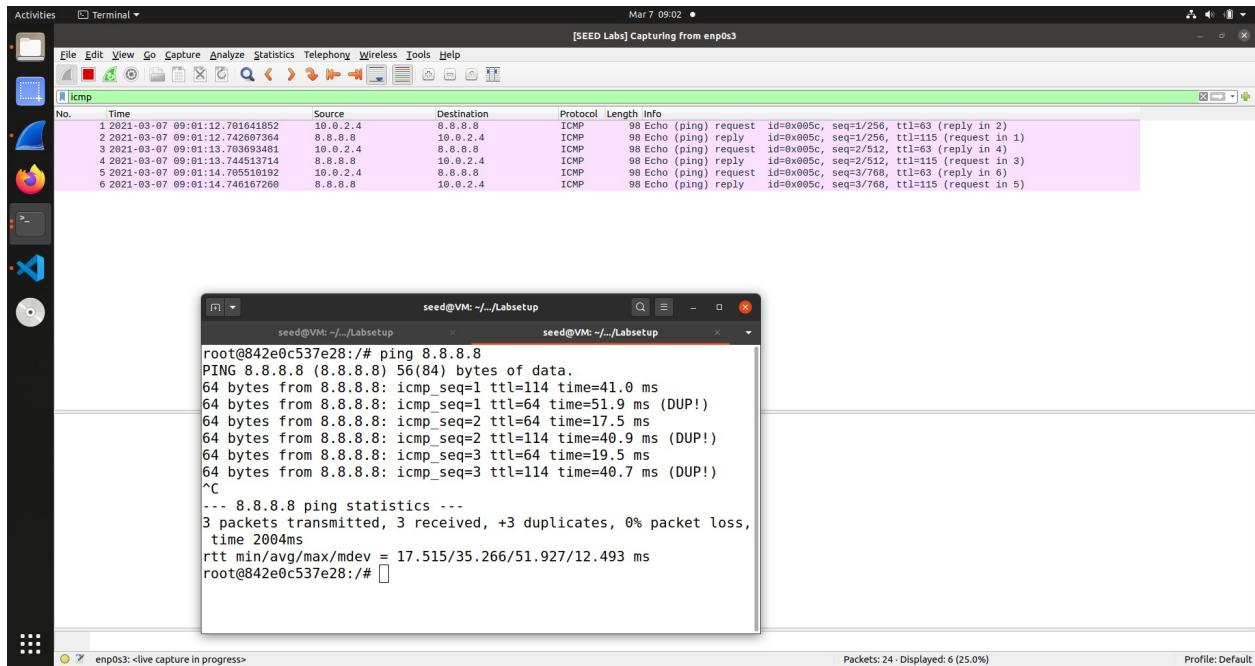
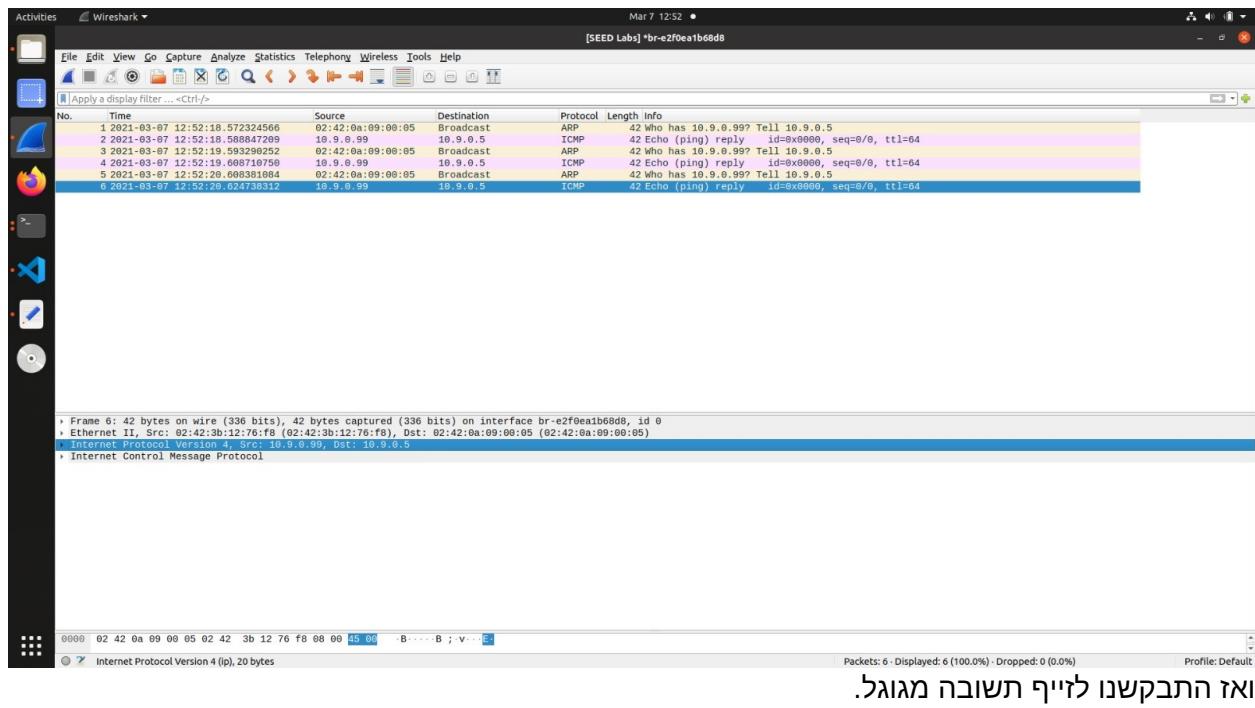


: Task 1.4

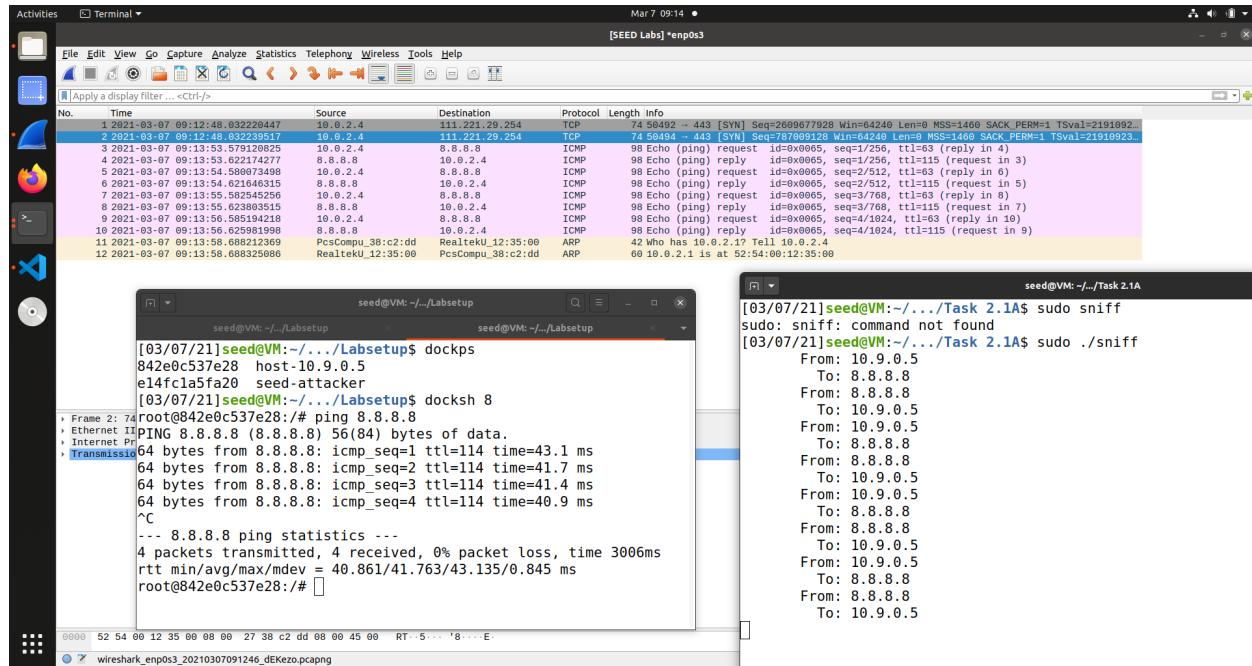
התקבשנו ליזף תשובה לping שנשלח לכתובת שמשתמש קצה שלא קיים באינטרנט.



לאחר מכן התקבשנו ליזף תשובה לping לכתובת **מקומית** שלא קיימת. זאת עשינו על ידי האזנה לביקשות ARP ויזיפ תשובה על ידי ICMP REPLY. הרצינו את honeypot על הרשות של הדוקר המכיל את הכתובות 10.9.0.1 - 10.9.0.5 וניתן לואות בתמונה שהצלחנו ליזף ICMP מכתובת מקור (שלא קיימת) 10.9.0.99.



Task 2.1A :



שאלה 1 – הפענץיה הראשונה נקראת `pcap_open_live` שבה משתמשים כדי לפתח SESSION חדש על הממשק (NIC) הנוכחי ב `promiscuous mode`.

לאחר מכן השתמשנו ב `pcap_compile` כדי להמיר את הфиילטר של המחרוזות למשהו ש ה BPF יכול להבין ולהשתמש בו.

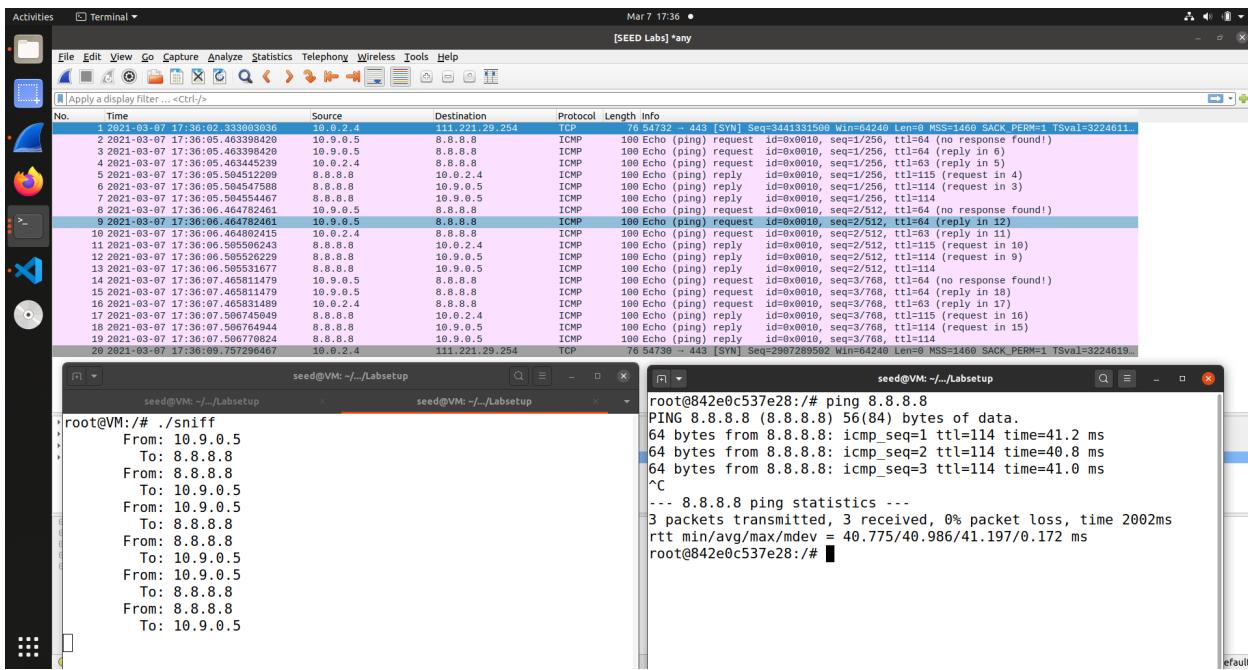
לאחר מכן, הכוונו את הфиילטר על הסקוט שיאין לטעבורה על ידי השימוש ב `loopcap` לכמה מוגבלות של זמן שהפענץיה עצמה קיבלהマイתו. `loopcap` יפנה לפענץיה שאנו חנו שעתה את הפקודות שהצליחה לתפוס .

לאחר כל זה, `pcap_close` תסגור את כל הסקוטים בפרט ואת כל SESSION בכללי כדי למנוע דליפות.

שאלה 2 – אנחנו צריכים מנהל כדי להפעיל `promiscuous mode` ו `RAW SOCKETS` אשר דושים הרשאות מנהל כדי לעקוף את הגדרות האוטומטיות של מערכת הפעלה.

אם נרץ ללא הרשות מנהל `pcap_open_live` יתן שגיאה כי ננסה לקבל גישה להגדירות מותנות הרשות מנהל ללא הרשות מנהל .

שאלה 3 – כאשר הפעילנו את הסניף במצב ניטור, התוכנה פעלה וקיבלו את התוצאה הבאה:



כיבוי PROMISCUOUS MODE אפשרי ע"י פונקציה `PCAP_OPEN_LIVE` ניתנת לכבות אותו ע"י איפוס הפקטרומטר המתקבל `promisc`. כאשר PROMISCUOUS MODE דלוק ניתן לראות את כל התעבורה העוברת ב LAN, כולל לראות פקודות שלא מיועדות ל MAC של המחשב בו משתמשים כאשר PROMISCUOUS MODE כבוי ניתן רק את התעבורה המזענדת לכרטיס הרשות ספציפית של המחשב. ניתן לראות ההבדל בין PROMISCUOUS MODE דלוק לכבוי בכך שכזה דלוק במחשב א' ניתן לשולח פקודות ICMP ממחשב ב' למחשב ג' וממחשב א' יהיה מסוגל לצפות ולהסניף אותם מה שCOMMNON לא אפשר במצב בו MODE PROMISCUOUS כבוי.

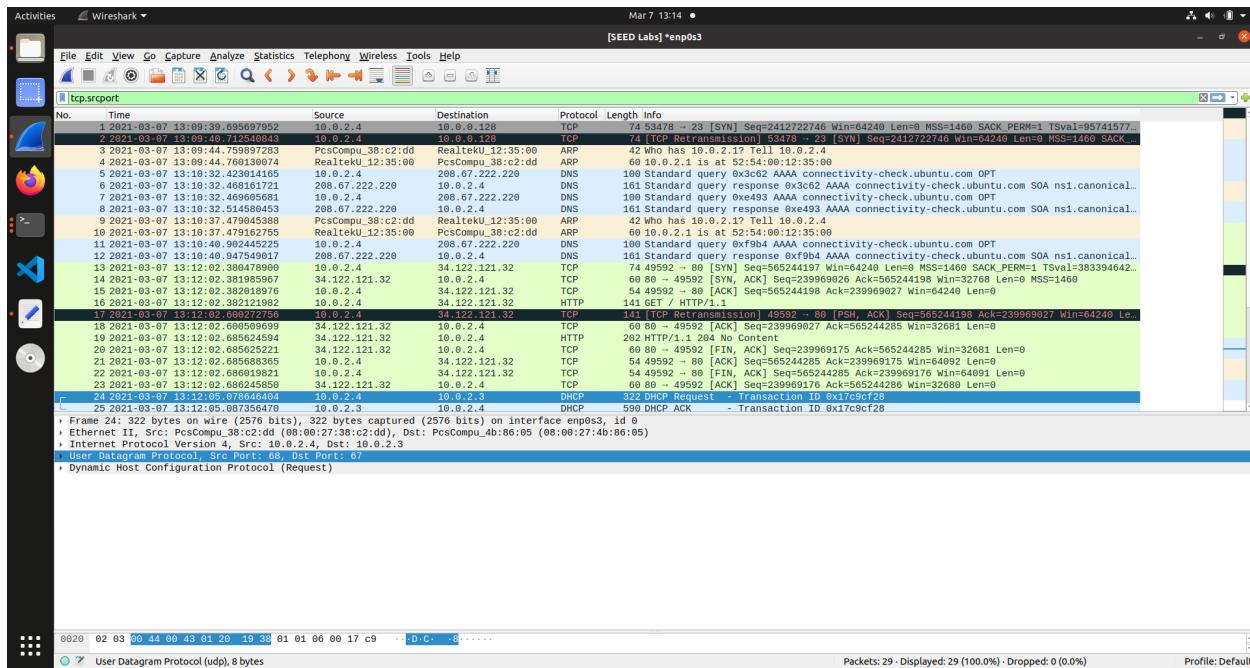
Task 2.1B :

ICMP :

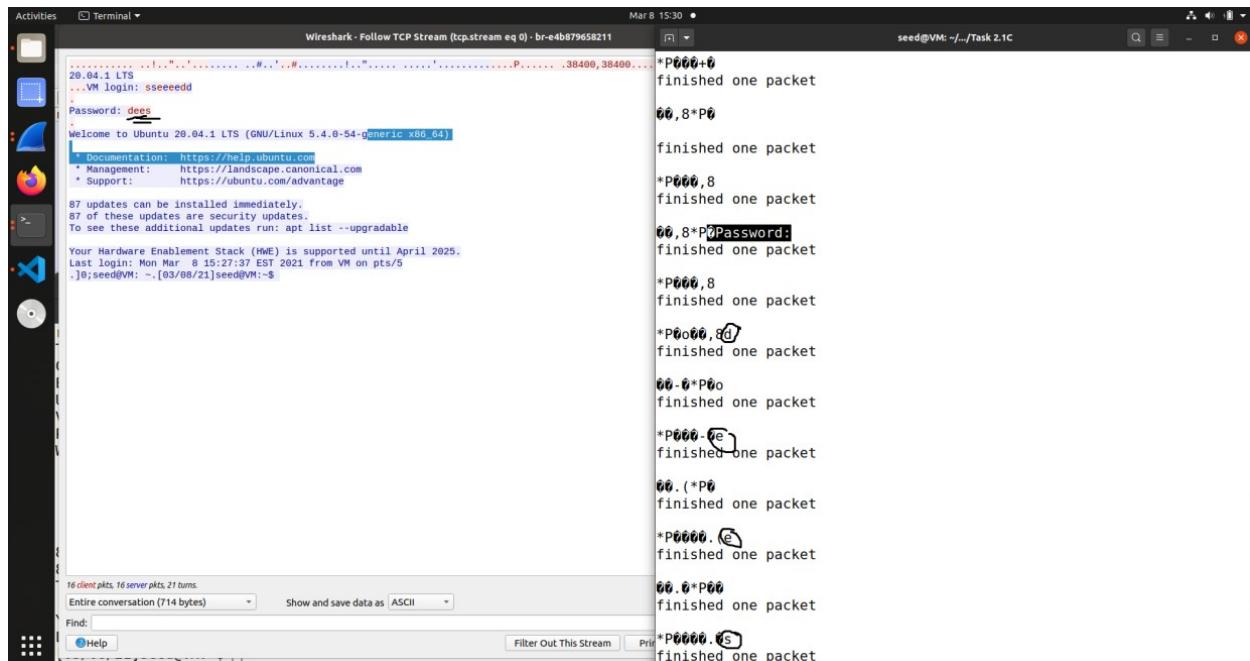
הסנונו דרך כרטיס הרשת של `root@VM` את הפקודות ICMP שיזמיצו או נכנסות מל-10.9.0.5 ל-10.9.0.2 (כתובת IP של גול). פעולה ההסנה עבדה במסגרת promiscuous mode כלומר התוכנה ארגמה לכרטיס הרשת לראות את כל התעבורה העובר ב-LAN אפלו אם לא מועד לכרטיס זהה ספציפית והדפיסה את התעבורה אשר יצאה או נכנסה מ-10.9.0.5.

TCP :

יש גם תפיסת פורטימן נוספת:



Task 2.1C :

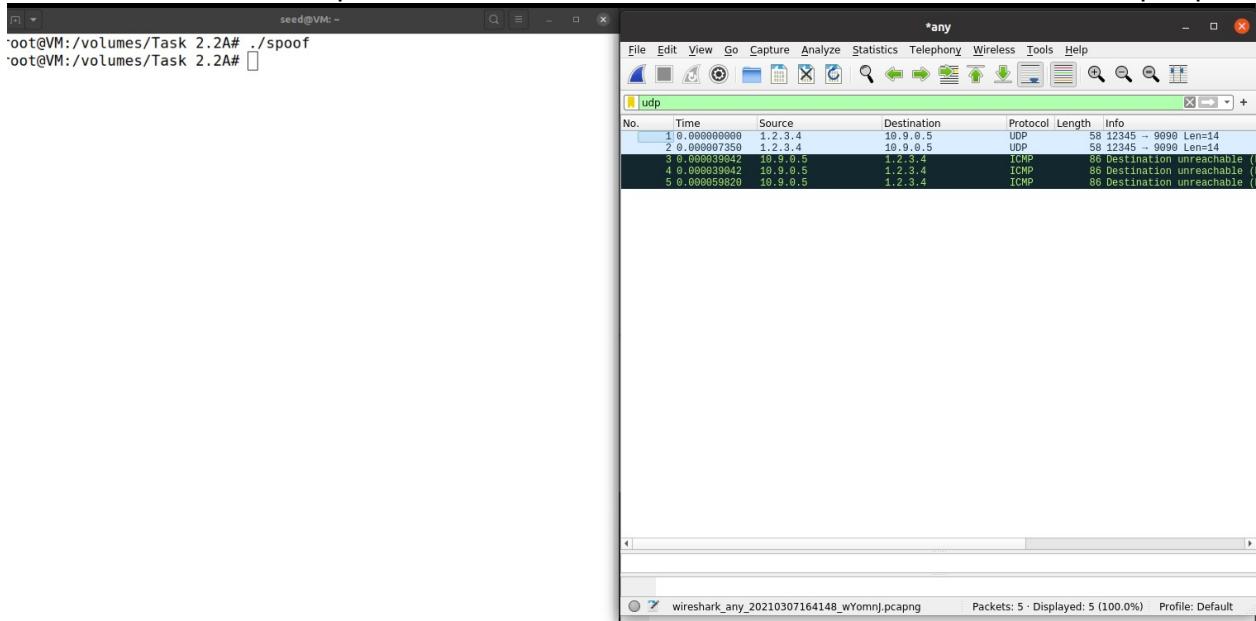


כשנסנו לדוקר וממנו הרצינו את פקודת telnet לכתובת של ה VM. לאחר הכניסה לדוקר, הפעלנו את הסניפר והרצינו במקביל את wireshark ניתן לראות בתמונה מסמאל את הסיסמה שלחנו דרך telnet ומימין ניתן לראות את הפלט של הניספר אשר קלט את הסיסמה.

Task 2.2A :

```
root@VM:/volumes/Task 2.1B# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    group default qlen 1000
        link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
            inet 127.0.0.1/8 scope host lo
                valid_lft forever preferred_lft forever
            inet6 ::1/128 scope host
                valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel
    state UP group default qlen 1000
        link/ether 08:00:27:f5:67:b9 brd ff:ff:ff:ff:ff:ff
            inet 10.0.2.5/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
                valid_lft 538sec preferred_lft 538sec
            inet6 fe80::6b84:ecab:2915:f751/64 scope link noprefixroute
                valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:0e:7a:ed:b4 brd ff:ff:ff:ff:ff:ff
        inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
            valid_lft forever preferred_lft forever
5: br-1ca35f87b2fa: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:85:23:b1:ea brd ff:ff:ff:ff:ff:ff
        inet 10.9.0.1/24 brd 10.9.0.255 scope global br-1ca35f87b2fa
            valid_lft forever preferred_lft forever
        inet6 fe80::42:85ff:fe23:b1ea/64 scope link
            valid_lft forever preferred_lft forever
7: vethd06e113@if6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
root@2c0f11b59b56:# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    group default qlen 1000
        link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
            inet 127.0.0.1/8 scope host lo
                valid_lft forever preferred_lft forever
6: eth0@if7: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:09:00:05 brd ff:ff:ff:ff:ff:ff link-layer
        inet 10.9.0.5/24 brd 10.9.0.255 scope global eth0
            valid_lft forever preferred_lft forever
root@2c0f11b59b56:#
```

כאן ניתן לראות את כתובות ה IP ופרטי הכרטיסי הרשות שיש לכל אחד מהקונטינרים .



כמו שניתן לראות התוכנה מבצעת ספופינג כר שהיא משנה את כתובות המקור ל 1.2.3.4 כשהכתובה האמיתית היא 10.0.2.5 ושולחת הודעת ICMP מסוג 8 לכתובה הזו, כר שהכתובה המקורי של המחשב לא תהיה ניתנת אליו. בעצם שלחנו הודעת ICMP מהמחשב לעצמו אבל כתובות מקור הסתכו ממסק ANY ואנו יכולים לראות שהפקטה במעבר מ 1.2.3.4 אל 10.9.0.5 נרשמה פעמים זאת מהתיבה שהיא נרשמה פעם אחת כשייצאה ממחשב התוקף ופעם שנייה כשנכנסה למחשב הנטקף. פקעת התשובה של ICMP שאותו משליח הפה לא מתקבל נרשמה לשושה פעמים, זאת מהתיבה שהיא פעם אחת נרשמה כשייצאה מהמחשב הנטקף, פעם שנייה כשייצאה מחוץ לחיבור LAN כדי להביא את החבילה ל IP חיצוני ופעם

שלישית כאשר הפקטה עברה ב SUBNET , כלומר עבר כל ממשק שבו עברה הפקטה היא השאיתה. תיעוד זאת מהסיבה שהשתמשנו בממשק ANY שנותן לנו פרנספקטיבה על המתרחש ברשת הפנימית. כלומר התבוננות דרך הממשק ANY נותן לנו את האפשרות לנטר את כל תהליך מעבר הפקות.

Task 2.2B :

The screenshot shows two terminal windows and a Wireshark capture window. The top-left terminal window shows the command `./spoofIcmpEcho` being run. The top-right terminal window shows the output of the `ip a` command, listing network interfaces lo, eth0, and eth0:1. The bottom window is Wireshark displaying an ICMP echo request and response between source 1.1.1.1 and destination 10.9.0.5.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	1.1.1.1	10.9.0.5	ICMP	44	Echo (ping) request id=0x0000 seq=0/0, ttl=20 (no response found)
2	0.000047018	1.1.1.1	10.9.0.5	ICMP	44	Echo (ping) request id=0x0000 seq=1/0, ttl=20 (reply in 3)
3	0.000047018	10.9.0.5	1.1.1.1	ICMP	44	Echo (ping) reply id=0x0000 seq=0/0, ttl=64
4	0.000047018	10.9.0.5	1.1.1.1	ICMP	44	Echo (ping) reply id=0x0000 seq=1/0, ttl=64
5	0.000069499	10.9.0.5	1.1.1.1	ICMP	44	Echo (ping) reply id=0x0000 seq=0/0, ttl=64

כמו שניתן לראות ב WIRESHARK נתפסה פקחת ICMP ECHO שלושה פעמים כי התוכנה עבדה מהממשק של כרטיס הרשת הכללי שראה גם את הכרטיס של התקשורת ואם של הנתקף כלומר פעם אחת הפקטה נתפסה עבור הממשק של כרטיס הרשת של התקשורת ופעם אחת עבור כרטיס הרשת של הנתקף , לאחר מכן נשלחה הודעת REPLY שניתן לראות אותה 3 פעמים , זהה מהסיבה שפעם אחת היא נרשמה כשהיא יצא מהמחשב הנתקף , פעם אחת שהתקבלה במחשב התקשורת ועוד פעם אחת שהיא נתפסה כ REPLY בממשק של הכרטיס רשת הכללי.

שאלה 4 - כן, מבחינה טכנית אפשר לשנות גודל של פקטה לכל אובל בסופו של דבר זה יוחזר לאודל שהיא בהתחלה. ניתן להגיד אובל שירוטי אובל בגבול מסוים. צירר בין המינימום יהיה 20 byte ועד מקסימום של 65535 byte .

שאלה 5 - לא צריך לבצע את החישוב של ה CHECKSUM בשביל שליחת הפקטה , מערכת ההפעלה ביצורה דיפולטיבית מחשבת אותה בשביבו.

שאלה 6 - אנו צריכים הרשות שורש (ROOT/SUDO) בשביל להעביר את הכרטיס רשות ל mode Promiscuousraw socket ננסה להגדיר את ה Raw socket , אם ננסה להריץ בלי הרשות התכניתית טיפול בזמן הדרת ה SOCKET כי האדרת דברים כמו IP מקור או גישה לפורט 0 - 1024 זה מאוד רגיש ונמצא תחת סמכותה של מערכת ההפעלה.

Task 2.3 :

No.	Time	Source	Destination	Protocol	Length	Info
1	0.0000000000	10.9.6.5	1.1.1.1	ICMP	80	Echo (ping) request
2	0.0554446811	1.1.1.1	10.9.6.5	ICMP	80	Echo (ping) reply
3	0.1108893622	1.1.1.1	10.9.6.5	ICMP	80	Echo (ping) reply
4	0.1663246776	10.9.6.5	1.1.1.1	ICMP	80	Echo (ping) request
5	0.1870687671	1.1.1.1	10.9.6.5	ICMP	80	Echo (ping) reply
6	0.2425037586	1.1.1.1	10.9.6.5	ICMP	80	Echo (ping) reply
7	0.2982455198	10.9.6.5	1.1.1.1	ICMP	80	Echo (ping) request
8	0.3238933183	1.1.1.1	10.9.6.5	ICMP	80	Echo (ping) reply
9	0.3793212747	1.1.1.1	10.9.6.5	ICMP	80	Echo (ping) reply
10	0.4348532047	10.9.6.5	1.1.1.1	ICMP	80	Echo (ping) request
11	0.4803522431	1.1.1.1	10.9.6.5	ICMP	80	Echo (ping) reply
12	0.5358442431	10.9.6.5	1.1.1.1	ICMP	80	Echo (ping) request
13	0.5813242431	1.1.1.1	10.9.6.5	ICMP	80	Echo (ping) reply
14	0.6368042431	10.9.6.5	1.1.1.1	ICMP	80	Echo (ping) request
15	0.4548591999	1.1.1.1	10.9.6.5	ICMP	80	Echo (ping) reply
16	0.0508959999	10.9.6.5	1.1.1.1	ICMP	80	Echo (ping) request
17	0.1063342926	1.1.1.1	10.9.6.5	ICMP	80	Echo (ping) reply
18	0.4538262617	1.1.1.1	10.9.6.5	ICMP	80	Echo (ping) request
19	0.6973439296	10.9.6.5	1.1.1.1	ICMP	80	Echo (ping) reply
20	0.851950612	1.1.1.1	10.9.6.5	ICMP	80	Echo (ping) reply
21	0.9065989897	1.1.1.1	10.9.6.5	ICMP	80	Echo (ping) reply

כמו שניתן לראות מחשב "הקורבן" שולח פנית ICMP אל 1.1.1.1, תוכנת ההסנפה אשר נמצא על מחשב שנמצא באותה רשת LAN רואה זאת ושולח פקעת ICMP REPLY (כלומר ICMP עם קוד 0) מזיהפת עם IP מקור 1.1.1.1.残缺的文本部分未被包含在内，导致上下文不完整。建议将该段落与前文合并以确保完整性。