

### מטלה 3

#### Packet Sniffing and Spoofing

##### תוכן העניינים:

מבוא	עמוד 2
Task a	עמודים 3-4
Task b	עמודים 5-6
Task c	עמודים 7-9

## מבוא:

במטלה זו נדרשנו לממש מימושים שונים של הסנפה וזיוף פקטות בתעבורת רשת. המטלה מחולקת ל4 משימות:

משימה 1 - הינה לכתוב את הקובץ "sniffer.c" המממש מסנן פקטות מסוג tcp , ומדפיס על גבי מסמך טקסט את המאפיינים הייחודיים של כל פקטה .  
נדרשנו להריץ לצורך משימה זו את מטלה 2 .

משימה 2 - הינה לכתוב את הקובץ "spoof.c" המממש מזייף פקטות מסוג icmp, והמאתחל בהתאם לרצונו של המשתמש את כלל השדות הנמצאים בפקטה.

משימה 3 - הינה שילוב בין משימה 1 למשימה 2, כאשר נדרשנו לממש קוד המסנן ומזייף פקטות icmp המועברות ברשת בין מכשירים שונים.  
זיוף זה בא לידי ביטוי באמצעות החלפת כתובת ה ip של היעד בכתובת ה ip של המקור, שינוי סוג הפקטה מ "icmp Echo-request" ל "icmp Echo-reply", וכן שליחת הפקטה לאחר ביצוע פעולת הזיוף אל המקור ממנו נשלחה. על ידי ביצוע פעולות אלו, ה spoofed למעשה מתחזה ליעד אליו ממוענת החבילה.  
את הקוד נדרשנו להריץ באמצעות docker-compose לצורך הדמיית קשר בין מכשירים שונים על אותו LAN.

משימה 4 – הייתה לממש את הקובץ gateway.c באמצעות שימוש בפרוטוקול udp באופן המבטא רשת אינטרנט לא אמינה עם איבוד נתונים בשיעור של כ50%.

## Task a:

### תיאור הקוד:

בקוד זה השתמשנו בספריית pcap.h לצורך מימוש מסניף פקטות מסוג tcp. קבלת הפקטות התבצעה באמצעות הפונקציה pcap\_loop השייכת לספרייה pcap.h. ניתוח המידע שבפקטות התבצע דרך הפונקציה "print\_tcp\_packet". פונקציה זו, המועברת כפרמטר דרך הפונקציה pcap\_loop, מקבלת לתוך הארגומנטים היחודיים לה, את סך כל המידע הנמצא בפקטה המועברת ברשת. את המידע הזה היא מנתחת ומדפיסה בהתאם בקובץ txt מיוחד אשר שמו מורכב מתעודות הזרות שלנו. השתמשנו לצורך גילוי המידע הזה ב struct מיוחד בשם data\_head, שתפקידו הוא להכיל בתוכו את השדות היחודיים לheaders שבמטלה 2, המפורטים בקובץ ההוראות למטלה.

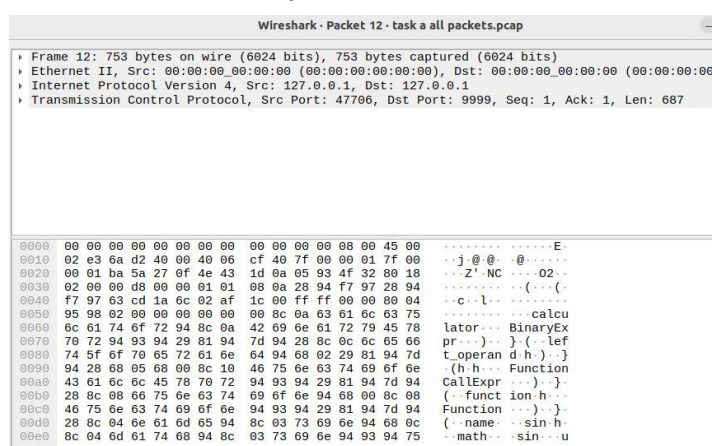
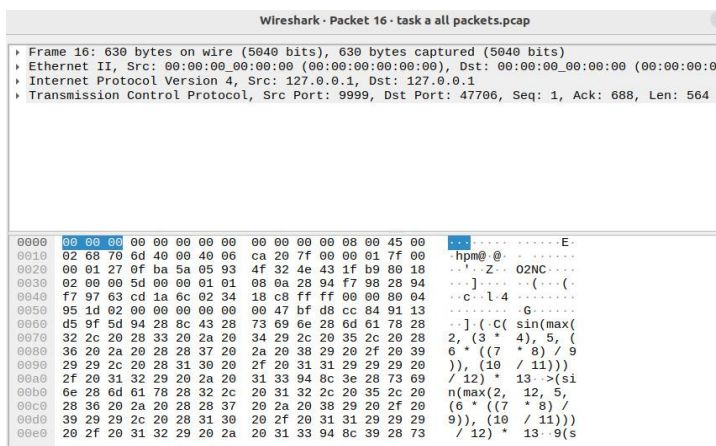
### להלן תיאור הרצת הקוד ע"פ קובץ הטקסט:

```
316276989_212689475.txt
378      NEW TCP PACKET:
379
380      Source IP      : 127.0.0.1
381      Destination IP : 127.0.0.1
382      Source Port    : 9999
383      Destination Port : 47706
384      Time stamp     : 1674386028
385      Total_length of data : 564
386      Cashe_flag     : 1
387      Step_flag      : 1
388      Type_flag      : 0
389      Status         : 200
390      Cache_control   : 65535
391
392      DATA
393
394      63 CD 1A 6C 02 34 18 C8 FF FF 00 00 80 04 95 1D
395      02 00 00 00 00 00 47 BF D8 CC 84 91 13 D5 9F
396      5D 94 28 8C 43 28 73 69 6E 28 6D 61 78 28 32 2C
397      20 28 33 20 2A 20 34 29 2C 20 35 2C 20 28 36 20
398      2A 20 28 28 37 20 2A 20 38 29 20 2F 20 39 29 29
399      2C 20 28 31 30 20 2F 20 31 31 29 29 29 20 2F 20
400      31 32 29 20 2A 20 31 33 94 8C 3E 28 73 69 6E 28
401      6D 61 78 28 32 2C 20 31 32 2C 20 35 2C 20 28 36
```

```
316276989_212689475.txt
267      NEW TCP PACKET:
268
269      Source IP      : 127.0.0.1
270      Destination IP : 127.0.0.1
271      Source Port    : 47706
272      Destination Port : 9999
273      Time stamp     : 1674386028
274      Total_length of data : 687
275      Cashe_flag     : 1
276      Step_flag      : 1
277      Type_flag      : 1
278      Status         : 0
279      Cache_control   : 65535
280
281      DATA
282
283      63 CD 1A 6C 02 AF 1C 00 FF FF 00 00 80 04 95 98
284      02 00 00 00 00 00 8C 0A 63 61 6C 63 75 6C 61
285      74 6F 72 94 8C 0A 42 69 6E 61 72 79 45 78 70 72
286      94 93 94 29 81 94 7D 94 28 8C 0C 6C 65 66 74 5F
287      6F 70 65 72 61 6E 64 94 68 02 29 81 94 7D 94 28
288      68 05 68 00 8C 10 46 75 6E 63 74 69 6F 6E 43 61
289      6C 6C 45 78 70 72 94 93 94 29 81 94 7D 94 28 8C
290      08 66 75 6E 63 74 69 6F 6E 94 68 00 8C 08 46 75
```

בתמונות אלו ניתן להבחין בבקשה מצד לקוח כלפי השרת באורך 687 bytes ובתשובת שרת ללקוח באורך 564 bytes.

### תיאור הרצת הקוד על פי wireshark:



בהקלטות אלו ניתן לראות את המידע על חבילת בקשה זו, וכן את המידע על חבילת התגובה לבקשה זו.

### Task a – continue:

לצורך הפעלת ה sniffer עלינו להשתמש בהרשאת מנהל, מכיוון שהפונקציות הממומשות בספריית pcap.h מבוססות על שימוש ב raw socket .  
ללא שימוש בהרשאת מנהל, התוכנה תקרוס כבר בשלבים הראשונים של הקוד.

יכולותיו של ה sniffer הן הסנפת פקטות המועברות ברשת וניתוח המידע השייך להן. בעזרת ניתוח זה ניתן לפתור בעיות שונות ברשת, לעקוב באופן צמוד אחר פעילות הרשת על מנת לוודא את תקינותה, ואף לזהות בעיות אבטחה במידת הצורך.

ל sniffer ישנן מספר מגבלות.

- 1) הוא אינו יכול לדעת בהכרח אם הפקטות אותן הוא מסניף הן מזויפות או אמיתיות.
- 2) הוא עלול להיות בעייתי בכל הנוגע לניתוח תעבורה מוצפנת.
- 3) ניתן לזיהוי וחסימה על ידי מערכות אבטחת רשת.

## Task b:

במשימה זו נדרשנו לממש תוכנה המזייפת פקטות icmp ברשת. תוכנה זו יוצרת למעשה פקטה חדשה, כאשר היא מאתחלת בהתאם לרצון המשתמש, את כלל השדות האפשריים בפקטה, הניתנים לעיצוב כפי רצונו.

### תיאור הקוד:

בקוד זה אנו משתמשים למעשה בשני structs (מבנים) שונים. struct ipheader - המכיל מצביעים לכלל השדות הנמצאים ב header של הפקטה. struct icmpheader - המכיל מצביעים לכלל השדות הנמצאים ב icmp-header של הפקטה. כמו כן נעשה אצלנו שימוש בפונקציית "in\_chksum" המחשבת את שדה ה checksum עבור כל פקטה אותה אנו יוצרים. באמצעות השדות הנמצאים במבנים שברשותנו ובפונקציה "in\_chksum" אנו למעשה יוצרים פקטה חדשה כרצוננו. לאחר מכן אנו שולחים אותה ליעדה באמצעות שימוש ב raw-socket.

### תיאור הרצת הקוד על פי ה wireshark:

No.	Time	Source	Destination	Protocol	Length	Info
29	6.827970434	8.8.8.8	10.0.2.15	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=20
30	6.827982811	10.0.2.15	8.8.8.8	ICMP	44	Echo (ping) reply id=0x0000, seq=0/0, ttl=64

בתמונה זו ניתן לראות כיצד זייפנו פקטה מסוג icmp Echo-request המציגה את ה source ip של הפקטה בתור 8.8.8.8 השייכת לגוגל, וכן כיצד התקבלה תגובה לבקשה מזויפת זו.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	1.2.3.4	10.0.2.15	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=20
2	0.000011	10.0.2.15	1.2.3.4	ICMP	44	Echo (ping) reply id=0x0000, seq=0/0, ttl=64

בתמונה זו ניתן להבחין כיצד התבצע זיוף פקטה מסוג icmp Echo-request עם כתובת source-ip 1.2.3.4, וכיצד התקבלה תגובה לבקשה זו.

את שדה האורך השייך לפקטה אנו יכולים לזייף כרצוננו, ולכן ניתן להכניס לו אילו ערכים שאנו רוצים, כל עוד ערכים אלו נמצאים במקומם השמור להם ב header של הפקטה ואינם חורגים מגודלה המקסימלי האפשרי של הפקטה. במקרה של חוסר התאמה בין גודלה האמיתי של הפקטה לבין ערך הגודל שלה כפי שהוא זויף, ייתכן שחלק מנתבי הרשת יחסמו את תעבורת הפקטה.

בכל פקטה אותה אנו יוצרים באמצעות raw-sockets, עלינו לחשב עבורה את השדה check\_sum, על מנת שהביצועים ואופן השליחה של הפקטה יהיו תקינים.

Task b – continue:

ל spoofer אותו יצרנו ישנן מגוון יכולות:

- 1) הוא יכול לשבש פעילות רשת תקינה באמצעות זיוף נתונים קריטיים המצויים בפקטות הנמצאות בתעבורת הרשת.
- 2) הוא יכול לאפשר התחזות לגורמים מסוימים ברשת בהתאם לאינטרסים של מזייף הפקטות.
- 3) הוא יכול להסתיר את זהותו האמיתית של המזייף, באמצעות שינוי כתובת המקור של החבילה.

מגבלותיו של ה spoofer הן:

- 1) ניתן לזיהוי על ידי אמצעי הגנה שונים ברשת.
- 2) עלול להיחסם על ידי אמצעי אבטחה שונים ברשת.
- 3) את חלק מהפקטות הוא עלול לא להצליח ללכוד, בין היתר בשל אמצעי אבטחה והצפנה שונים.

### Task c:

במשימה זו נדרשנו לממש קוד המסניף פקטות icmp מסוג "icmp Echo-request" ולאחר מכן מזייף אותן.

זיוף זה אמור לבוא לידי ביטוי באמצעות החלפה בין כתובת היעד לכתובת המקור של הפקטה המקורית, וכן שינוי סטטוס הפקטה, מ"בקשה" ל"תגובה".  
מ "icmp Echo-request" ל "icmp Echo-reply".

#### תיאור הקוד:

לצורך מילוי דרישות המשימה יצרנו קובץ c חדש בשם "snoofer.c".  
קובץ זה למעשה אמור לשלב בין יכולת הסנפת הפקטות הממומשת בקובץ "sniffer.c", לבין יכולת זיוף הפקטות הממומשת בקובץ "spoof.c", ומכאן שמו "snoofer.c".  
לצורך הסנפת הפקטות השתמשנו בספריית "pcap.h", ובפונקציות הממשות הסנפת פקטות שבספרייה זו.

בפונקציה "pcap\_loop" מהספרייה "pcap.h", אנו העברנו מצביע אל הפונקציה "got\_packet" האמורה להסניף את הפקטה מסוג icmp, ולאחר מכן גם לזייף אותה.

לצורך הזיוף השתמשנו ב structs (מבנים) הבאים:  
-struct ipheader המכיל מצביעים לכלל השדות הנמצאים ב header של הפקטה.  
-struct icmpheader המכיל מצביעים לכלל השדות הנמצאים ב icmp-header של הפקטה.  
בעזרת השימוש במבנים אלו, אנו למעשה יצרנו פקטה חדשה, המאותחלת בערכיה של הפקטה אותה הסנפנו, כאשר ההבדל היחיד ביניהן מתבטא בשינוי סטטוס הפקטה מ"בקשה" ל"תגובה", וכן בהחלפת כתובת המקור בכתובת היעד.  
את השדה "check\_sum" אשר בפקטה החדשה, אתחלנו באמצעות שימוש בפונקציית "in\_chksum", ולאחר מכן שלחנו אותה ליעדה באמצעות שימוש ב raw-socket.

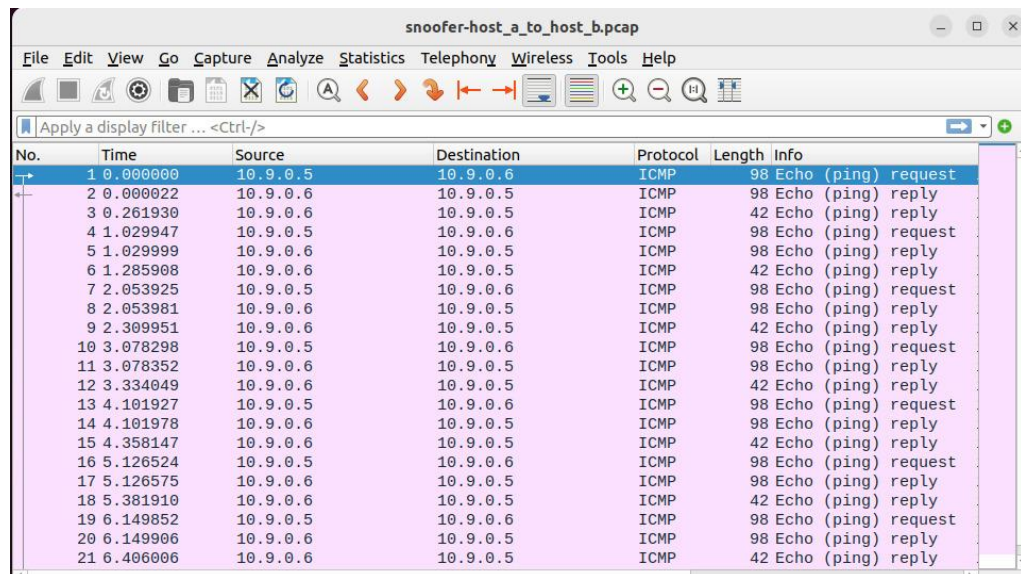
את הקובץ "snoofer.c" אנו הרצנו בקונטיינר של ה"attacker" הנמצא ב docker-compose שבתיקיית "Labsetup" שבמודל, בהתאם להוראות המטלה.  
לצורך בדיקת המטלה יש לשים את הקובץ "snoofer.c" אותו הגשנו, בתיקייה "volumes" הנמצאת בתיקייה "Labsetup" שבמודל.  
בתוך תיקייה זו ישנם 3 קונטיינרים: host A, host B, attacker כאשר לכל אחד מהם ישנה כתובת ip משלו, לצורך דימוי רשת LAN.  
יש לשים לב לפרמטר הראשון הנמצא בפונקציית "pcap\_open\_live" שיהיה זהה לסוג interface הייחודי לרשת LAN שב docker-compose.

אנו הרצנו את הקוד ב3 שלבים:

- 1) שליחת הודעת ping מ host A ל host B.
  - 2) שליחת הודעת ping מ host A ל כתובת ה ip של google - 8.8.8.8.
  - 3) שליחת הודעת ping מ host A לכתובת ip מזויפת.
- כאשר בכל שלב הפעלנו את הקוד הממומש בקובץ "snoofer.c" בקונטיינר של ה attacker.

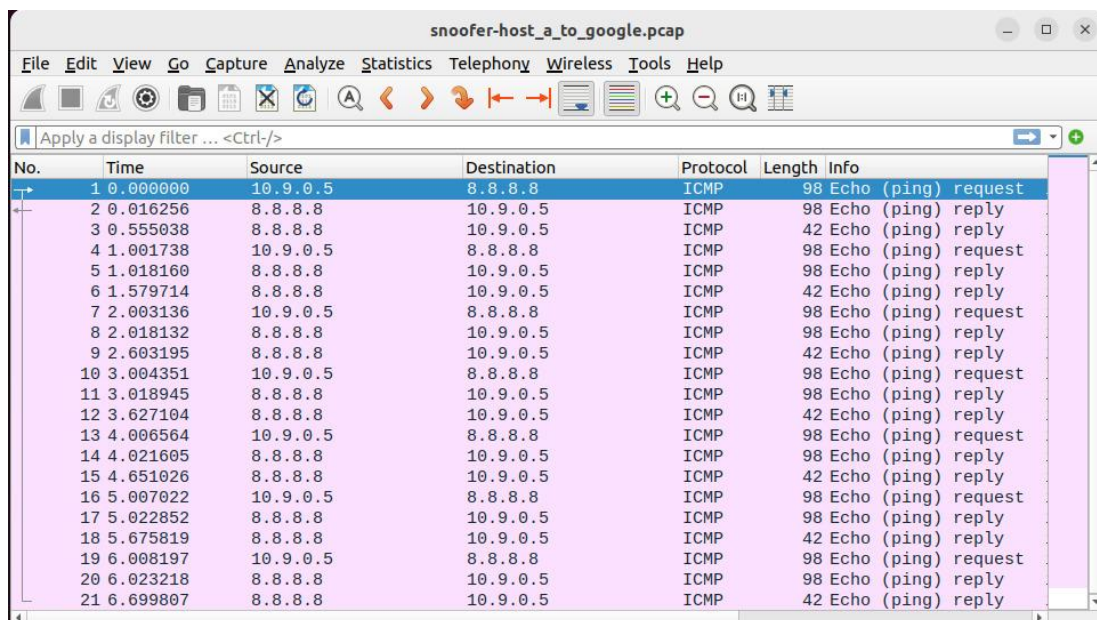
## Task c continue:

תיאור הרצת הקוד על פי Wireshark:



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.9.0.5	10.9.0.6	ICMP	98	Echo (ping) request
2	0.000022	10.9.0.6	10.9.0.5	ICMP	98	Echo (ping) reply
3	0.261930	10.9.0.6	10.9.0.5	ICMP	42	Echo (ping) reply
4	1.029947	10.9.0.5	10.9.0.6	ICMP	98	Echo (ping) request
5	1.029999	10.9.0.6	10.9.0.5	ICMP	98	Echo (ping) reply
6	1.285908	10.9.0.6	10.9.0.5	ICMP	42	Echo (ping) reply
7	2.053925	10.9.0.5	10.9.0.6	ICMP	98	Echo (ping) request
8	2.053981	10.9.0.6	10.9.0.5	ICMP	98	Echo (ping) reply
9	2.309951	10.9.0.6	10.9.0.5	ICMP	42	Echo (ping) reply
10	3.078298	10.9.0.5	10.9.0.6	ICMP	98	Echo (ping) request
11	3.078352	10.9.0.6	10.9.0.5	ICMP	98	Echo (ping) reply
12	3.334049	10.9.0.6	10.9.0.5	ICMP	42	Echo (ping) reply
13	4.101927	10.9.0.5	10.9.0.6	ICMP	98	Echo (ping) request
14	4.101978	10.9.0.6	10.9.0.5	ICMP	98	Echo (ping) reply
15	4.358147	10.9.0.6	10.9.0.5	ICMP	42	Echo (ping) reply
16	5.126524	10.9.0.5	10.9.0.6	ICMP	98	Echo (ping) request
17	5.126575	10.9.0.6	10.9.0.5	ICMP	98	Echo (ping) reply
18	5.381910	10.9.0.6	10.9.0.5	ICMP	42	Echo (ping) reply
19	6.149852	10.9.0.5	10.9.0.6	ICMP	98	Echo (ping) request
20	6.149906	10.9.0.6	10.9.0.5	ICMP	98	Echo (ping) reply
21	6.406006	10.9.0.6	10.9.0.5	ICMP	42	Echo (ping) reply

בתמונה זו ניתן לראות כיצד העברנו הודעת ping מ host A שכתובת ה ip שלו היא 10.9.0.5, אל עבר host B אשר כתובת ה ip שלו היא 10.9.0.6 וקיבלנו בחזרה פעמיים הודעה מסוג "icmp Echo-reply".  
הסיבה לכפילות זו, היא משום שה "attacker" למעשה התחזה ל host B, והעביר הודעת "icmp Echo-reply" בשמו, כך ש host A קיבל בחזרה פעמיים הודעת "icmp Echo-reply".



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.9.0.5	8.8.8.8	ICMP	98	Echo (ping) request
2	0.016256	8.8.8.8	10.9.0.5	ICMP	98	Echo (ping) reply
3	0.555038	8.8.8.8	10.9.0.5	ICMP	42	Echo (ping) reply
4	1.001738	10.9.0.5	8.8.8.8	ICMP	98	Echo (ping) request
5	1.018160	8.8.8.8	10.9.0.5	ICMP	98	Echo (ping) reply
6	1.579714	8.8.8.8	10.9.0.5	ICMP	42	Echo (ping) reply
7	2.003136	10.9.0.5	8.8.8.8	ICMP	98	Echo (ping) request
8	2.018132	8.8.8.8	10.9.0.5	ICMP	98	Echo (ping) reply
9	2.603195	8.8.8.8	10.9.0.5	ICMP	42	Echo (ping) reply
10	3.004351	10.9.0.5	8.8.8.8	ICMP	98	Echo (ping) request
11	3.018945	8.8.8.8	10.9.0.5	ICMP	98	Echo (ping) reply
12	3.627104	8.8.8.8	10.9.0.5	ICMP	42	Echo (ping) reply
13	4.006564	10.9.0.5	8.8.8.8	ICMP	98	Echo (ping) request
14	4.021605	8.8.8.8	10.9.0.5	ICMP	98	Echo (ping) reply
15	4.651026	8.8.8.8	10.9.0.5	ICMP	42	Echo (ping) reply
16	5.007022	10.9.0.5	8.8.8.8	ICMP	98	Echo (ping) request
17	5.022852	8.8.8.8	10.9.0.5	ICMP	98	Echo (ping) reply
18	5.675819	8.8.8.8	10.9.0.5	ICMP	42	Echo (ping) reply
19	6.008197	10.9.0.5	8.8.8.8	ICMP	98	Echo (ping) request
20	6.023218	8.8.8.8	10.9.0.5	ICMP	98	Echo (ping) reply
21	6.699807	8.8.8.8	10.9.0.5	ICMP	42	Echo (ping) reply

גם בתמונה זו ניתן לראות כיצד host A שלח הודעה מסוג "icmp Echo-request" כלפי כתובת ה ip של google, וקיבל בחזרה פעמיים הודעת "icmp Echo-reply".  
הסיבה לכך, היא ההתחזות של ה "attacker" ל google.



## Task c continue:

snoofer-host_a_to_fake.pcap						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
Apply a display filter ... <Ctrl-/>						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.9.0.5	1.2.3.4	ICMP	98	Echo (ping) request id=0x
2	0.099304	1.2.3.4	10.9.0.5	ICMP	42	Echo (ping) reply id=0x
3	1.027276	10.9.0.5	1.2.3.4	ICMP	98	Echo (ping) request id=0x
4	1.123551	1.2.3.4	10.9.0.5	ICMP	42	Echo (ping) reply id=0x
5	2.051515	10.9.0.5	1.2.3.4	ICMP	98	Echo (ping) request id=0x
6	2.147101	1.2.3.4	10.9.0.5	ICMP	42	Echo (ping) reply id=0x
7	3.076308	10.9.0.5	1.2.3.4	ICMP	98	Echo (ping) request id=0x
8	3.171108	1.2.3.4	10.9.0.5	ICMP	42	Echo (ping) reply id=0x
9	4.099009	10.9.0.5	1.2.3.4	ICMP	98	Echo (ping) request id=0x
10	4.195248	1.2.3.4	10.9.0.5	ICMP	42	Echo (ping) reply id=0x
11	5.123006	10.9.0.5	1.2.3.4	ICMP	98	Echo (ping) request id=0x
12	5.219652	1.2.3.4	10.9.0.5	ICMP	42	Echo (ping) reply id=0x
13	6.146992	10.9.0.5	1.2.3.4	ICMP	98	Echo (ping) request id=0x
14	6.243682	1.2.3.4	10.9.0.5	ICMP	42	Echo (ping) reply id=0x
15	7.171075	10.9.0.5	1.2.3.4	ICMP	98	Echo (ping) request id=0x
16	7.266993	1.2.3.4	10.9.0.5	ICMP	42	Echo (ping) reply id=0x
17	8.195017	10.9.0.5	1.2.3.4	ICMP	98	Echo (ping) request id=0x
18	8.291679	1.2.3.4	10.9.0.5	ICMP	42	Echo (ping) reply id=0x
19	9.219023	10.9.0.5	1.2.3.4	ICMP	98	Echo (ping) request id=0x
20	9.315342	1.2.3.4	10.9.0.5	ICMP	42	Echo (ping) reply id=0x
21	10.243994	10.9.0.5	1.2.3.4	ICMP	98	Echo (ping) request id=0x
22	10.339788	1.2.3.4	10.9.0.5	ICMP	42	Echo (ping) reply id=0x
Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)						
Ethernet II, Src: 02:42:0a:09:00:05 (02:42:0a:09:00:05), Dst: 02:42:44:6a:79:61 (02:42:44:6a:79:61)						
0000	02 42 44 6a 79 61 02 42	0a 09 00 05 08 00 45 00	BDjya B .....E			
0010	00 54 f8 83 40 00 40 01	34 12 0a 09 00 05 01 02	T..@. 4.....			
0020	03 04 08 00 d1 1a 00 09	00 01 0f 4c cc 63 00 00	.....L.c...			
0030	00 00 88 58 04 00 00 00	00 00 10 11 12 13 14 15	...X.....			
0040	16 17 18 19 1a 1b 1c 1d	1e 1f 20 21 22 23 24 25	.....!"#\$%			
0050	26 27 28 29 2a 2b 2c 2d	2e 2f 30 31 32 33 34 35	&'()*+,-./012345			
0060	36 37		67			

בתמונה זו ניתן להבחין ביכולות זיוף הפקטות של ה "attacker", כאשר הוא למעשה יוצר פקטה מסוג "icmp Echo-reply" עם כתובת ip מזויפת ושולח אותה בחזרה ל host A, לאחר ששלח הודעת פינג אל עבר הכתובת המזויפת.