פרויקט סיום- תקשורת ומחשוב

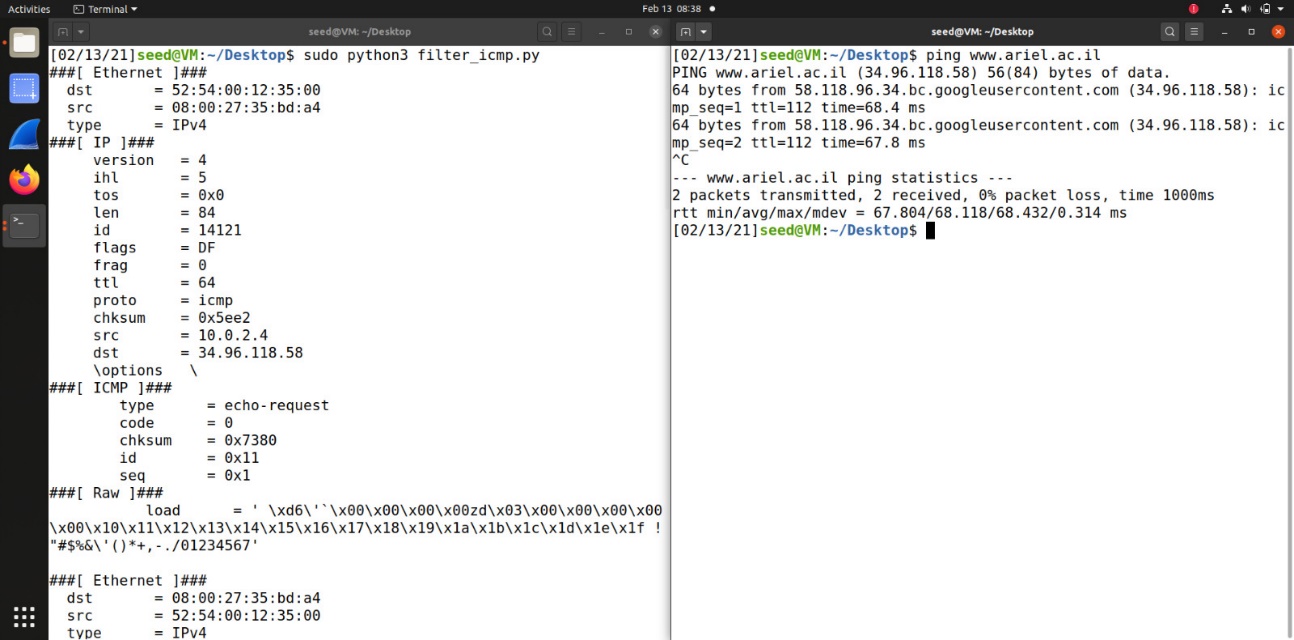
מגישים: עמית חג'ג'- 205837727

כפיר אטינגר- 311611677

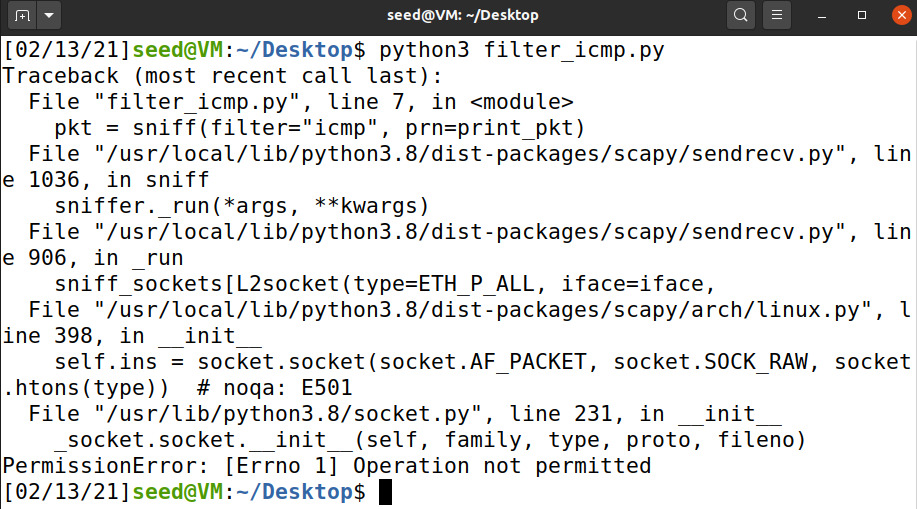
חלק 1- Python

משימה A1.1:

במשימה זו אנחנו נדרשים להפעיל את התכנית עם root privilege. ניתן לראות בתמונה שהתכנית עובדת כראוי ופאקטות נתפסות. כמו כן, ניתן לראות את שכבות הפאקטה. (IP, ICMP, ETH, RAW data)

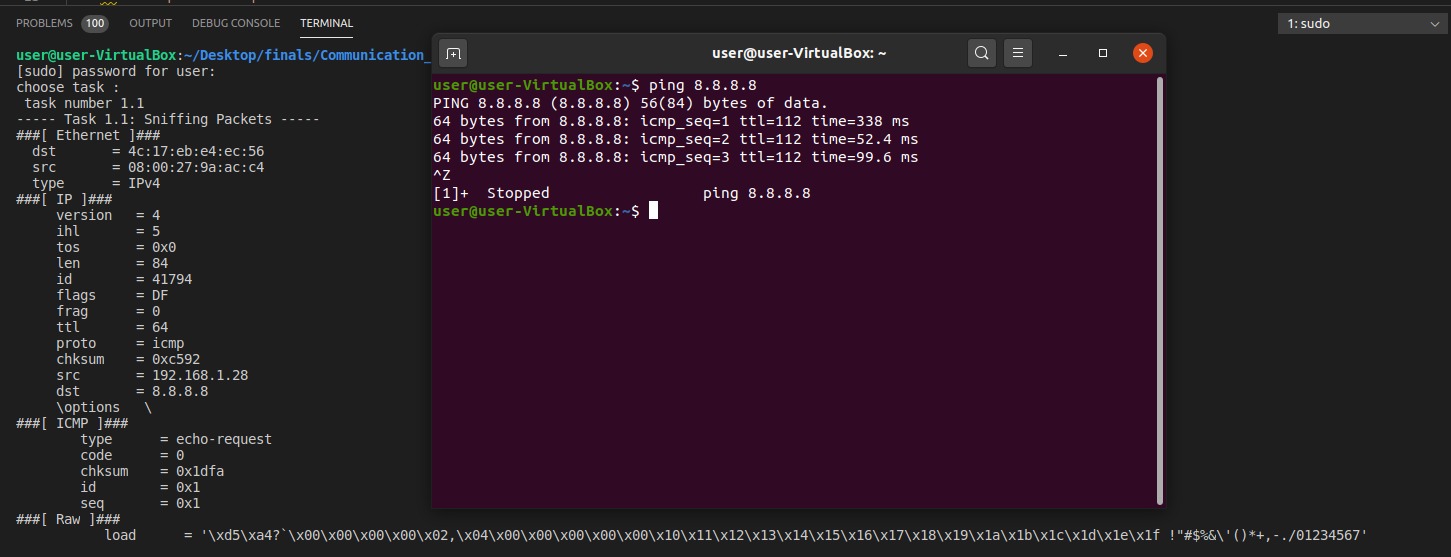


כעת אנחנו מריצים ללא root privilege. (הסבר בתמונה)

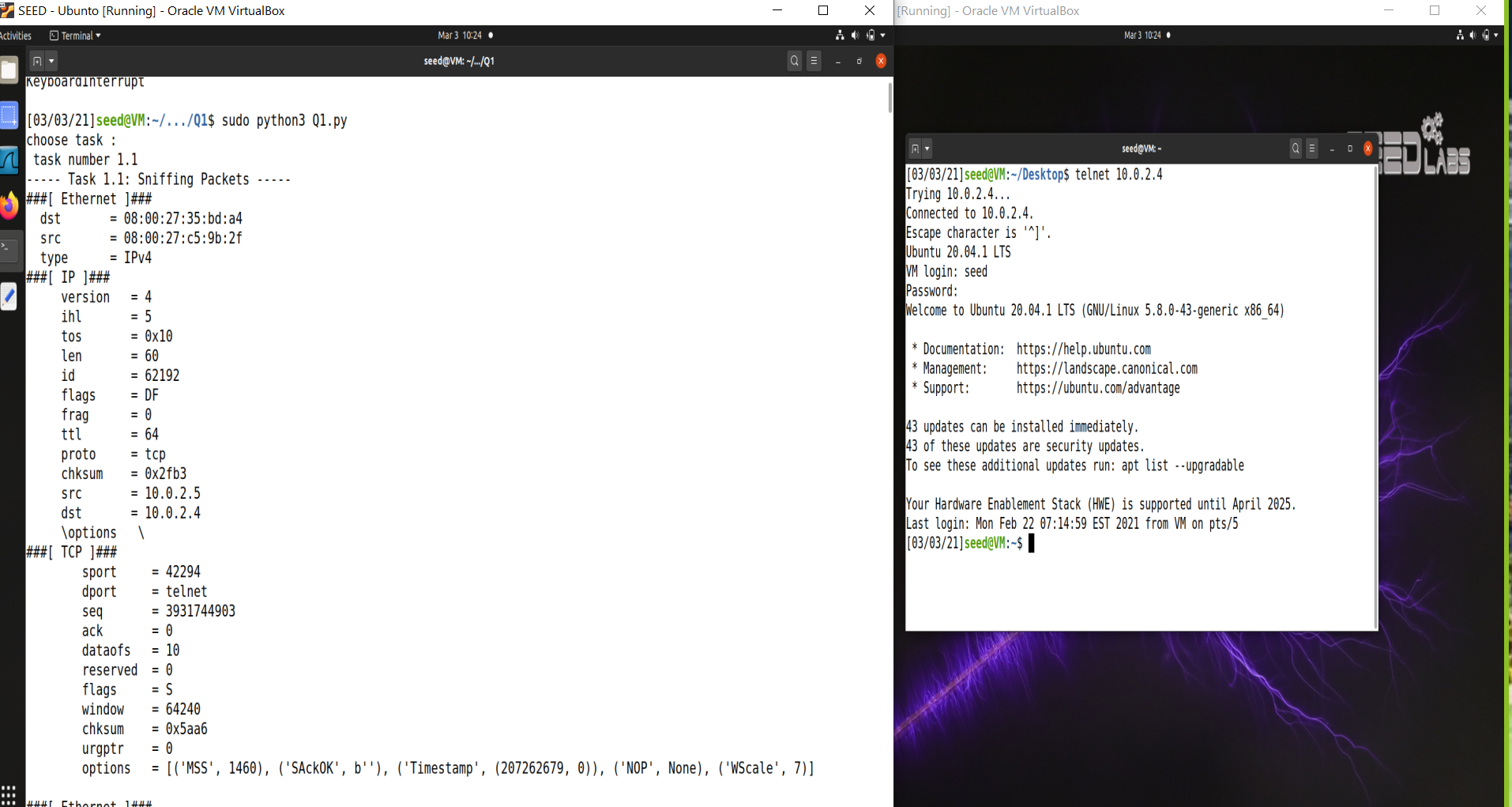


בגלל שהרצנו ללא הרשאות קיבלנו permission error.

משימה B1.1:

כאן אנחנו נדרשים להריץ את התכנית אבל בכל פעם להגדיר פילטר אחר.  
פילטר 1- ICMP.  


ICMP

פילטר 2- IP & dest port 23.  


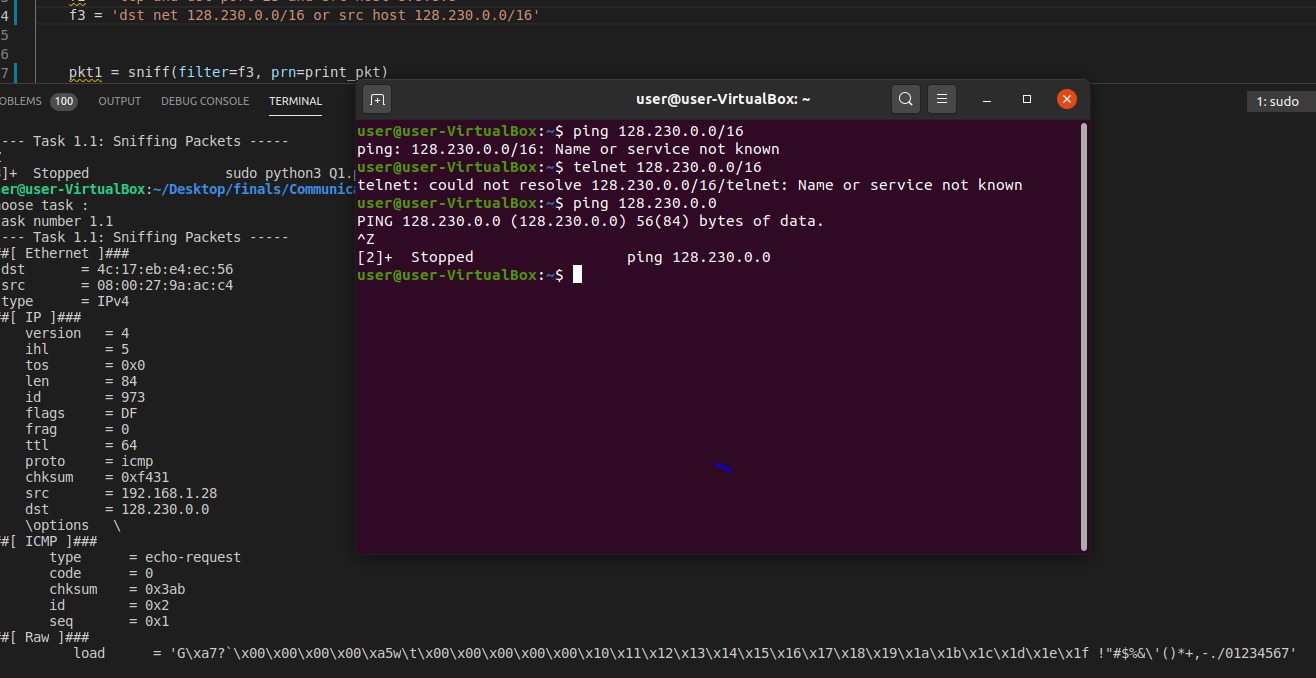
10.0.2.4

10.0.2.5

Telnet listen over port 23

Our particular src IP

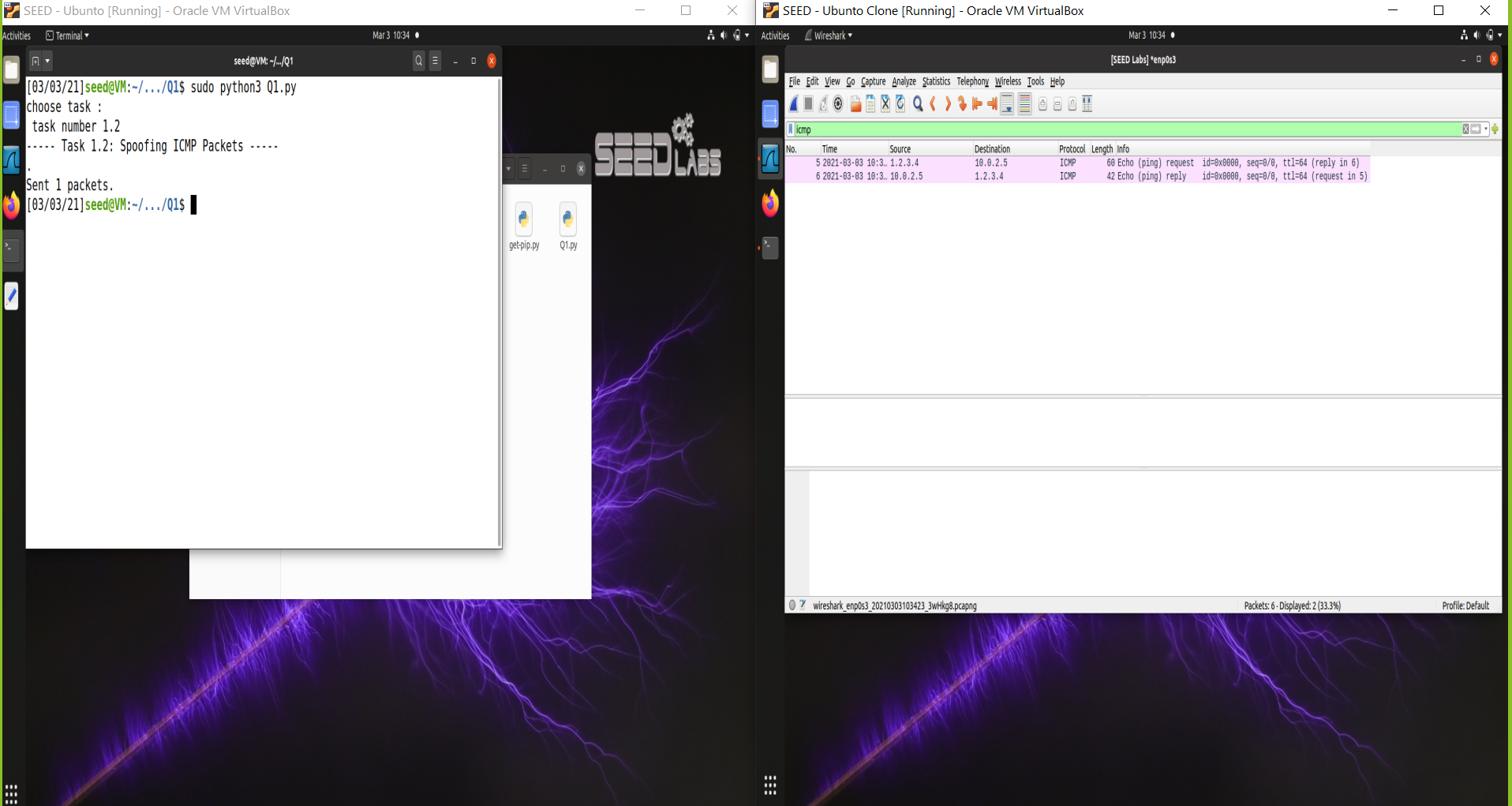
TCP

פילטר 3- מ-subnet ספציפי.  


Icmp packet from our subnet.

משימה 1.2:

במשימה הזאת אנחנו נדרשים לעשות ספוף לחבילה שנשלחת אל מכונה שנייה על הרשת מכתובת IP כלשהי. (שלחנו ממכונה אחת(שמאלית) למכונה השנייה(ימנית) מכתבות פיקטיבית)



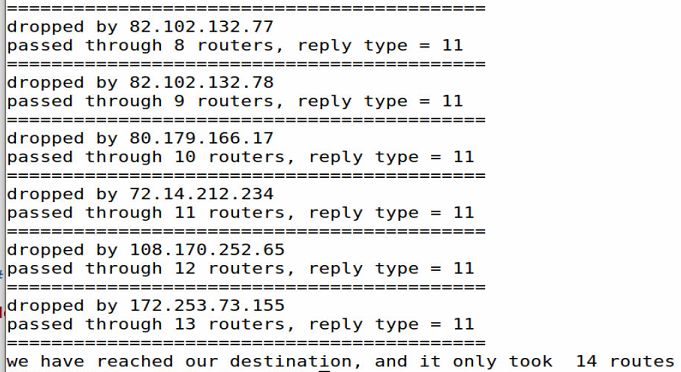
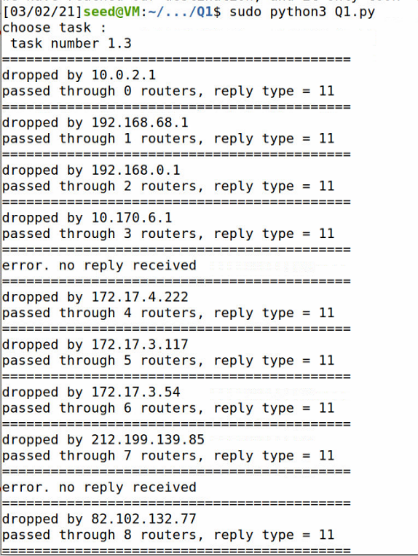
קיבלנו בקשה מ-1.2.3.4

10.0.2.4

10.0.2.5

משימה 1.3:

במשימה זאת אנחנו התבקשנו לממש סוג של traceroute. התוכנית צריכה לספור את מספר הראוטרים מכתובת המקור עד לכתובת היעד. ביצענו זאת ע"י שליחת פאקטה מסוג ICMP (echo-request) עם ttl=1. בכל איטרציה הגדלנו את ttl באחד עד שהגענו ליעד. כל ראוטר בדרך מקטין את ttl באחד וכאשר ראוטר מקבל פאקטה, אם ttl=1 הוא מחזיר הודעת ICMP מסוג 11(time exceeded) במקום להמשיך לראוטר הבא. ואם הראוטר הוא כתובת היעד אז הוא יחזיר למקור ICMP מסוג 0 (echo reply). בתמונה ניתן לראות שלאחר 14 ראוטרים הגענו אל היעד. אפשר לראות שלא קיבלנו תשובה מכל ראוטר בדרך ולכן הוספנו בקוד timeout במידה ולא הגיעה תשובה בזמן.



1

2

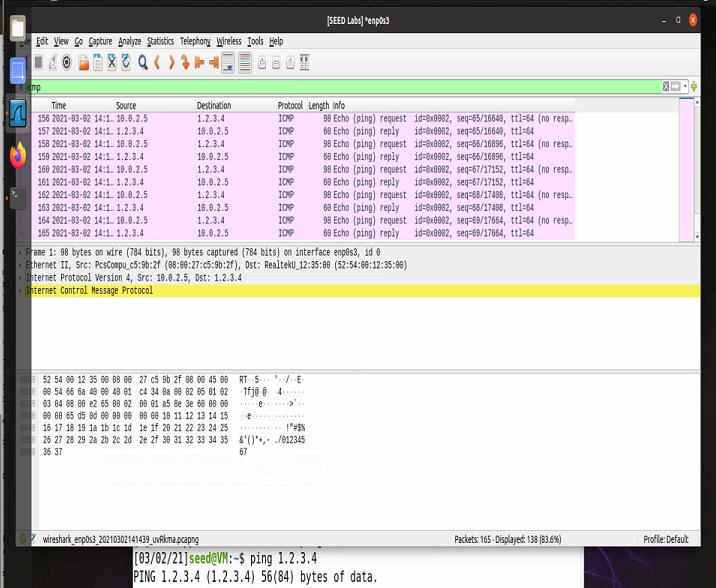
הפקטה הגיעה אל היעד אחרי 14 ראוטרים

הראוטר הראשון במסלול של הפקטה

משימה 1.4:

במשימה זאת אנחנו מבצעים סניף ואז ספוף בשלושה מקרים שונים.

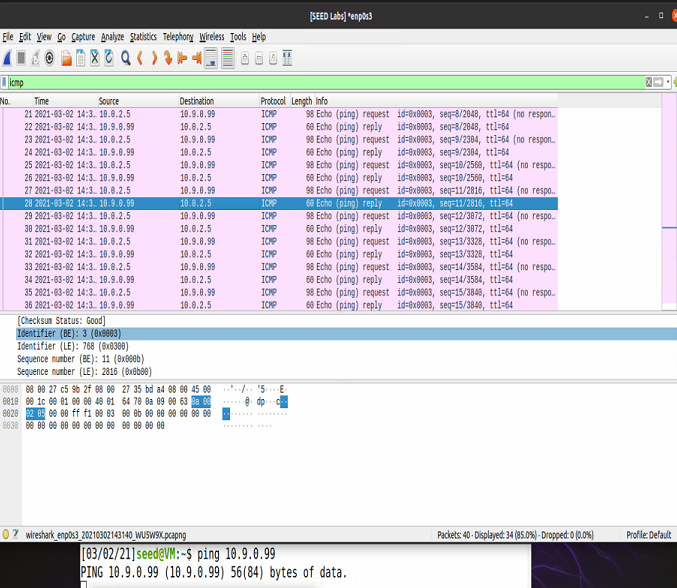
מקרה 1- כתובת לא אמיתית ברשת.



הודעות reply שאנחנו יצרנו

בקשות לכתובת לא אמיתית

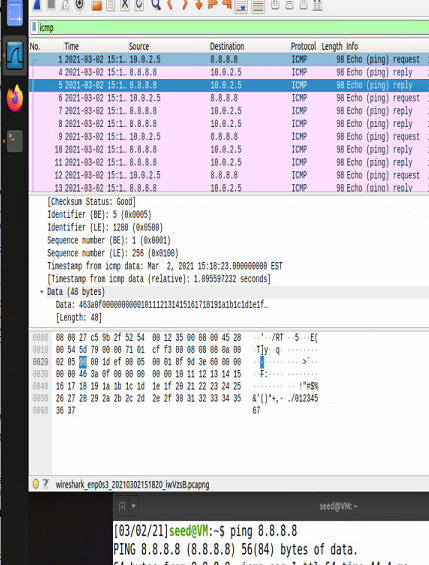
מקרה 2- כתובת לא אמיתית על ה-LAN.



הודעת reply שאנחנו שלחנו.

בקשות לכתובת לא קיימת על הLAN

מקרה 3- כתובת אמיתית ברשת.



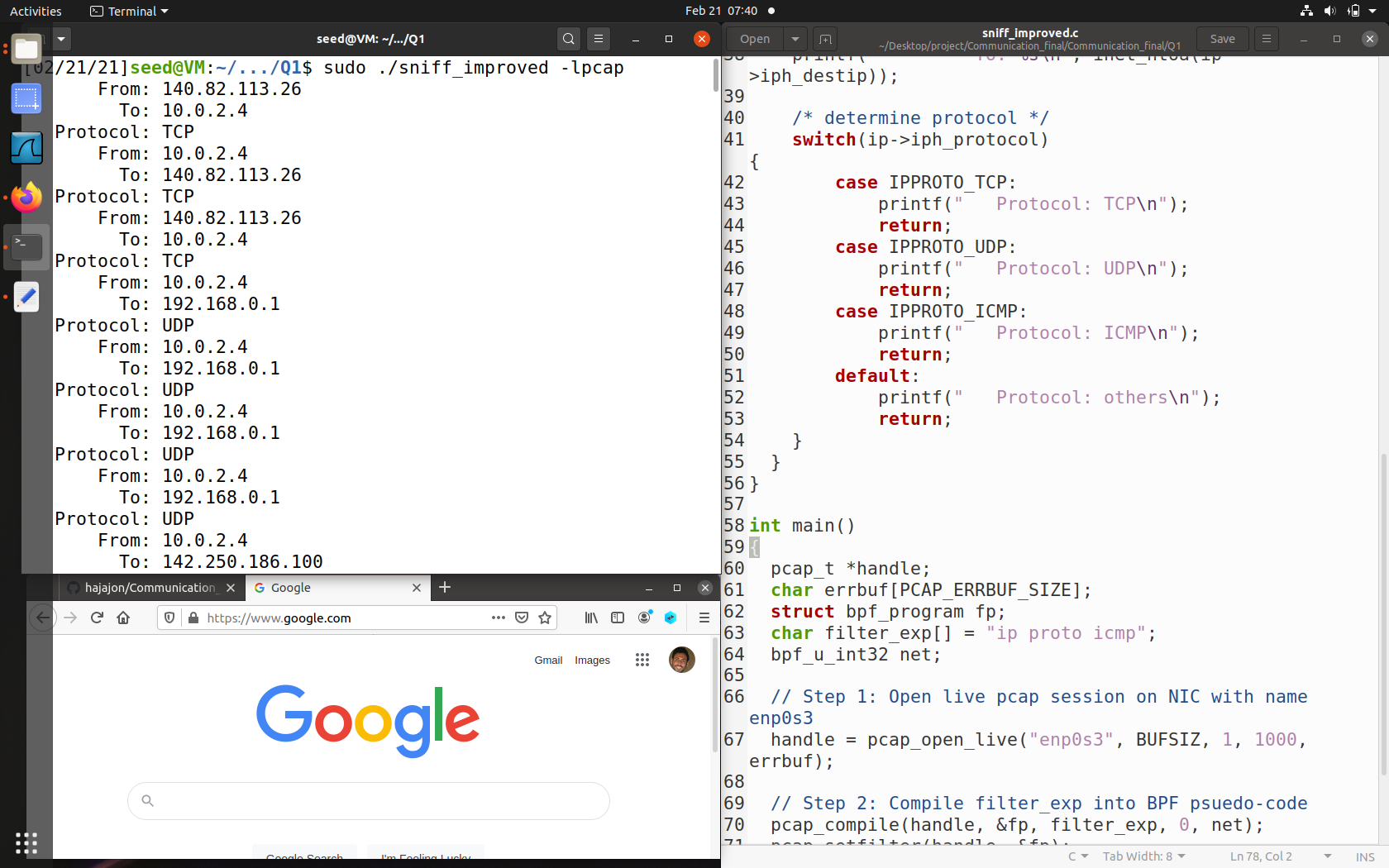
מכיוון שמדובר בכתובת אמיתית, קיבלנו 2 הודעות reply. אחת שאנחנו שלחנו ואחת מהכתובת עצמה.

בקשה מכתובת אמיתית באינטרנט

חלק 2- C

משימה A2.1:

בתמונה למטה ניתן לראות את התכנית שתופסת פאקטות מכל סוג.



שאלה 2.1.1:

בקוד המצורף, התהליך מתחיל בmain כאשר אנחנו פותחים סשן pcap חדש עבור הממשק שבחרנו.(שדה ריק, יקלוט מכל הממשקים שעל כרטיס הרשת) לאחר מכן אנחנו הופכים את המחרוזת filter\_exp[] לקובץ שיהווה פילטר עבור  pcap בשורה שאחריה.(מכניסים קובץ זה אל fp) בשלב האחרון אנחנו עוברים בלולאה על הפאקטות שעוברות בממשק שהגדרנו ואז תופסים אותם לפי הפילטר שהגדרנו גם כן. כל פאקטה שנתפסה הולכות לפונקציה ייעודית לטיפול. אותה פונקציה במקרה שלנו היא הפונקציה בתחילת הקוד שבעצם מדפיסה למסך, איזה סוג פאקטה זאת, כתובת מקור וכתובת יעד.

שאלה 2.1.2:

אנחנו מוכרחים root privileges מכיוון שאנחנו רוצים לשנות promiscuous mode של כרטיס הרשת. הרעיון מאחורי זה הוא שלא כל משתמש יוכל לגעת בהגדרות אלו ולהפוך את המחשב שלנו לפגיע. לגבי השאלה השנייה, השלב בו אנחנו נתקעים הוא של ההרצה עצמה של התכנית.(לאחר הקימפול)

שאלה 2.1.3:

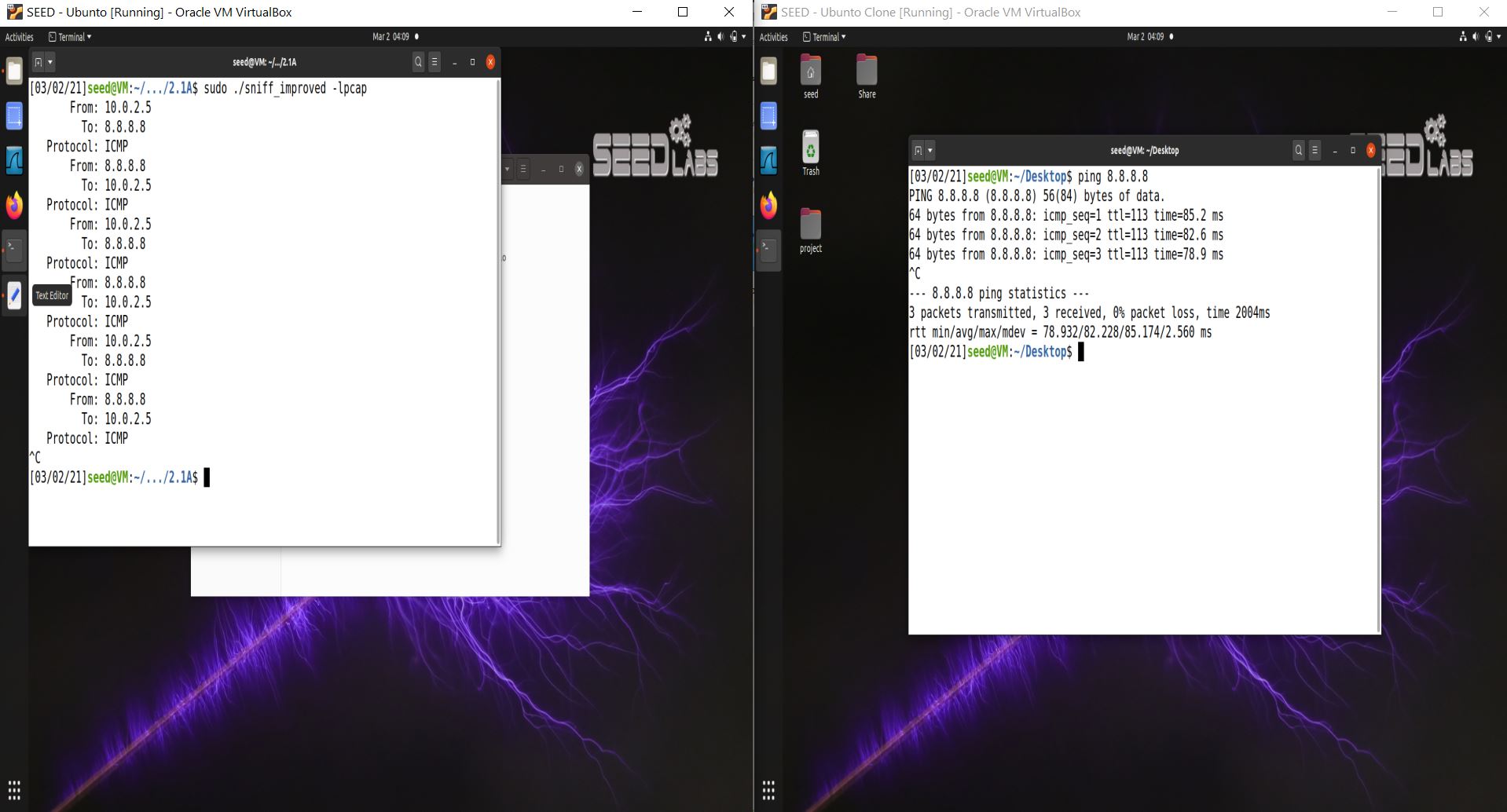
כאשר מפעילים p.m אנחנו מרחרחים כל פאקטה שעוברת בכרטיס הרשת. אבל, כאשר p.m כבוי אנחנו יכולים לרחרח רק פאקטות שמיועדות אלינו או יוצאות מאיתנו.

תמונה ראשונה: מצב p.m היה דלוק.

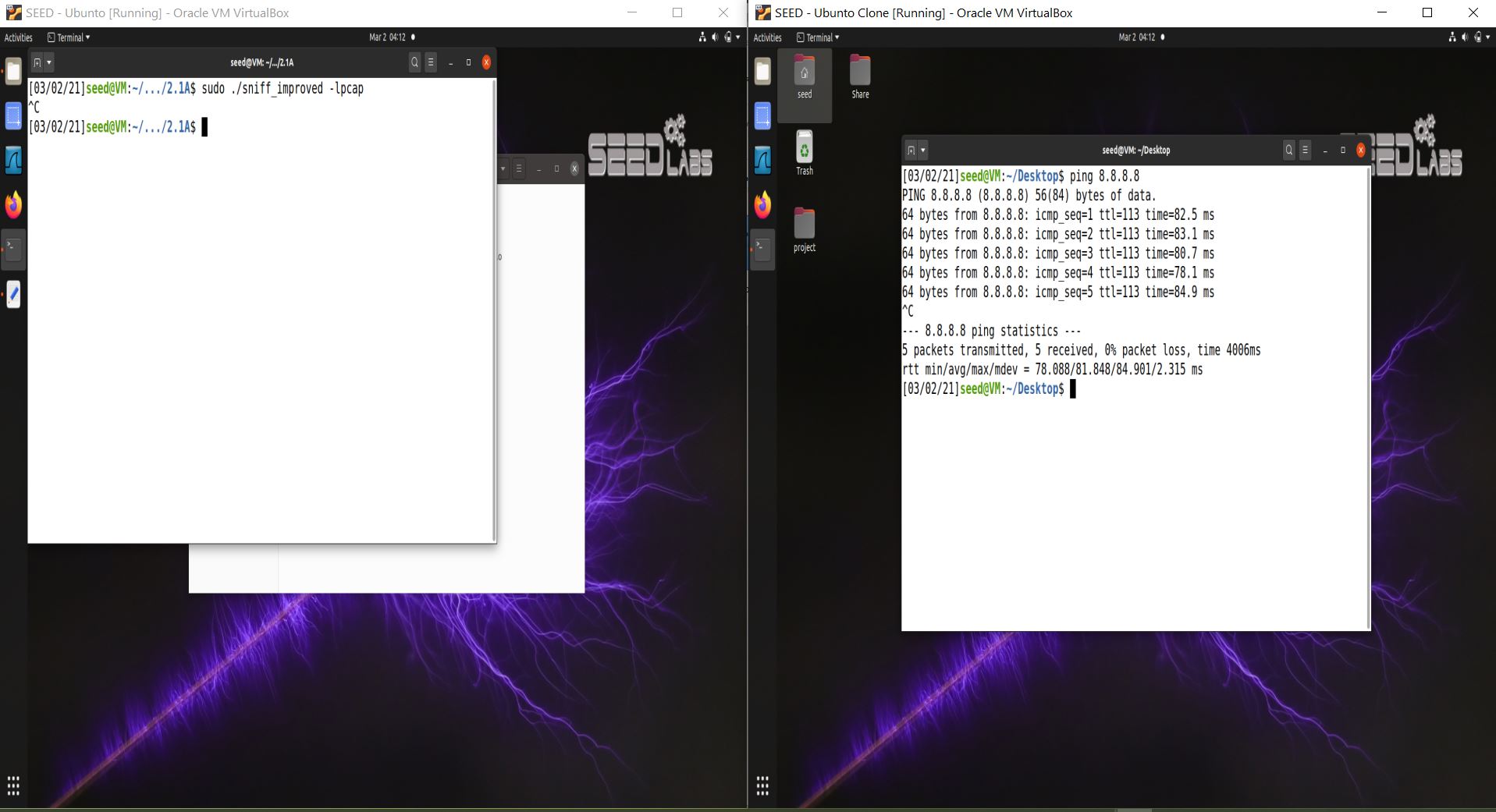
בקשה של המכונה שנייה נקלטת במכונה הזאת

10.0.2.5

10.0.2.4



תמונה שנייה: מצב p.m היה כבוי.



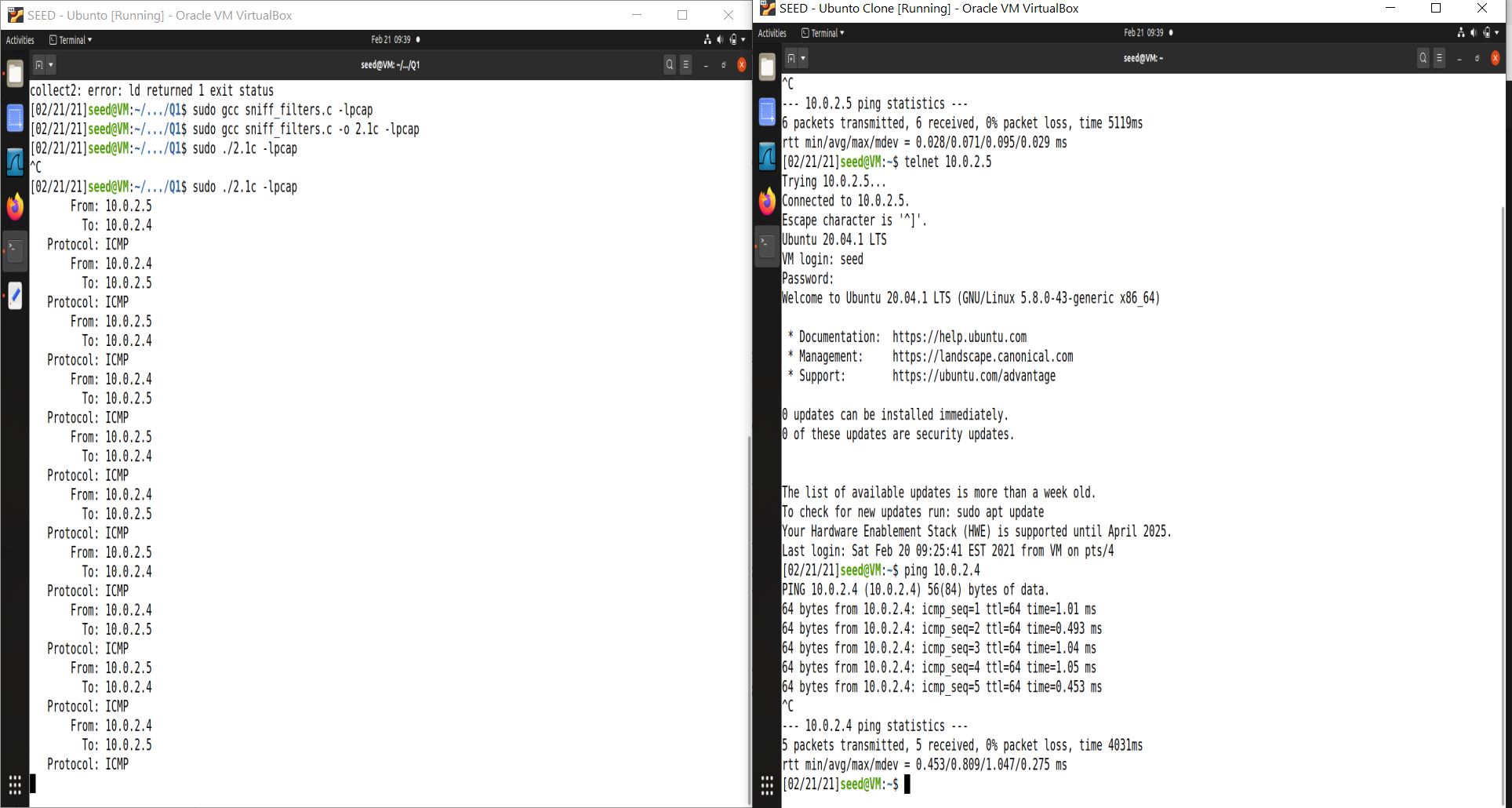
10.0.2.4

בשונה מהתמונה הקודמת, כאן לא קלטנו שום בקשה של המכונה השנייה כאשר p.m היה כבוי.

10.0.2.5

משימה B2.1:

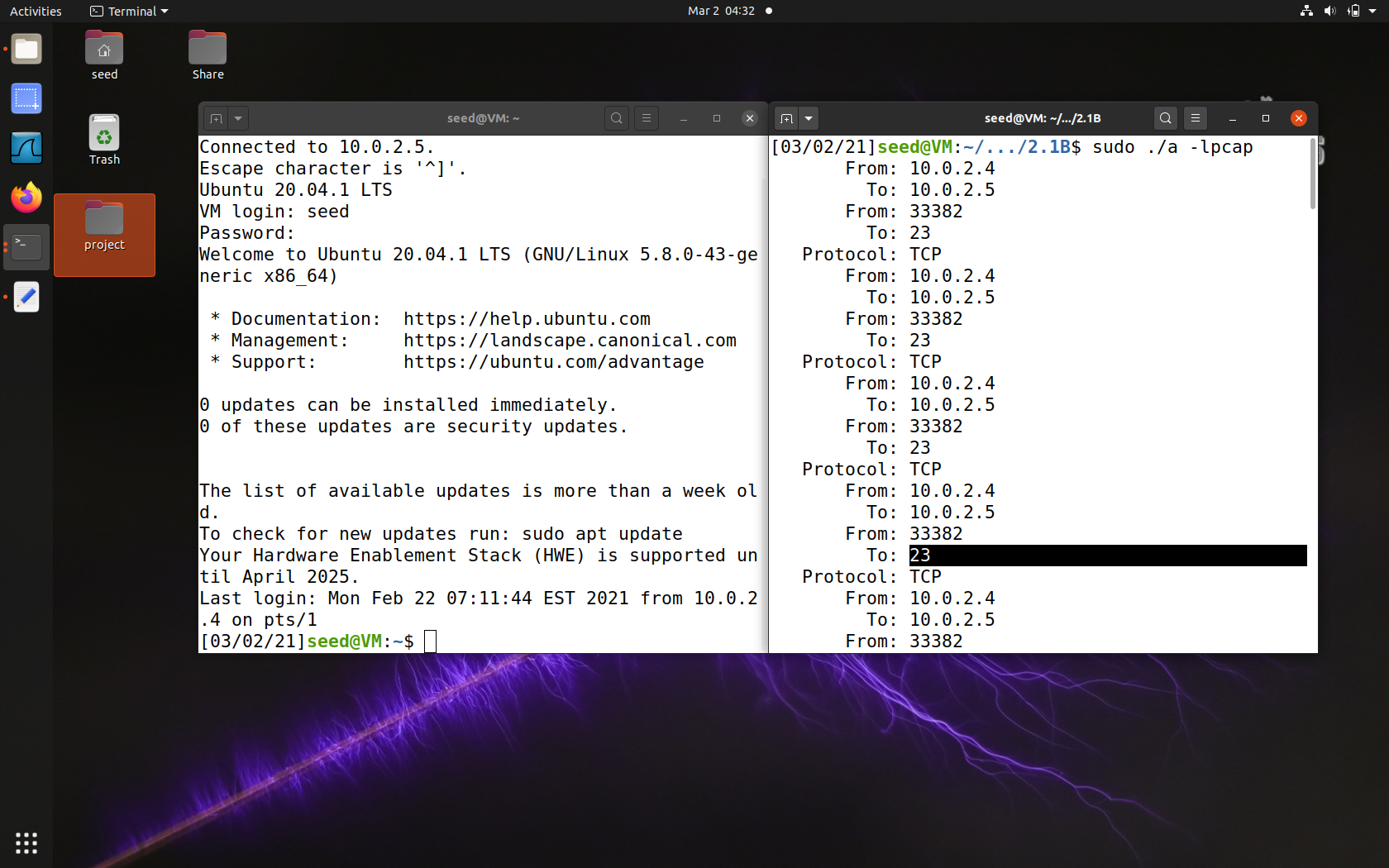
בתמונה כאן אנחנו תופסים רק חבילות מסוג ICMP.



10.0.2.4

10.0.2.5

בתמונה כאן אנחנו תופסים חבילות מסוג TCP ובטווח פורטים מסוים(10-100).

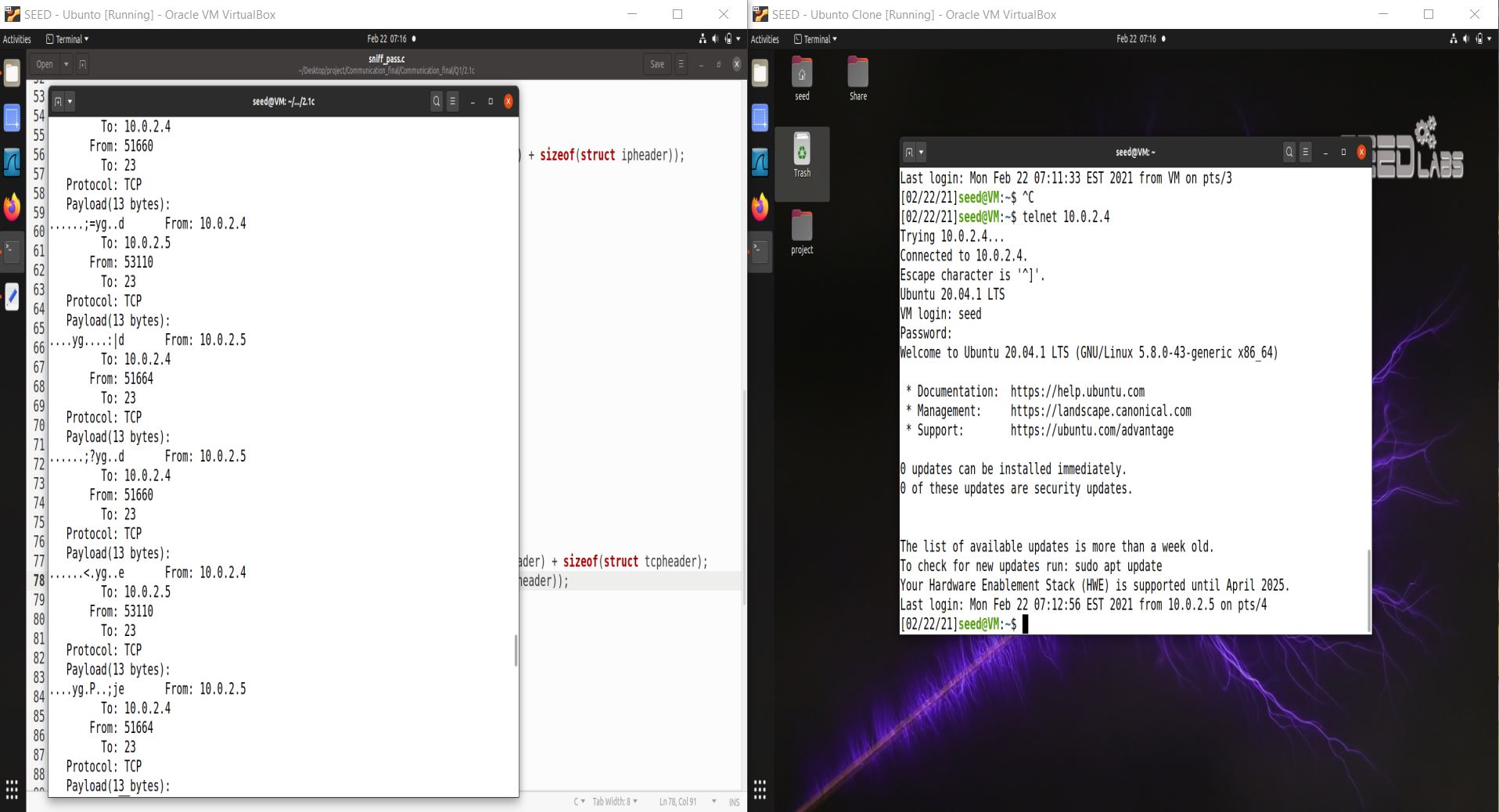


10 < dest port = 23 < 100

10.0.2.4

משימה C2.1:

כאן אנחנו נרצה לתפוס פאקטה ולשלוף ממנה את הסיסמא שנשלחה. המידע ששלחנו עבר באמצעות בקשת telnet שהיא מסוג TCP ומכילה בתוכה את הסיסמא למכונה השולחת את הבקשה.

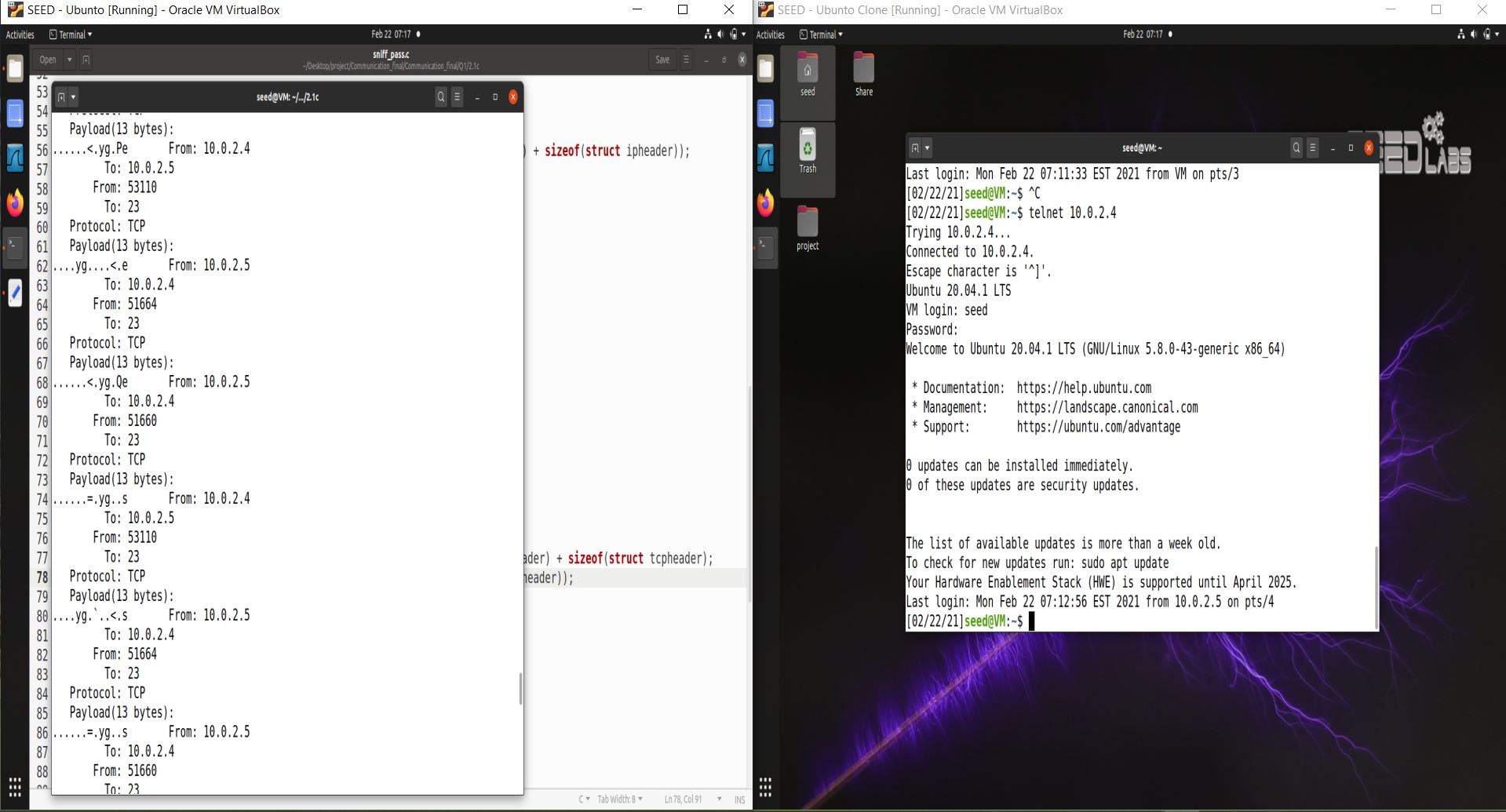


d

e

10.0.2.4

10.0.2.5



10.0.2.5

10.0.2.4

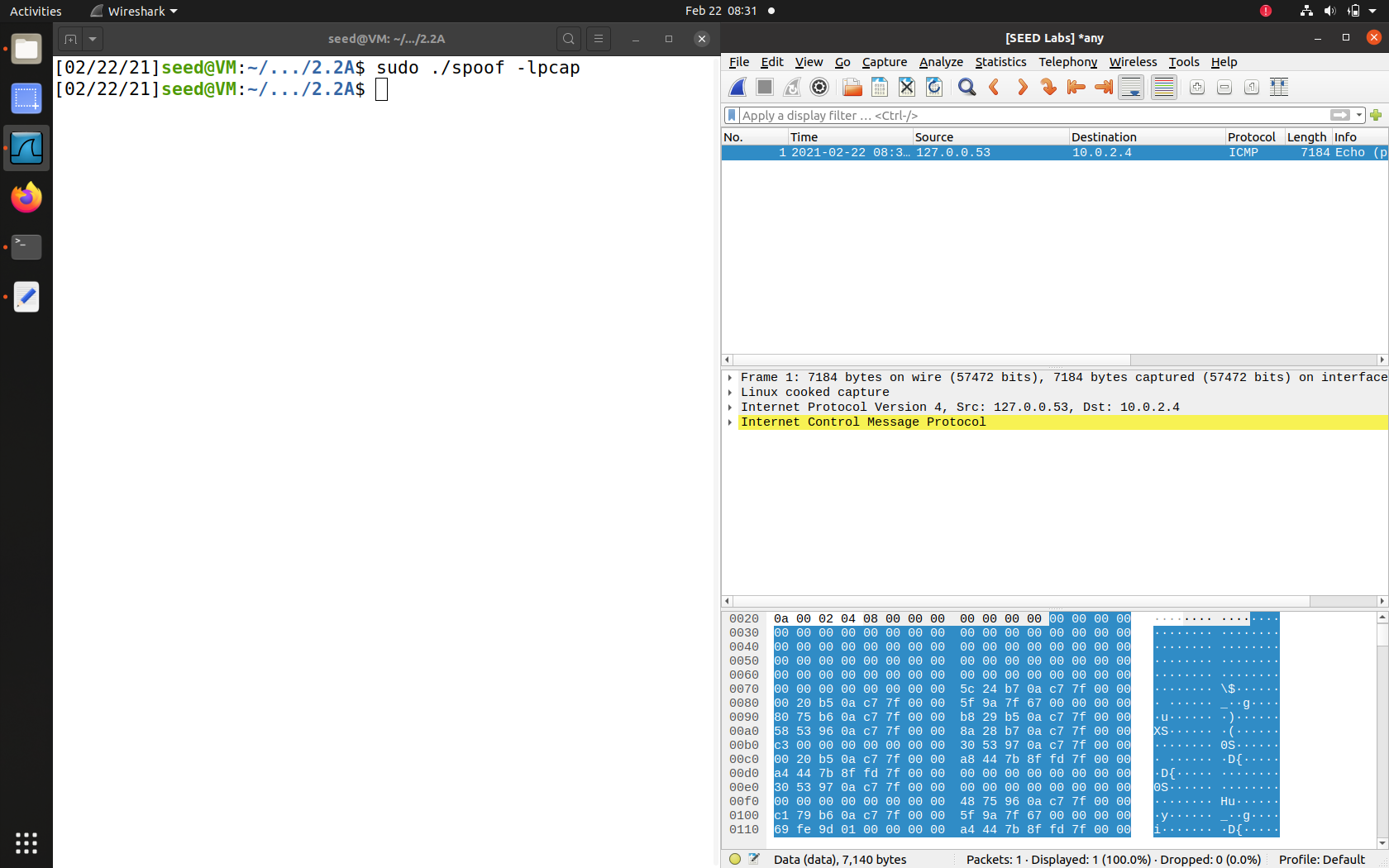
s

e

ניתן לראות בחצים שהצלחנו להוציא מכל חבילה את הדאטה שלה ובכל חבילה היה קיים חלק מהסיסמא כולה אותה חיפשנו.

משימה A2.2:

כאן אנחנו כתבנו תוכנית שתעשה ספופינג ותשלח חבילה מסוג ICMP מכתובת IP כלשהי אל המכונה שלנו.

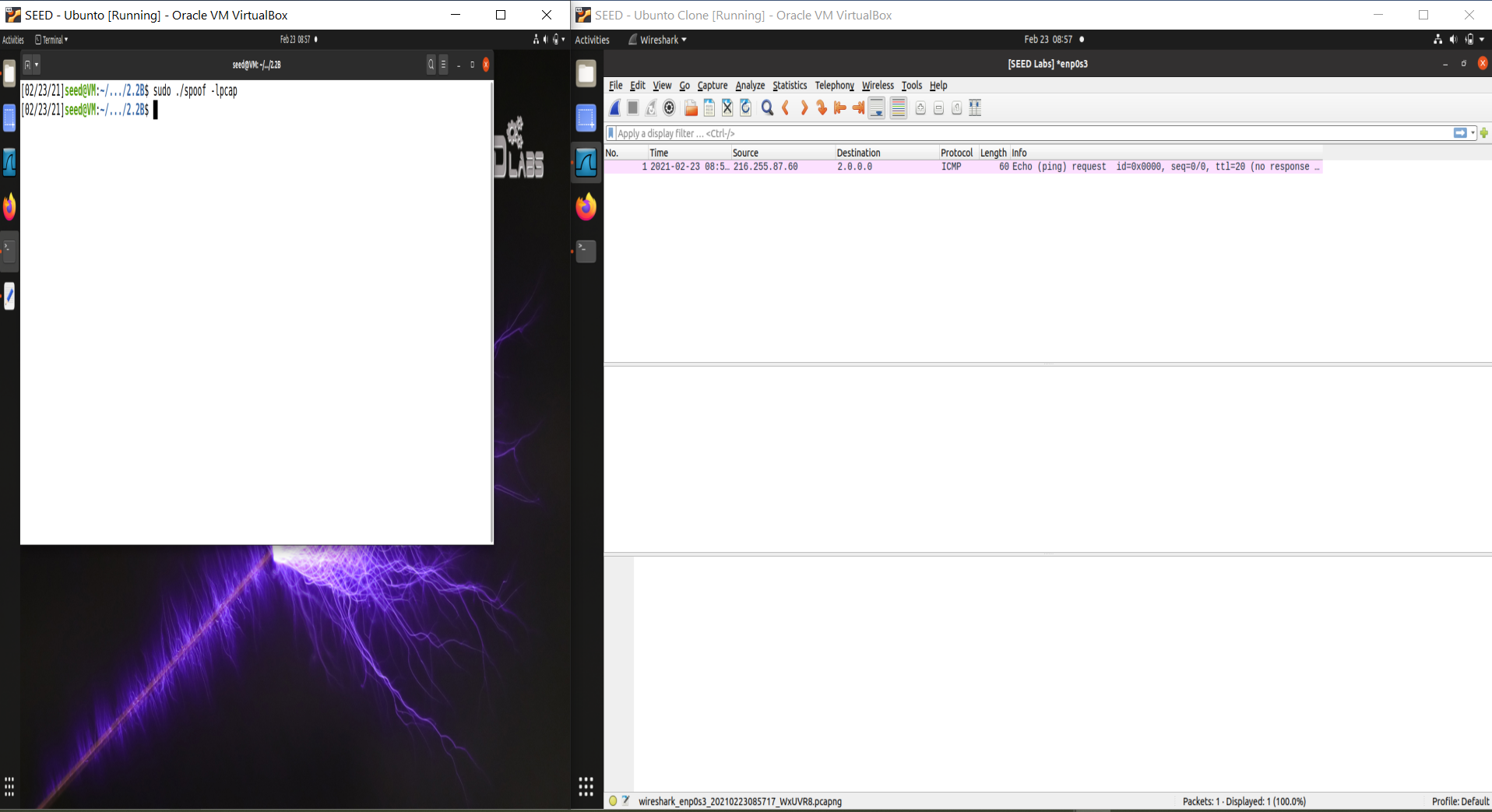


כתובת כלשהי

כתובת של המכונה שלי

משימה B2.2:

במשימה הזאת אנחנו הפעלנו את המכונה הראשית (10.0.2.4) וממנה עשינו ספופינג למכונה השנייה (10.0.2.5) מכתובת כלשהי.



10.0.2.4

Request

10.0.2.5

Some IP

שאלה 2.2.4:

לאחר שניסינו לשנות לגודל שרירותי כלשהו, אנו יכולים להסיק שאי אפשר מכיוון שהקוד רץ אבל שלח שגיאה כל פעם.

שאלה 2.2.5:

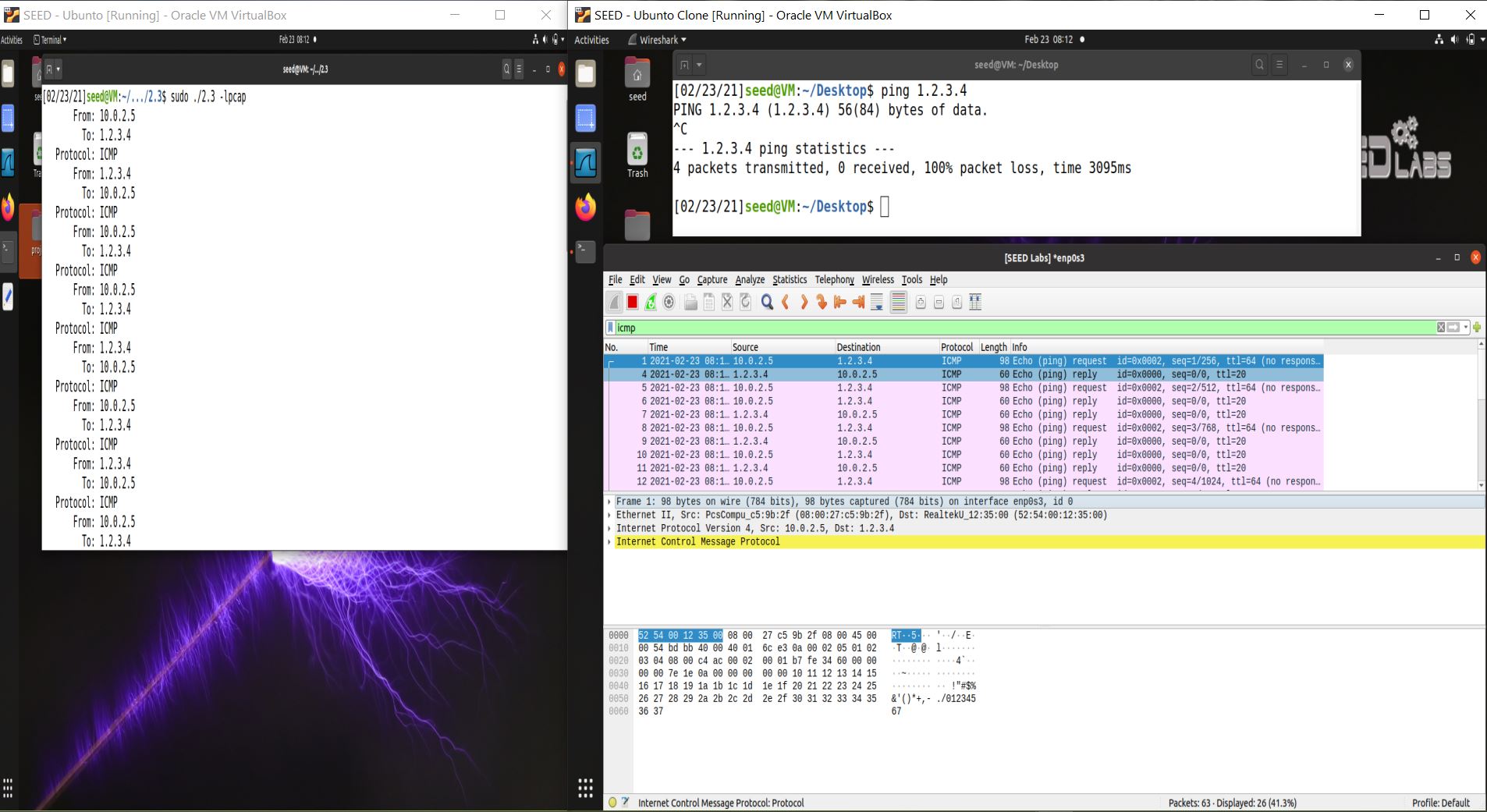
לא חייב, המערכת הפעלה דואגת לחשב לבד. בקוד שלנו ניתן לראות שאין חישוב של check sum.

שאלה 2.2.6:

אנו חייבים הרשאות root כי בתרגיל הזה אנחנו שלחנו פאקטות שאנחנו הרכבנו בsocket שפתחנו במיוחד. גישה כזאת לרכיבים של מערכת ההפעלה מצריכה כמובן, הרשאות גבוהות יותר.

משימה 2.3:

במשימה הזאת אנחנו עושים מאזינים ממכונה אחת (10.0.2.4) למכונה השנייה(10.0.2.5) ששולחת פינג לכתובת לא קיימת, ועבור כל חבילה שנתפוס אנחנו עושים ספופינג ושולחים הודעת reply למכונה השנייה מהכתובת שלא קיימת מהמכונה ראשונה.



10.0.2.4

10.0.2.5

כתובת לא קיימת

Reply messages for each request.

4 requests messages