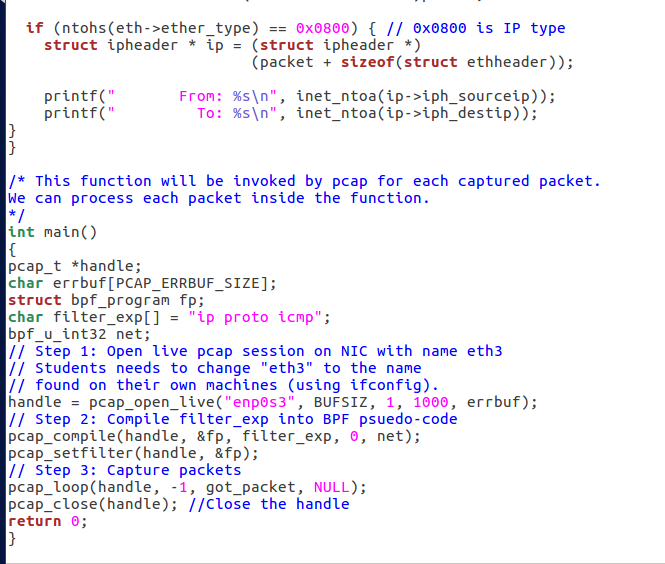
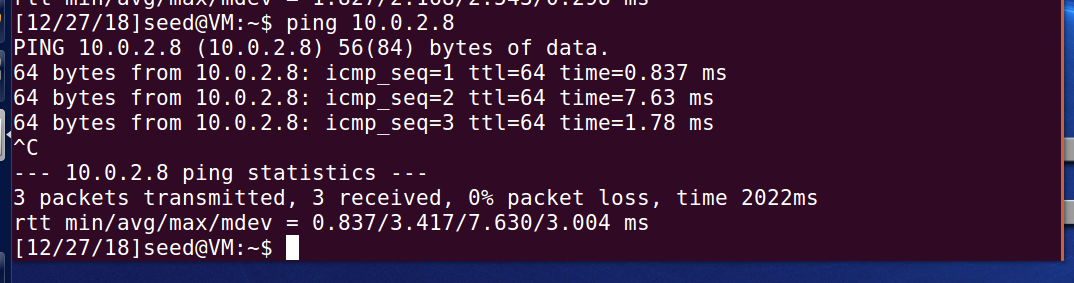
מטלה 6 :

**שאלה 2.1 A –**

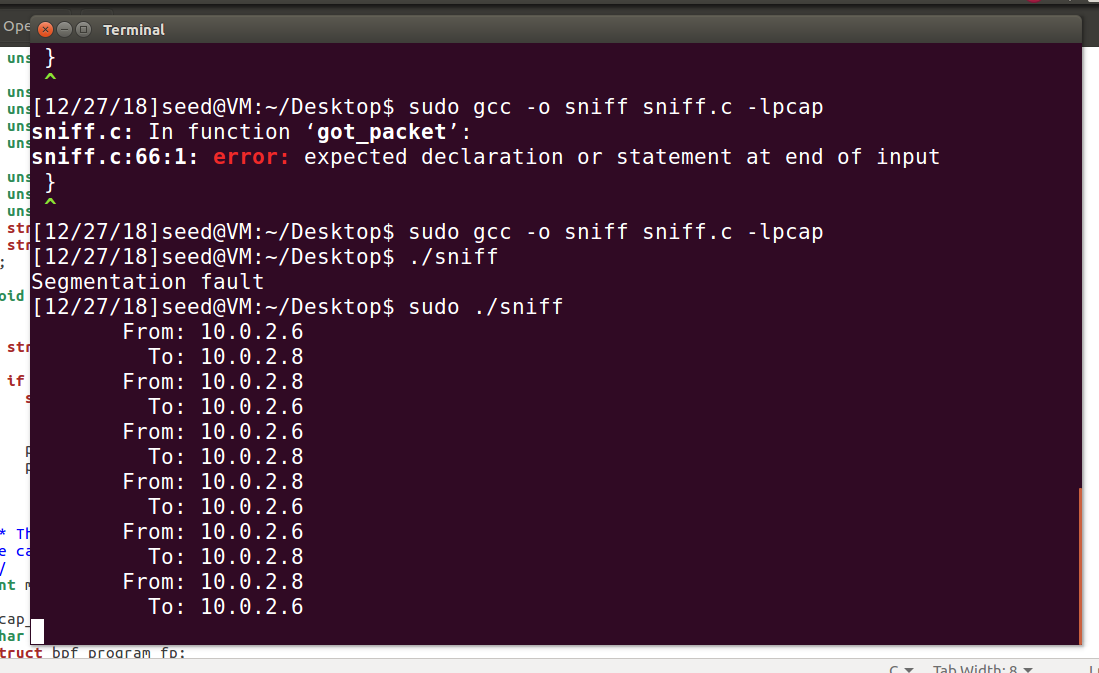
הקוד שבנינו את הסניפר על ידו, וגם מדפיסים למי הפקטה מיועדת ומי השולח



מכונה 1 שולחת פינג למכונה השנייה (10.0.2.8 המכונה השניה)



הפקטות שנתפסו במכונה 2 על ידי הסניפר שבנינו



שאלה 1 :

שלב ראשון, להגדיר עם איזה ממשק רוצים לעבוד ודרכו ללכוד .

שלב שני, לפתוח session וליצור את הסניפר על ידי הפונקציה:

pcap\_open\_live ובפונקציה הזאת אנחנו נקבע : על איזה מכשיר אנחנו רוצים להפעיל את הסניפר ונתונים נוספים כמו הגודל המרבי של בתים שאפשר לתפוס, זמנים, תפיסת שגיאות ועוד.

שלב שלישי, אם אנחנו רוצים לתפוס פקטות מסויימות על ידי איזשהו פילטר שהגדרנו אז צריך להשתמש בפונקציה :

pcap\_compile שפונקציה זו קובעת את החוקים של ההסנפה ומקבלת את המחרוזת של הפילטר הרצוי ו'מקמפלת' אותם לשפה שהפורמט של הספריה PCAP תוכל לקרוא .

ואז מפעילים את הפונקציה pcap\_setfilter כי לאחר הקימפול אפשר להשתמש בנתונים המקופלים ולכן פונקציה זו מספקת לנו את האפשרות לעבוד איתם אחרי הקמפול.

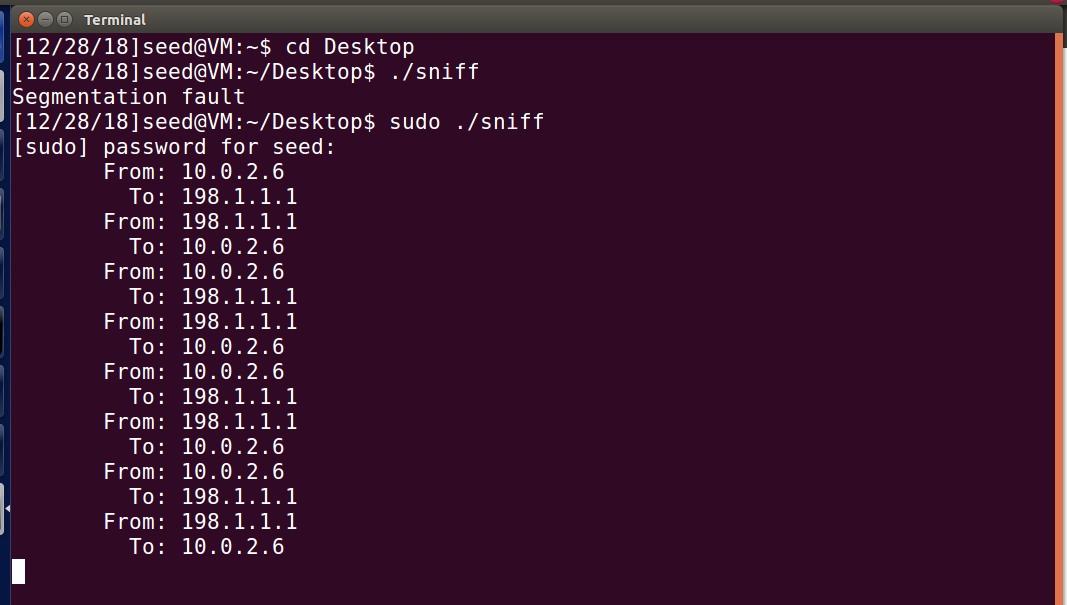
שלב רביעי, לאחר כל התהליך נרצה שהסניפר יתחיל לעבוד - הפונקציה : pcap\_loop כלומר יכנס ללואה בה הוא תופס פקטות עד שמגיע לגודל שהגדרנו לו מראש ובכל פעם שהוא מקבל פקטה הוא קורא לפונקציה got\_packet שבה נגדיר מה אנחנו רוצים לעשות עם הפקטה הזאת : להדפיס את הIPשולח והIPמקבל, לנתח את החבילה ועוד.

שלב אחרון, לאחר שסיימנו לקבל את כל המידע שרצינו הפונקציה pcap\_close סוגרת את הsession.

שאלה 2:

התוכנית של הסניפר שבנינו צריכה לגשת למידע של מערכת ההפעלה (על ידי תהליכים מסויימים) כדי לקבל מידע אודות הרשת בה אנחנו עובדים ולגשת אליו ולשלוח אליה חבילות ולכן פעולות כמו אלה לא יכולות להתבצע ללא הרשאת שורש (מנהל) .

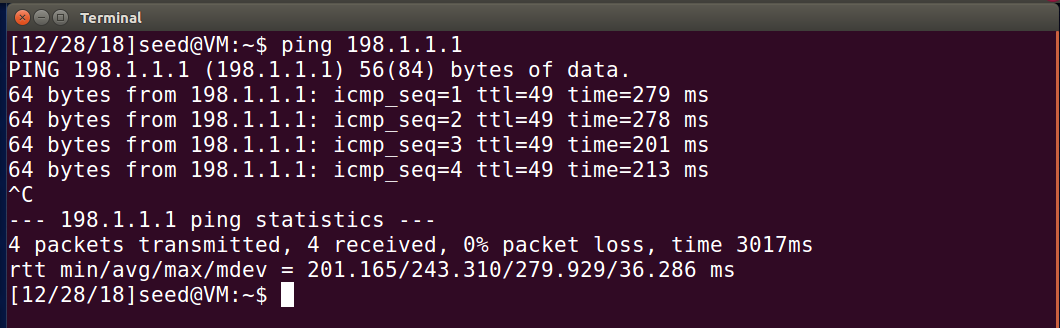
התוכנית לא תבצע את הנדרש ברגע שנפעיל אותה מהסיבות הנל. ראה תמונה:



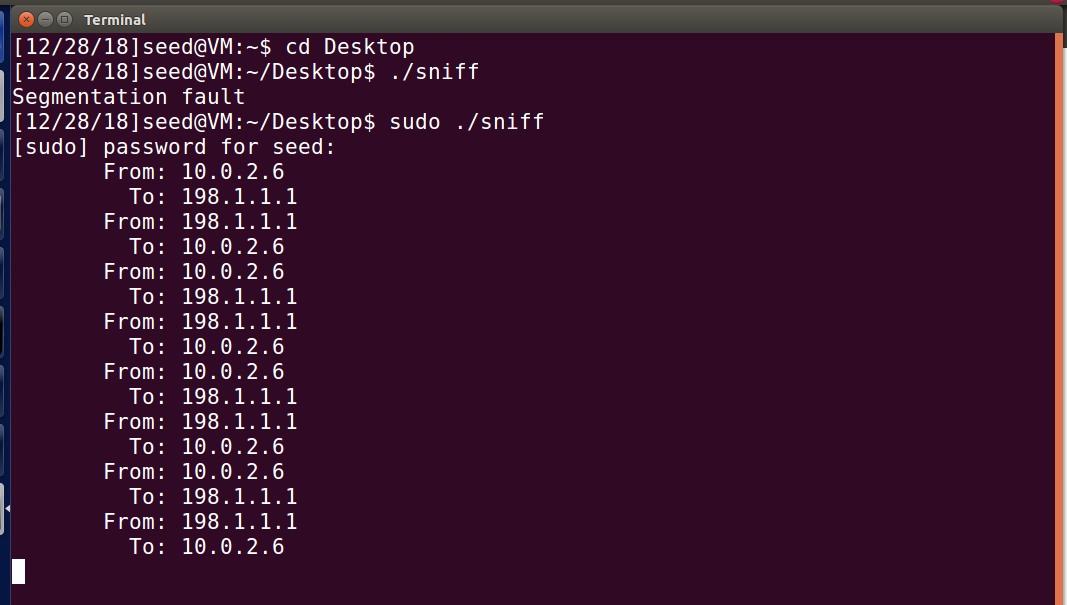
שאלה 3 :

מצב מופקר זה מצב בו פקטות שנשלחות ממחשב מסויים נקלטות ומתקבלות בסניפר של מחשב אחר אפילו שהן לא מיועדות למחשב הזה ומצב לא מופקר מקבל רק פקטות שמיועדות למחשב שלו, לכתובת שלו :

מצב מופקר – שליחת פינג ממכונה 1 לכתובת שרירותית

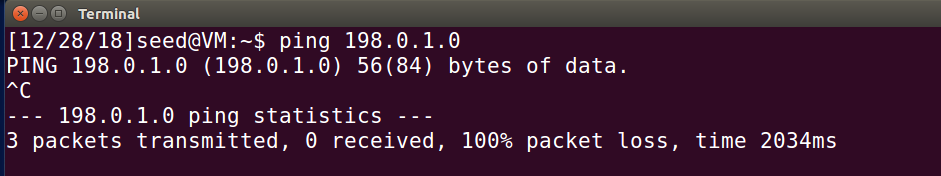


קבלת הפקטות במכונה 2 על ידי הsniffer שנמצא כעת במצב מופקר



מצב לא מופקר –

שליחת פינג ממכונה 1 לכתובת שרירותית(ניתן לראות שהפקטות אבדו כלומר לא התקבלו )



הפקטות במכונה 2 לא התקבלו על ידי הסניפר שנמצא כעת במצב לא מופקר

**שאלה 2.1 B -**

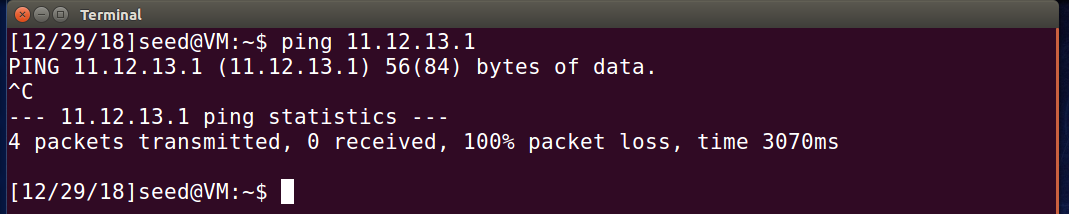
נקודה ראשונה: לכידת פקטות ICMP בין שתי מחשבים ספציפיים.

הקוד (השינויים מסומנים)

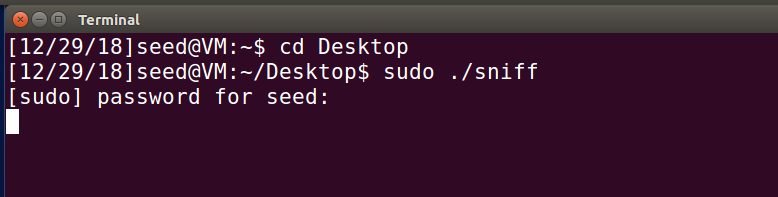


ניתן לראות שנשלח פינג לכתובת שרירותית והוא לא הוסנף

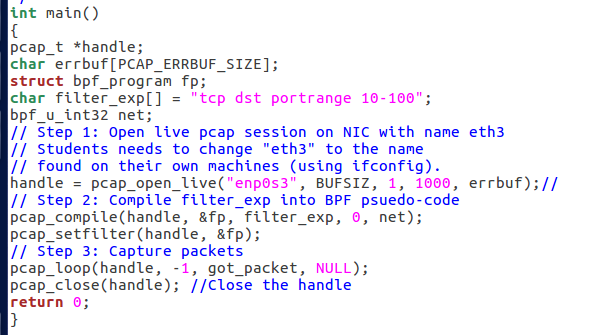
מכונה 1



מכונה 2 (עם סניפר)

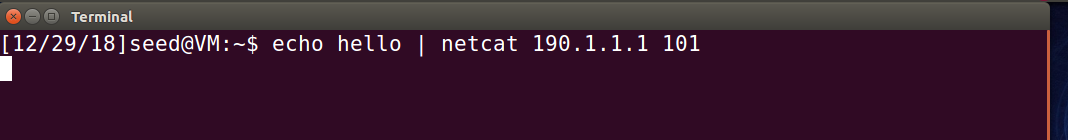


נקודה שניה : לכידת פקטות TCP עם מספר פורט יעד בטווח בין 10 100.

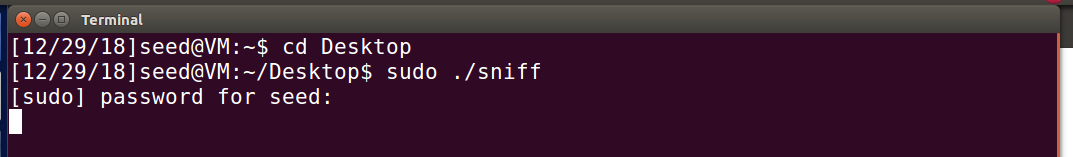
הקוד (השינויים מסומנים)

ניתן לראות שנשלחה הודעת tcp לכתובת שרירותית ולפורט שלא נמצא בטווח בין 10 ל100 ולכן הודעה זו לא תתקבל

מכונה 1



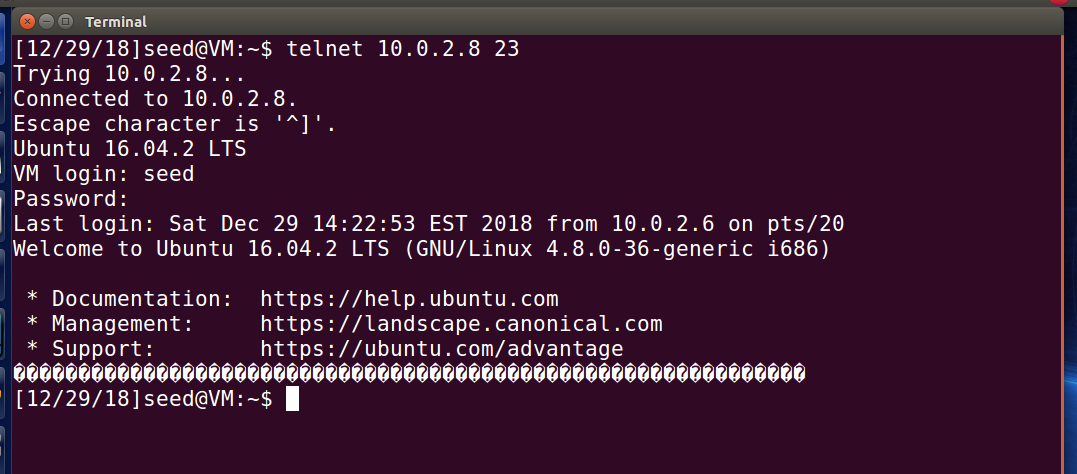
מכונה 2 (עם הסניפר)- ניתן לראות שהפקטה לא נתפסה



**שאלה 2.1 C –**

מצורף קובץ קוד (ראה sniff.c)

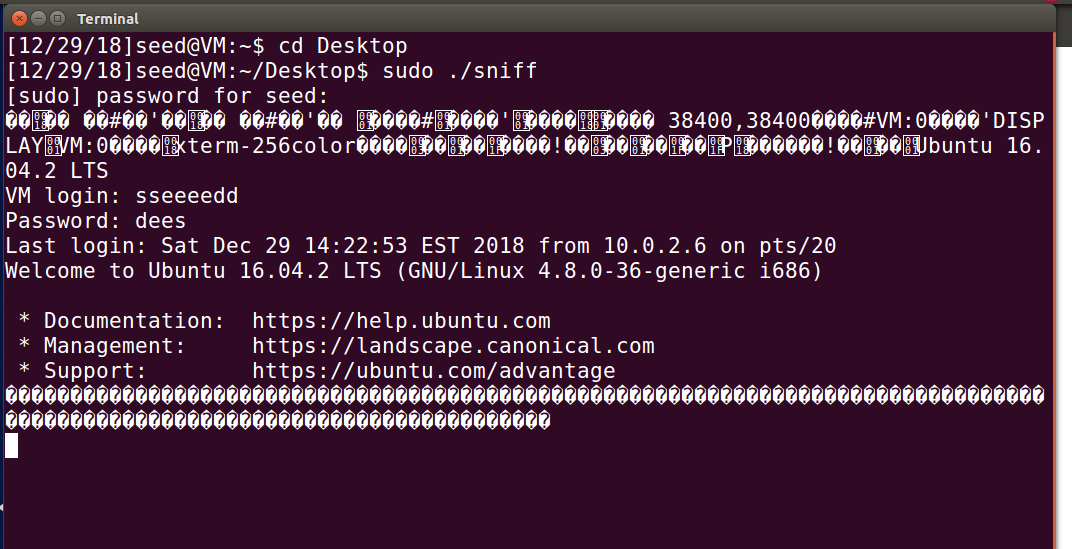
התחברות של מכונה מספר 1 – הקשת פרטי התחברות (סיסמא : dees)



פרטי ההתחברות

לאיזה כתובת 'שרת' להתחבר דרך הפורט המסוים של הפרוטוקול

מכונה 2 מסניפה את הפרטים של מכונה 1



הסיסמא

מכונה מספר 1

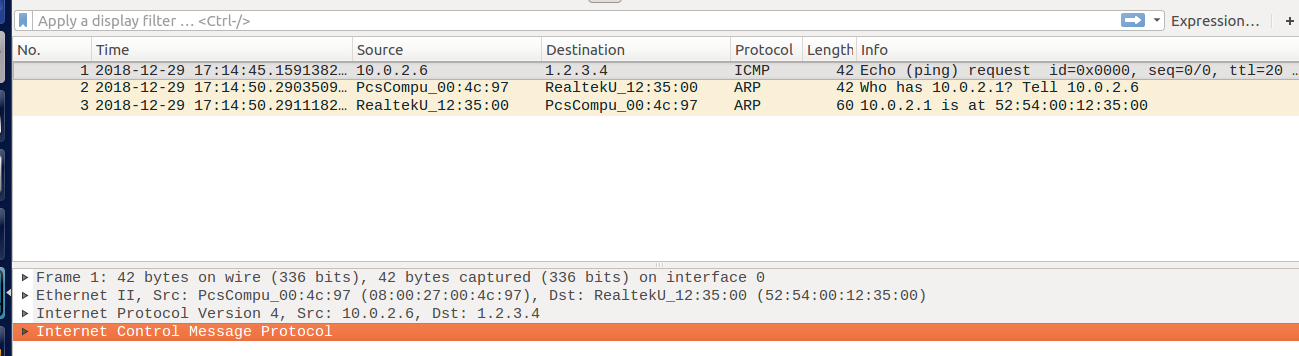
ניתן לראות שמכונה מספר 2 קולטת את כל פרטי ההתחברות ומציגה את הסיסמא של מכונה 1.

החלק של המידע נמצא בסוף הפקטה כלומר בחלק האחרון בה נוכל לראות את כל המידע הנוסף שהפקטה מכילה.

**שאלה 2.2A –**

מצורף קובץ קוד (ראה spoof1.c )

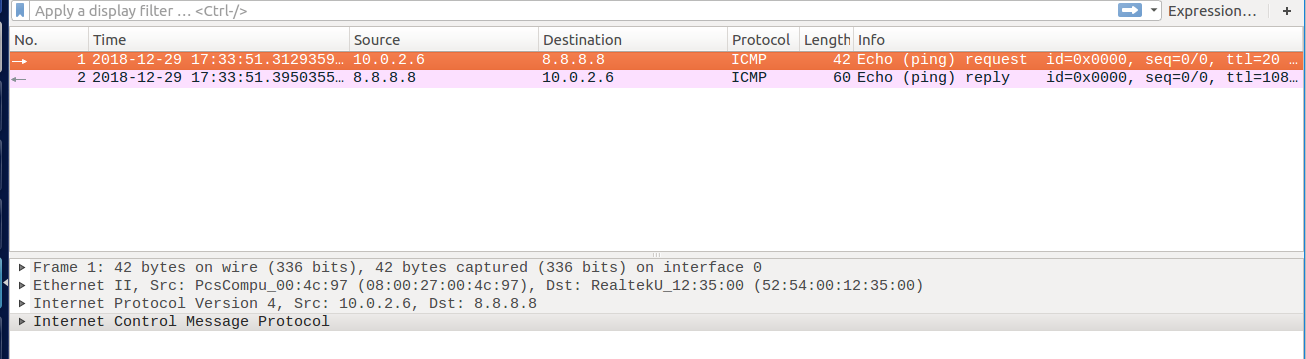
בתמונה רואים שהפקטה שנבנתה, ויוצאת ליעד שרירותי, נשלחה ונתפסה .



**שאלה 2.2 B –**

מצורף קובץ קוד (ראה spoof2.c )

בתמונה רואים שנשלחה פקטה ICMP לכתובת IP (קיימת) מסויימת ומקבלים echo בחזרה מהכתובת הזו.



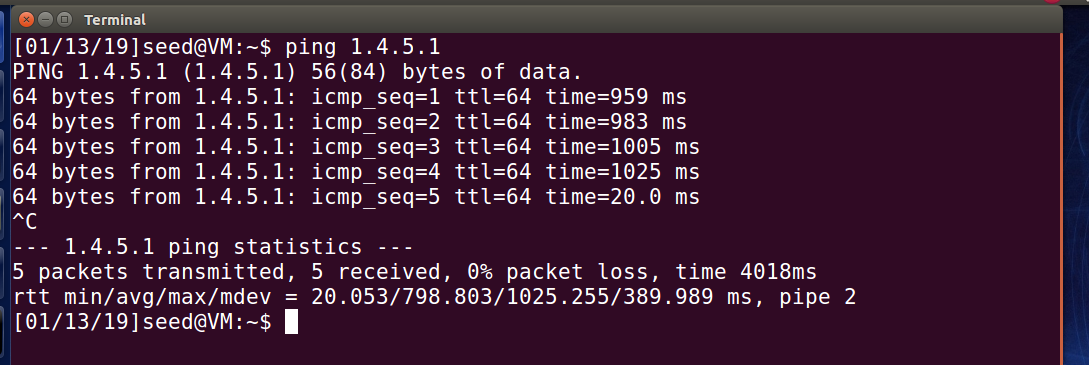
קיבלנו reply

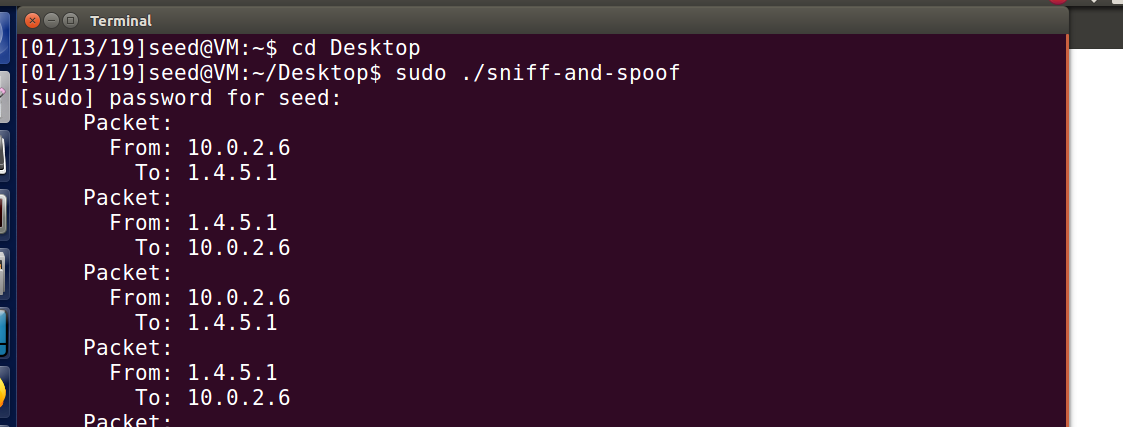
שאלה 4 : כן, ניתן להגדיר את אורך הIP header לערכים שרירותיים.

שאלה 5: לא, המחשב ממלא את זה באופן אוטומטי.

שאלה 6: כי ללא הרשאת שורש ייתכן והיה ניתן לבנות פקטות מתואמות אישית שעלולות לפגוע במחשבים. ולכן החליטו לתת לכך הרשאת שורש.

**שאלה 2.3 –** מצורף קוד (ראה sniff-and-spoof )

בתמונה זו ניתן לראות שמכונה מספר 1 מנסה לשלוח ping לכתובת שרירותית 

ניתן לראות שהפקטות מתקבלות, כלומר יש מענה לping שנשלח וזה על ידי מכונה מספר 2 שמסניפה את פקטות הICMP שמגיעות ממכונה מספר 1 ומרכיבה פקטה של ICMP response מזוייפת למכונה מספר 1 עם התשובה שמכונה מספר 1 מצפה לקבל 

נשלחה פקטה מזוייפת שמכונה מספר 2 יצרה

פקטת ICMP של מכונה 1 נתפסה