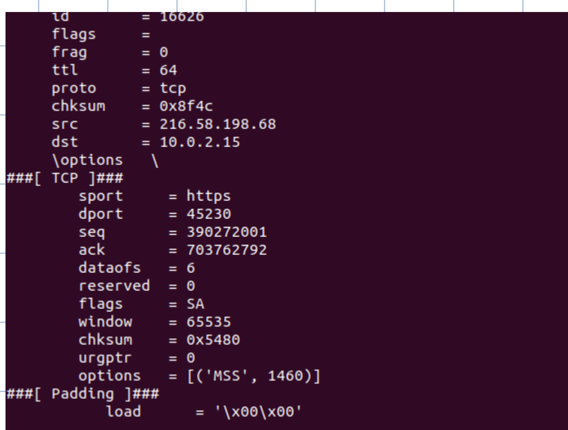
פרויקט גמר

* כל הקודים מצורפים בתיקייה, לצד כל סעיף יהיה כתוב שם הקובץ הרלוונטי.

A1.1: קובץ הקוד בשם Sniff.py

כאשר הרצנו את הקובץ sniffer עם פקודת Sudo המכונה פעלה וה Sniffer החל בהסנפת פאקטות (נכנס למצב "האזנה", לכמות ממונה של פאקטות שמספרה).



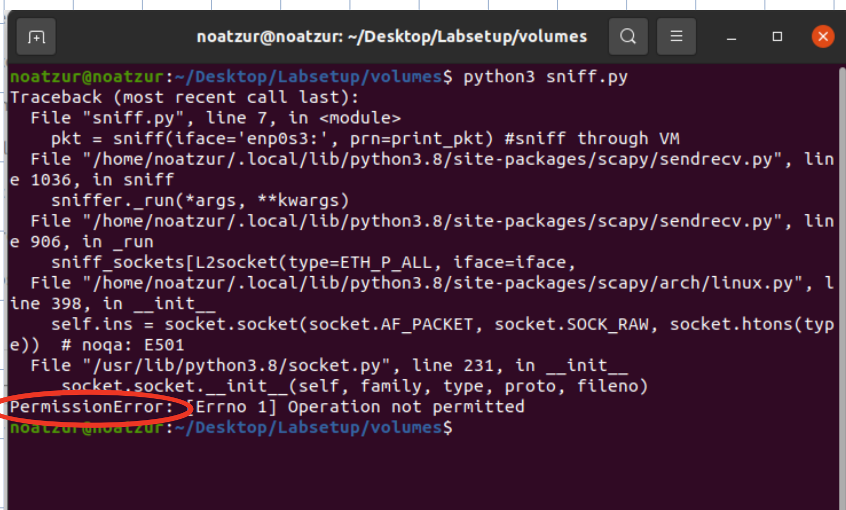
```
    id        = 16626
    flags     =
    frag      = 0
    ttl       = 64
    proto     = tcp
    chksum    = 0x8f4c
    src       = 216.58.198.68
    dst       = 10.0.2.15
    \options   \
###[ TCP ]###
       sport     = https
       dport     = 45230
       seq       = 390272001
       ack       = 703762792
       dataofs   = 6
       reserved  = 0
       flags     = SA
       window    = 65535
       chksum    = 0x5480
       urgptr    = 0
       options   = [('MSS', 1460)]
###[ Padding ]###
          load      = '\x00\x00'
```

דוגמא לפאקטת TCP שהוסנפה ע"י הרצת הקובץ עם Sudo:

אך כאשר הרצנו את הפקודה ללא Sudo, קפצה שגיאת הרשאה, זאת רק משתמש בעל הרשאות Admin יכול להריץ את המכונה של ה Sniffer, זאת כי פונקציות בספריה של Scapy אשר בעזרתן מבצעים את ההסנפה, דורשות הרשאה זו.

כמו שלמדנו בהרצאה, ה interface מוכנס למצב "מוניטור" (פרוץ) ולא נרצה שכל אחד עם גישה למחשב שהוא לא Admin יוכל לעשות זאת.

בפונ' של Scapy כמו פונקצית ה Sniff שדורשת הרשאה של מנהל
כדי שלא כל משתמש רגיל יוכל להסתכל על התעבורה/ליצור פאקטות
מניפולציות/לגשת לשדות מסוימים בפאקטים.
הפונ' האלו ניגשות לכרטיס הרשת ובעצם פותחות raw socket
שמאפשר למשתמש לשמש תעבורה שלא שלו (נצטרך הרשאות מנהל.
(ניתן לראות את הפונקציה עפ"י סדר קריאות שצריך שגיאה עקב כך שלא
הרצנו עם sudo.)

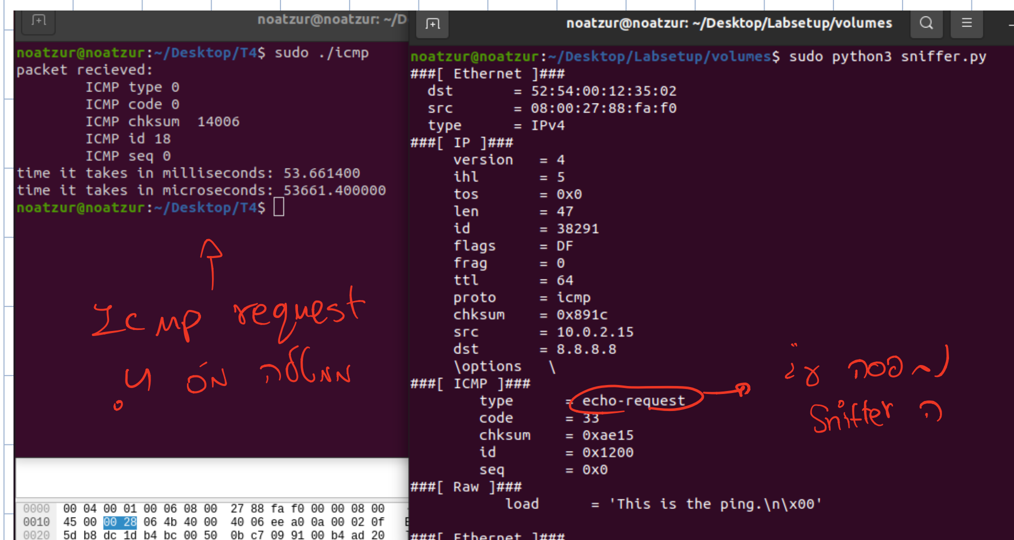


:1.1B Scapy's filter הקוד בקובץ בשם Sniffer.py

• ICMP - ע״י הכנסת filter לפונק׳ Sniff:

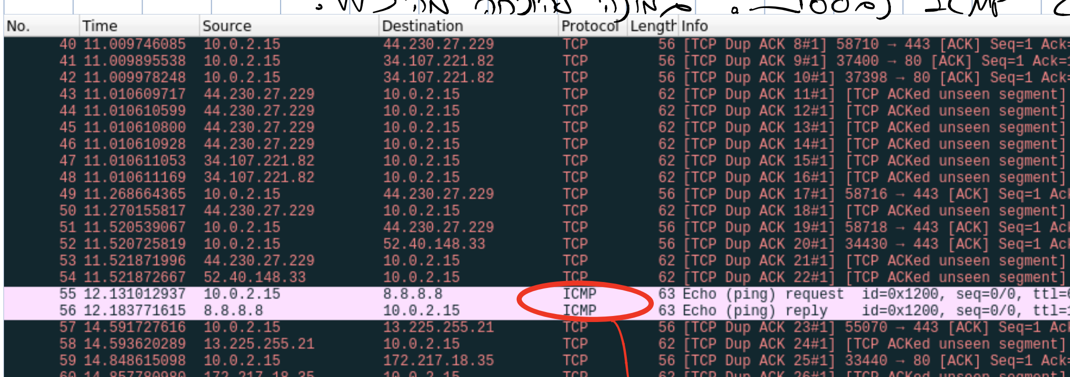
Sniff(iface='enp0s3', filter='icmp', prn=print_pkt)

השתמשנו בשליחת Icmp request משאלה הקודמת.
הפעלנו את הSniffer ושלחנו את הIcmp request. נ״ו



(נתפסה גם פקטת הreply שהתקבלה עבור הrequest)

בנוסף, בדקנו בWireshark היה פתוח במקביל כדי לוודא שרק פקטים מסוג ICMP נתפסו. מאומת בתמונה מהWS:



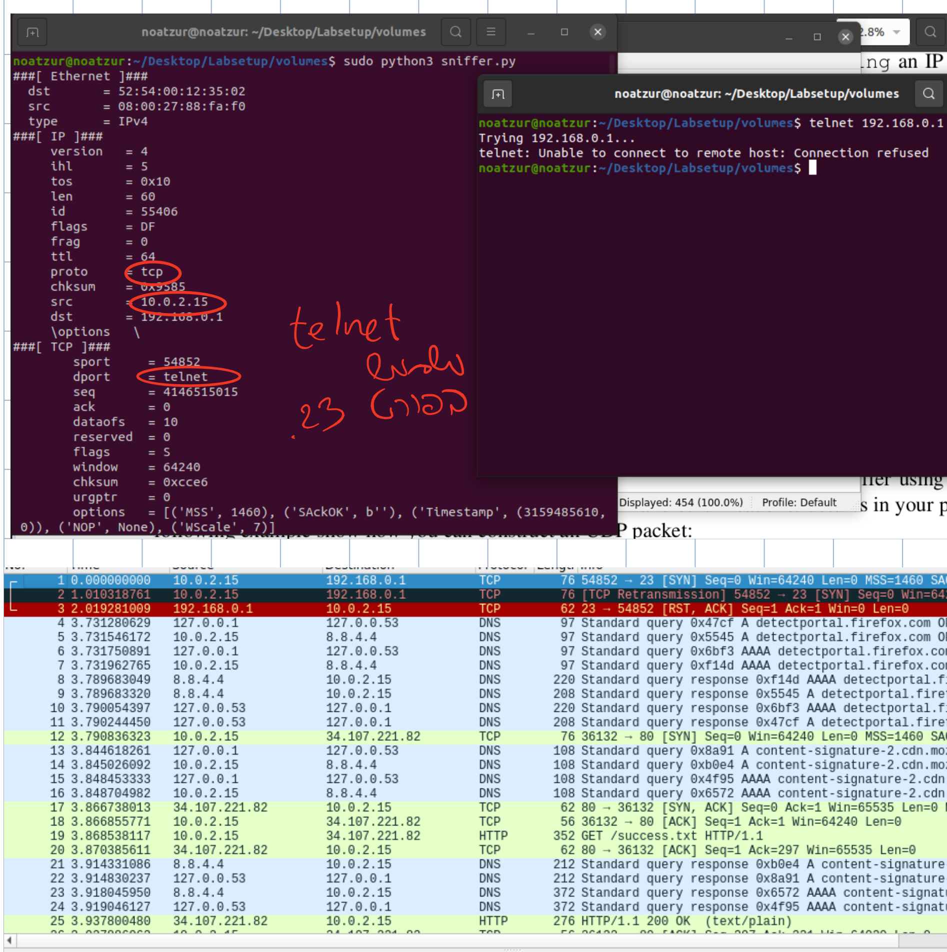
No.	Time	Source	Destination	Protocol	Length	Info
40	11.009746085	10.0.2.15	44.230.27.229	TCP	56	[TCP Dup ACK 8#1] 58710 → 443 [ACK] Seq=1 Ack=
41	11.009895538	10.0.2.15	34.107.221.82	TCP	56	[TCP Dup ACK 9#1] 37400 → 80 [ACK] Seq=1 Ack=1
42	11.009978248	10.0.2.15	34.107.221.82	TCP	56	[TCP Dup ACK 10#1] 37398 → 80 [ACK] Seq=1 Ack=
43	11.010609717	44.230.27.229	10.0.2.15	TCP	62	[TCP Dup ACK 11#1] [TCP ACKed unseen segment]
44	11.010610599	44.230.27.229	10.0.2.15	TCP	62	[TCP Dup ACK 12#1] [TCP ACKed unseen segment]
45	11.010610800	44.230.27.229	10.0.2.15	TCP	62	[TCP Dup ACK 13#1] [TCP ACKed unseen segment]
46	11.010610928	44.230.27.229	10.0.2.15	TCP	62	[TCP Dup ACK 14#1] [TCP ACKed unseen segment]
47	11.010611053	34.107.221.82	10.0.2.15	TCP	62	[TCP Dup ACK 15#1] [TCP ACKed unseen segment]
48	11.010611169	34.107.221.82	10.0.2.15	TCP	62	[TCP Dup ACK 16#1] [TCP ACKed unseen segment]
49	11.268664365	10.0.2.15	44.230.27.229	TCP	56	[TCP Dup ACK 17#1] 58716 → 443 [ACK] Seq=1 Ack
50	11.270155817	44.230.27.229	10.0.2.15	TCP	62	[TCP Dup ACK 18#1] [TCP ACKed unseen segment]
51	11.520539067	10.0.2.15	44.230.27.229	TCP	56	[TCP Dup ACK 19#1] 58718 → 443 [ACK] Seq=1 Ack
52	11.520725819	10.0.2.15	52.40.148.33	TCP	56	[TCP Dup ACK 20#1] 34430 → 443 [ACK] Seq=1 Ack
53	11.521871996	44.230.27.229	10.0.2.15	TCP	62	[TCP Dup ACK 21#1] [TCP ACKed unseen segment]
54	11.521872667	52.40.148.33	10.0.2.15	TCP	62	[TCP Dup ACK 22#1] [TCP ACKed unseen segment]
55	12.131012937	10.0.2.15	8.8.8.8	ICMP	63	Echo (ping) request id=0x1200, seq=0/0, ttl=6
56	12.183771615	8.8.8.8	10.0.2.15	ICMP	63	Echo (ping) reply id=0x1200, seq=0/0, ttl=1
57	14.591727616	10.0.2.15	13.225.255.21	TCP	56	[TCP Dup ACK 23#1] 55070 → 443 [ACK] Seq=1 Ack
58	14.593620289	13.225.255.21	10.0.2.15	TCP	62	[TCP Dup ACK 24#1] [TCP ACKed unseen segment]
59	14.848615098	10.0.2.15	172.217.18.35	TCP	56	[TCP Dup ACK 25#1] 33440 → 80 [ACK] Seq=1 Ack=
60	14.857780980	172.217.18.35	10.0.2.15	TCP	62	[TCP Dup ACK 26#1] [TCP ACKed unseen segment]

רק 2 אלו הופיעו.

• כאן התבקשנו לבצע פילטר של פאקט TCP
שמגיע מ- IP מסוים ומפורט 23 (telnet).

Sniff(iface='enp0s3', filter='tcp and host 10.0.2.15 and dst port 23', prn=print_pkt)

על מנת לבדוק את עצמנו, בדקנו את השלבים הבאים:
כאשר הפעלנו את ה Sniffer, פתחנו גם דפדפן אינטרנט.
שום פאקטה לא נתפסה ע"י ה Sniffer.
פתחנו terminal נוסף ובדקנו שליחה ע"י פרוטוקול telnet.
פרוטוקול telnet עשה שימוש בפורט 23,
ויוצר חיבור TCP.



רק פאקטות מ IP
10.0.2.15
שהועברו ל dport
23 נתפסו:

הפאקטה הראשונה
היא מפורט 23
כפי שניתן לראות
מה IP 10.0.2.15
אך ישנם גם פאקטות
נוספות מפורטים
אחרים וIP שונים
שלא הוספנו.

• בחרנו particular Subnet 128.230.0.0/16.

* לקחתי מאתר ipinfo.io אתר של אוניברסיטת Syracuse
עם IP ← 128.230.18.123 כדי לגלוש לאתר ב-Subnet המצוין מלמעלה.

Sniff (i face =' en p 053', filter =' net 128.230.0.0/16', prn=print-Pkt)

הסנפה – פאקטה ע"י
ה Sniffer

```
###[ Ethernet ]###
  dst       = 52:54:00:12:35:02
  src       = 08:00:27:88:fa:f0
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 40
     id        = 14743
     flags     = DF
     frag      = 0
     ttl       = 64
     proto     = tcp
     chksum    = 0x61c9
     src       = 10.0.2.15
     dst       = 128.230.18.123
     \options   \
###[ TCP ]###
        sport     = 58870
        dport     = https
        seq       = 1149736888
        ack       = 481152953
        dataofs   = 5
        reserved  = 0
        flags     = A
        window    = 63986
        chksum    = 0x9f8a
        urgptr    = 0
        options   = []
```

זהו האוניברסיטה
ב Subnet
המתאים.

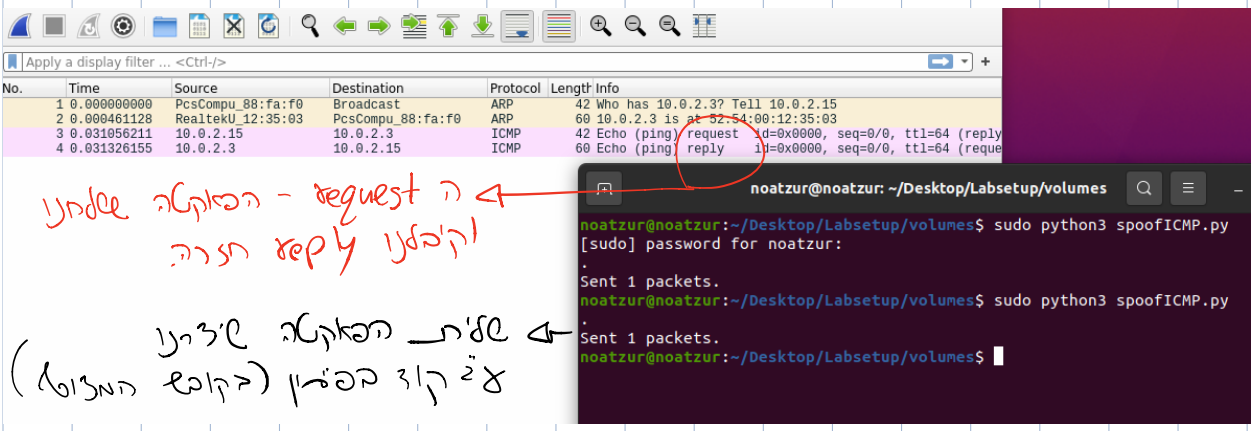
תמונה מ-Wireshark
מאותה הסנפה:

No.	Time	Source	Destination	Protocol	Length	Info
246	7.627144522	216.58.205.196	10.0.2.15	TLSv1.2	2909	Application Data, Application Data, Application Data, Applica…
247	7.627196557	10.0.2.15	216.58.205.196	TCP	54	55128 → 443 [ACK] Seq=710 Ack=13249 Win=62480 Len=0
248	7.628415106	128.230.18.63	10.0.2.15	TCP	60	443 → 41768 [ACK] Seq=129972 Ack=2012 Win=65535 Len=0
249	7.636230395	10.0.2.15	216.58.205.196	TLSv1.2	93	Application Data
250	7.638268243	216.58.205.196	10.0.2.15	TCP	60	443 → 55128 [ACK] Seq=13249 Ack=749 Win=65535 Len=0
251	7.654356881	69.25.199.41	10.0.2.15	TLSv1.3	214	Server Hello
252	7.654430099	10.0.2.15	69.25.199.41	TCP	54	46260 → 443 [ACK] Seq=514 Ack=161 Win=64080 Len=0
253	7.657573723	69.25.199.41	10.0.2.15	TLSv1.3	4314	Change Cipher Spec
254	7.657635223	10.0.2.15	69.25.199.41	TCP	54	46260 → 443 [ACK] Seq=514 Ack=4421 Win=62480 Len=0
255	7.658155907	8.8.4.4	10.0.2.15	DNS	132	Standard query response 0xf38c AAAA syracuse.edu SOA ns1.syr…
256	7.658717424	69.25.199.41	10.0.2.15	TLSv1.3	1153	Application Data
257	7.658756384	10.0.2.15	69.25.199.41	TCP	54	46260 → 443 [ACK] Seq=514 Ack=5520 Win=61381 Len=0
258	7.665592056	69.25.199.41	10.0.2.15	TLSv1.3	648	Application Data
259	7.665652276	10.0.2.15	69.25.199.41	TCP	54	46260 → 443 [ACK] Seq=514 Ack=6114 Win=63900 Len=0
260	7.677081332	10.0.2.15	69.25.199.41	TLSv1.3	134	Change Cipher Spec, Application Data
261	7.677690018	10.0.2.15	69.25.199.41	TLSv1.3	413	Application Data
262	7.678631391	69.25.199.41	10.0.2.15	TCP	60	443 → 46260 [ACK] Seq=6114 Ack=594 Win=65535 Len=0
263	7.678631760	69.25.199.41	10.0.2.15	TCP	60	443 → 46260 [ACK] Seq=6114 Ack=953 Win=65535 Len=0
264	7.731173737	128.230.18.123	10.0.2.15	TLSv1.2	89	Application Data
265	7.731237052	10.0.2.15	128.230.18.123	TCP	54	58834 → 443 [ACK] Seq=471 Ack=193 Win=63900 Len=0
266	7.746661101	10.0.2.15	172.217.171.232	TLSv1.2	147	Application Data
267	7.748230440	172.217.171.232	10.0.2.15	TCP	60	443 → 58252 [ACK] Seq=164 Ack=244 Win=65535 Len=0

רק הפאקטה
שמצאתי
של IP הזה
הוסנפה.

1.2 : Spoof

במשימה זו נדרשנו לשלוח בקשות ICMP (request) ולקבל תשובות (reply). לשם כך כתבנו את הקוד הבא בקובץ spoofICMP.py



ה request - המחקה שלנו וקיבלנו reply חזרה.

שלחנו המחקה שיצרנו עי"י הקוד בספרון (בקובץ המצורף)

הרצנו גם כדי שנוכל לראות יותר טוב את התוכן של האובייקט IP שיצרנו.

```
3 a=IP()
4 a.dst = '10.0.2.3'
5 b = ICMP()
6 p=a/b
7 send(p)
8 print("lis(a) : ")
9 ls(a)
```

```
^Z
[1]+  Stopped                 sudo python3 sniffer.py
noatzur@noatzur:~/Desktop/Labsetup/volumes$ sudo python3 spoofICMP.py
.
Sent 1 packets.
lis(a) :
version    : BitField  (4 bits)                = 4               (4)
ihl        : BitField  (4 bits)                = None            (None)
tos        : XByteField                        = 0               (0)
len        : ShortField                        = None            (None)
id         : ShortField                        = 1               (1)
flags      : FlagsField  (3 bits)              = <Flag 0 ()>     (<Flag 0 ()>)
frag       : BitField  (13 bits)               = 0               (0)
ttl        : ByteField                         = 64              (64)
proto      : ByteEnumField                     = 0               (0)
chksum     : XShortField                       = None            (None)
src        : SourceIPField                     = '10.0.2.15'     (None)
dst        : DestIPField                       = '10.0.2.3'      (None)
options    : PacketListField                   = []              ([])
noatzur@noatzur:~/Desktop/Labsetup/volumes$
```

1.3 : traceroute

קובץ בשם ttl.py

במשימה זו, שלחנו מחקה ICMP לdestination IP - אוניברסיטה אראל. ורצינו לבדוק מה ה-ttl, כלומר כמה hops (routers) המחקה מעבור עד שיגיע ליעדה. בשביל זה הרצנו לולאה for שתריץ את השורה כל פעם שינה את ה ttl ותשלח אותו ב-1 מה שיגדיל.

שמנו את הלולאה בהתחלה שתרוץ 50 פעמים.
ב-wireshark ראינו שכאשר ה-ttl הגיע ל-15
קיבלנו reply בפעם הראשונה מכיוון (נניח) שהבין שלאחר שהפקטה
עברה כ-15 hops, היא הגיעה ל-destination

0-15

No.	Time	Source	Destination	Protocol	Length	Info
5	0.056847681	10.0.2.15	34.96.118.58	ICMP	42	Echo (ping) request id=0x0000, seq=0/0 ttl=0 (no response f…
6	0.057008557	10.0.2.2	10.0.2.15	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
7	0.096977099	10.0.2.15	34.96.118.58	ICMP	42	Echo (ping) request id=0x0000, seq=0/0 ttl=1 (no response f…
8	0.097166992	10.0.2.2	10.0.2.15	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
9	0.140823626	10.0.2.15	34.96.118.58	ICMP	42	Echo (ping) request id=0x0000, seq=0/0 ttl=2 (no response f…
10	0.144041676	192.168.0.1	10.0.2.15	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
11	0.179104207	10.0.2.15	34.96.118.58	ICMP	42	Echo (ping) request id=0x0000, seq=0/0 ttl=3 (no response f…
12	0.181271716	192.168.1.1	10.0.2.15	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
13	0.216481892	10.0.2.15	34.96.118.58	ICMP	42	Echo (ping) request id=0x0000, seq=0/0 ttl=4 (no response f…
14	0.227649339	10.174.128.1	10.0.2.15	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
15	0.253163583	10.0.2.15	34.96.118.58	ICMP	42	Echo (ping) request id=0x0000, seq=0/0 ttl=5 (no response f…
16	0.292638658	10.0.2.15	34.96.118.58	ICMP	42	Echo (ping) request id=0x0000, seq=0/0 ttl=6 (no response f…
17	0.314294796	172.17.3.101	10.0.2.15	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
18	0.328786915	10.0.2.15	34.96.118.58	ICMP	42	Echo (ping) request id=0x0000, seq=0/0 ttl=7 (no response f…
19	0.365740462	10.0.2.15	34.96.118.58	ICMP	42	Echo (ping) request id=0x0000, seq=0/0 ttl=8 (no response f…
20	0.378454252	212.25.116.149	10.0.2.15	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
21	0.405161187	10.0.2.15	34.96.118.58	ICMP	42	Echo (ping) request id=0x0000, seq=0/0 ttl=9 (no response f…
22	0.420958402	10.25.19.10	10.0.2.15	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
23	0.440497930	10.0.2.15	34.96.118.58	ICMP	42	Echo (ping) request id=0x0000, seq=0/0 ttl=10 (no response …
24	0.451857419	212.25.77.14	10.0.2.15	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
25	0.480602529	10.0.2.15	34.96.118.58	ICMP	42	Echo (ping) request id=0x0000, seq=0/0 ttl=11 (no response …
26	0.491926007	10.90.99.13	10.0.2.15	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
27	0.517199443	10.0.2.15	34.96.118.58	ICMP	42	Echo (ping) request id=0x0000, seq=0/0 ttl=12 (no response …
28	0.560780487	10.0.2.15	34.96.118.58	ICMP	42	Echo (ping) request id=0x0000, seq=0/0 ttl=13 (no response …
29	0.569996997	74.125.51.88	10.0.2.15	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
30	0.600897579	10.0.2.15	34.96.118.58	ICMP	42	Echo (ping) request id=0x0000, seq=0/0 ttl=14 (no response …
31	0.614191776	74.125.244.225	10.0.2.15	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
32	0.641453005	10.0.2.15	34.96.118.58	ICMP	42	Echo (ping) request id=0x0000, seq=0/0 ttl=15 (reply in 34)
33	0.651938050	72.14.234.95	10.0.2.15	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
34	0.695357519	34.96.118.58	10.0.2.15	ICMP	60	Echo (ping) reply id=0x0000, seq=0/0, ttl=111 (request in …

reply

1.4 Sniff & spoof בקובץ sniff_spoof.py

בפועל לא השתמשנו ב-2 מכונות אלא ב docker שהופעל במכונה 1.
שימוש ב docker (נעשה ע"י הפקודה docker-compose build
ו-up. פתחנו 2 טרמינלים ב vm, בראשון הפעלנו את הcontainers
של התוקף ע"י הפקודה ps ו exec (ציינו את id של ה-container הרלוונטי
ופתחנו אותו.)
בטרמינל השני פתחנו ע"י אותה דרך את ה container
של ה user "הרגיל".

```
noatzur@noatzur:~/Desktop/Labsetup$ sudo docker ps
[sudo] password for noatzur:
CONTAINER ID        IMAGE                                  COMMAND
  CREATED             STATUS              PORTS              NAMES
9d56b5cb125d        handsonsecurity/seed-ubuntu:large    "/bin/sh -c /bin/bash"
  50 seconds ago      Up 46 seconds                          host-10.9.0.5
30001aa1e150        handsonsecurity/seed-ubuntu:large    "/bin/sh -c /bin/bash"
  50 seconds ago      Up 47 seconds                          seed-attacker
```

user (9d56b5cb125d)

attacker (30001aa1e150)

בקובץ ה Sniff_spoof שינינו את פונ ה Sniff כך שתקבל כפרמטר של iface את השם של ה interface של הContainer כמו שהסברנו בחלק המשימה (מתחיל ב- br-)

ואז הפונ שמופעלת בכל פעם שהוספנה פאקטה מסוג ICMP.

הפונ החדשה מקבלת את הפאקטה שנתפסה, ויוצרת פאקטה חדשה מסוג ICMP. הפאקטה החדשה מורכבת מהשדות הבאים:

dest IP - לאן הפאקטה מיועדת

Src IP - ממי נשלחה.

Seq - מס הפאקטה מתוך סדר הפאקטות שנשלחו (על מנת לשמור על סדר שליחה וקבלה).

load - המידע שמכילה הפאקטה.

נוצר מכך שבנו את פאקטה חדשה, כך שה dst IP שלה יהיה ה-Src של הפאקטה שהתקבלה (שלחה הping)! Src IP שלה יהיה ה-dst של הפאקטה שנשלחה. כמו כן ה type של ה Icmp הוא 0 כלו' מסוג reply (request=8) וכך בעצם יצרנו הודעת reply מזויפת מ IP שאינו קיים בכלל.

בContainer של ה user ביצענו את פקודת ה ping עם הIP המבוקש ובContainer של התוקף, הרצנו את הקובץ Sniff_spoof

מבקי' ה- Volumes המשפר.

• ping 1.2.3.4

user מבצע ping ל 1.2.3.4

```
root@9d56b5cb125d:/# ping 1.2.3.4
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
64 bytes from 1.2.3.4: icmp_seq=1 ttl=64 time=80.2 ms
64 bytes from 1.2.3.4: icmp_seq=2 ttl=64 time=18.9 ms
64 bytes from 1.2.3.4: icmp_seq=3 ttl=64 time=26.8 ms
64 bytes from 1.2.3.4: icmp_seq=4 ttl=64 time=30.7 ms
64 bytes from 1.2.3.4: icmp_seq=5 ttl=64 time=28.8 ms
64 bytes from 1.2.3.4: icmp_seq=6 ttl=64 time=28.9 ms
64 bytes from 1.2.3.4: icmp_seq=7 ttl=64 time=40.2 ms
64 bytes from 1.2.3.4: icmp_seq=8 ttl=64 time=30.2 ms
64 bytes from 1.2.3.4: icmp_seq=9 ttl=64 time=24.4 ms
64 bytes from 1.2.3.4: icmp_seq=10 ttl=64 time=31.2 ms
64 bytes from 1.2.3.4: icmp_seq=11 ttl=64 time=18.2 ms
^Z
[11]+  Stopped                 ping 1.2.3.4
root@9d56b5cb125d:/#
```

הכתובת הזו לא מקינה, כי לא קיים host אשר מחזיק בה.
ובכל זאת התקבלו פונים כאילו באמת host מסוים שיש לו את
ה IP הזה קיבל את ה request והחזיר reply.
תמונה מה-wireshark של אותה הרצה:

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
74	23.015656423	10.0.2.15	52.35.31.120	TCP	56	42678 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
75	23.016715500	10.0.2.15	52.35.31.120	TLSv1.2	269	Client Hello
76	23.017377296	52.35.31.120	10.0.2.15	TCP	62	443 → 42678 [ACK] Seq=1 Ack=214 Win=65535 Len=0
77	23.242578324	52.35.31.120	10.0.2.15	TLSv1.2	1516	Server Hello
78	23.242624848	10.0.2.15	52.35.31.120	TCP	56	42678 → 443 [ACK] Seq=214 Ack=1461 Win=63900 Len=0
79	23.243575163	52.35.31.120	10.0.2.15	TLSv1.2	1997	Certificate, Server Key Exchange, Server Hello Done
80	23.243626104	10.0.2.15	52.35.31.120	TCP	56	42678 → 443 [ACK] Seq=214 Ack=3402 Win=62480 Len=0
81	23.249054309	10.0.2.15	52.35.31.120	TLSv1.2	182	Client Key Exchange, Change Cipher Spec, Encrypted Handshake …
82	23.249364020	52.35.31.120	10.0.2.15	TCP	62	443 → 42678 [ACK] Seq=3402 Ack=340 Win=65535 Len=0
83	23.251283117	10.0.2.15	52.35.31.120	TLSv1.2	2892	Application Data
84	23.251621684	52.35.31.120	10.0.2.15	TCP	62	443 → 42678 [ACK] Seq=3402 Ack=1800 Win=65535 Len=0
85	23.251621904	52.35.31.120	10.0.2.15	TCP	62	443 → 42678 [ACK] Seq=3402 Ack=3176 Win=65535 Len=0
86	23.351946293	10.9.0.5	1.2.3.4	ICMP	100	Echo (ping) request id=0x0018, seq=10/2560, ttl=64 (no respo…
87	23.351946293	10.9.0.5	1.2.3.4	ICMP	100	Echo (ping) request id=0x0018, seq=10/2560, ttl=64 (reply in…
88	[illegible]	10.0.2.15	1.2.3.4	ICMP	100	Echo (ping) request id=0x0018, seq=10/2560, ttl=63 (no respo…
89	23.383066671	1.2.3.4	10.9.0.5	ICMP	100	Echo (ping) reply id=0x0018, seq=10/2560, ttl=64 (request …
90	23.383092523	1.2.3.4	10.9.0.5	ICMP	100	Echo (ping) reply id=0x0018, seq=10/2560, ttl=64
91	[illegible]	52.35.31.120	10.0.2.15	TLSv1.2	107	Change Cipher Spec, Encrypted Handshake Message
92	23.473940668	10.0.2.15	52.35.31.120	TCP	56	42678 → 443 [ACK] Seq=3176 Ack=3453 Win=63900 Len=0
93	23.538839979	52.35.31.120	10.0.2.15	TLSv1.2	326	Application Data
94	23.538886135	10.0.2.15	52.35.31.120	TCP	56	42678 → 443 [ACK] Seq=3176 Ack=3723 Win=63900 Len=0
95	24.353006739	10.9.0.5	1.2.3.4	ICMP	100	Echo (ping) request id=0x0018, seq=11/2816, ttl=64 (no respo…
96	24.353006739	10.9.0.5	1.2.3.4	ICMP	100	Echo (ping) request id=0x0018, seq=11/2816, ttl=64 (reply in…
97	24.353153733	10.0.2.15	1.2.3.4	ICMP	100	Echo (ping) request id=0x0018, seq=11/2816, ttl=63 (no respo…
98	24.371106024	1.2.3.4	10.9.0.5	ICMP	100	Echo (ping) reply id=0x0018, seq=11/2816, ttl=64 (request …
99	24.371132955	1.2.3.4	10.9.0.5	ICMP	100	Echo (ping) reply id=0x0018, seq=11/2816, ttl=64

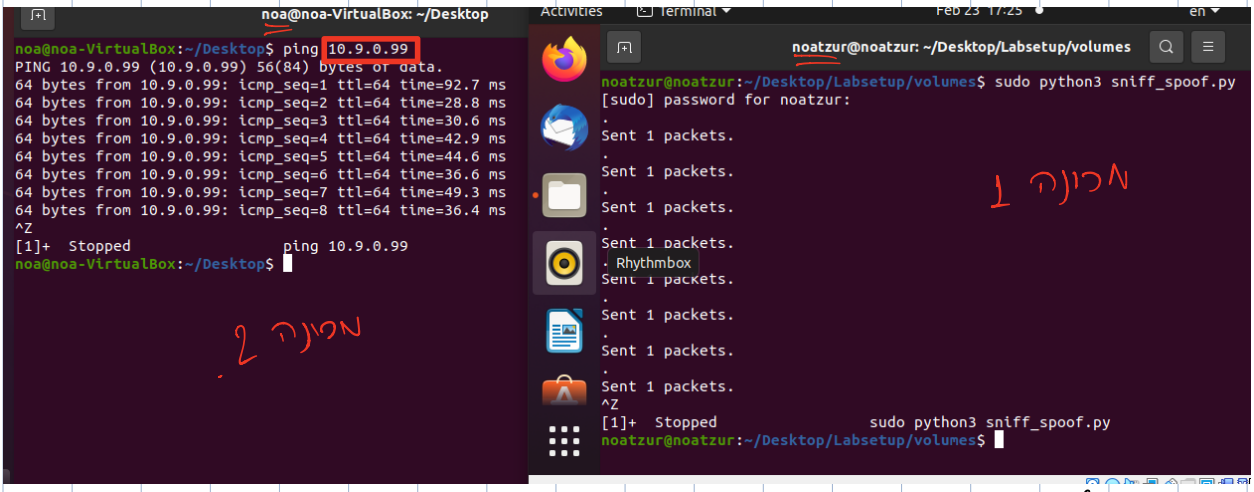
הודעת ה reply המזויפת.

• ביצוע ping ל- 10.9.0.99.

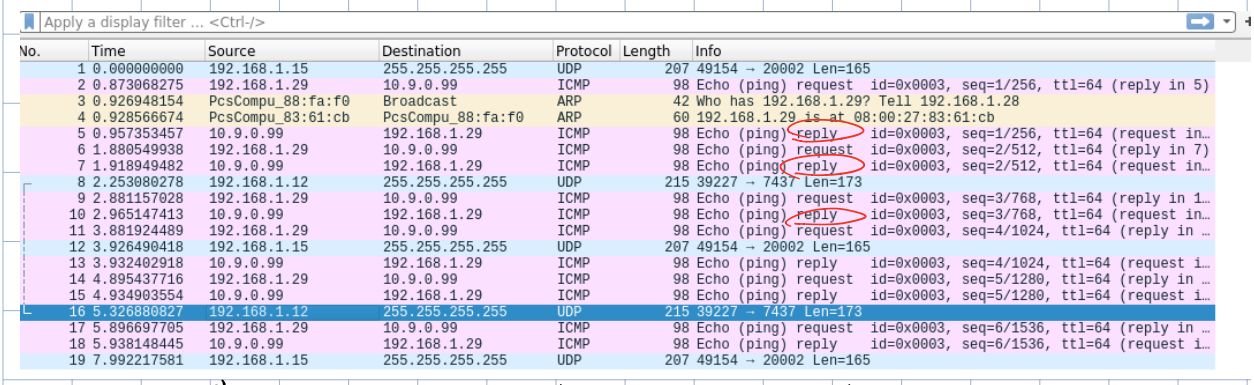
במכונה 1 הפעלנו את סקריפט ה- Sniff & Spoof שלנו.

במכונה השניה שלחנו ping ל IP המבוקש.

ניתן לראות שאכן התקבלו הודעות reply מהכתובת שזו כתובת שלא משויך אליה שום מכונה ולכן לא אמורים לקבל מענה מלכתחילה.



בנוסף - תמונה מ-WS שתוכל במקביל:



ניתן לראות שעבור כל request שנשלח ע"י ה ping התקבלה reply מה ICMP.