

אלעד ישראל 313448888 ונרקיס קרמיזי שלו 205832447

## חלק א - סעיף 1

הפעלנו את קוד השרת כולל השינויים הרצויים במכונה וירטואלית 1 והפעלת את קוד הלקוח כולל השינויים הרצויים במכונה וירטואלית 2 (תחת אותה הרשת).

להלן הפלט שקיבלנו בצד השרת -

```
elad@elad-virtualbox:~/Desktop$ python3 server.py
b'Narkis Shallev Kremizi 205832447, Elad Israel 313448888' ('10.0.2.15', 40703)
```

להלן הפלט שקיבלנו בצד הלקוח. ניתן לראות שהשרת אכן החזיר ללקוח את התשובה הנכונה (מה שהלקוח שלח, באותיות גדולות) -

```
elad@elad-virtualbox:~$ python3 client.py
b'NARKIS SHALLEV KREMIZI 205832447, ELAD ISRAEL 313448888' ('10.0.2.4', 12345)
```

## חלק א - סעיף 2

הסנפנו את התעבורה בין 2 המכונות ע"י wireshark וסיננו את החבילות שהלקוח שלח לשרת ואת החבילות שהשרת שלח ללקוח ע"י כתיבת תנאי מתאים בשורת הסינון:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.4	10.0.2.15	UDP	55	12345 → 40703
2	0.000366085	10.0.2.15	10.0.2.4	UDP	55	40703 → 12345

על מנת למצוא את כתובות הקו של השרת ושל הלקוח השתמשנו בפקודת ifconfig בטרמינל של שתי המכונות:

בשרת -

```
elad@elad-virtualbox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.4 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::d617:9621:3c46:2a93 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:6e:78:c7 txqueuelen 1000 (Ethernet)
    RX packets 44050 bytes 61522429 (61.5 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 20883 bytes 1285790 (1.2 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

בלקוח -

```
elad@elad-virtualbox:~/Desktop$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::8196:2590:3fe4:a179 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:ee:ac:9d txqueuelen 1000 (Ethernet)
    RX packets 43651 bytes 61028450 (61.0 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 21050 bytes 1296272 (1.2 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

## חלק א - סעיף 3

נביט בקוד השרת:

```
import socket

s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
s.bind(('', 12345))

while True:
    data, addr = s.recvfrom(1024)
    print(str(data), addr)
    s.sendto(data.upper(), addr)
```

בקוד זה השימוש הראשון בפורט הוא בשורה 3 בה אנחנו עושים bind (קשירה) בין socketn של השרת לבין הפורט 12345. בקשירה זו השרת מבקש ממערכת ההפעלה להקצות לו פורט מסוים ולא רנדומלי. פורט זה הוא קבוע על מנת שהלקוח יוכל לדעת לאן הוא יכול לשלוח הודעות.

בשורה 5 השרת ממטין להודעה מהלקוח וכאשר יש הודעה כזו תוכן ההודעה נכנס לdata והip והפורט של הלקוח נכנסים כtupel. לאחר מכן, השרת מדפיס את הdatan ואת addr בשורה 6.

בשורה 7 השרת שולח הודעה בחזרה ללקוח שממנו קיבל הודעה קודם לכן. תוכן ההודעה זהה לתוכן שקיבל קודם (רק שהפעם באותיות גדולות) וכדי לשלוח את ההודעה ללקוח הספציפי ששלח לו הוא משתמש בaddr.

כעת, נביט בקוד הלקוח:

```
import socket

s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
s.sendto(b'Narkis Shalev Kremizi 205832447, Elad Israel 313448888', ('10.0.2.4', 12345))
data, addr = s.recvfrom(1024)
print(str(data), addr)
s.close()
```

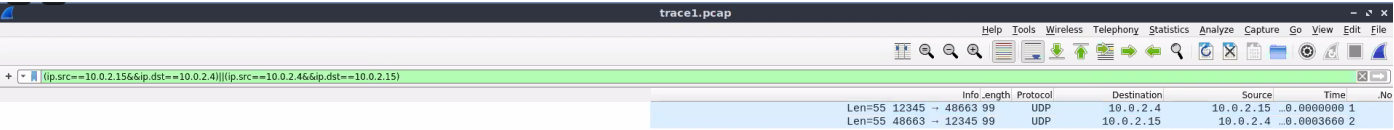
לא מצוין בקוד באופן מפורש – באיזשהו שלב מערכת ההפעלה בוחרת באופן רנדומלי פורט פנוי עבור הלקוח.

בקוד זה השימוש הראשון בפורט הוא בשורה 3 בה הלקוח שולח הודעה לשרת שהsocket שלו נמצא בפורט 12345.

בשורה 4 הלקוח ממטין להודעה חוזרת מהשרת וכאשר יש הודעה כזו תוכן ההודעה נכנס לdata והip והפורט של השרת נכנסים כtupel. לאחר מכן, הלקוח מדפיס את הdatan ואת addr בשורה 5.

בשורה 6 מתבצעת סגירה של socketn שבוללת בין היתר שחרור של מספר הפורט שהוקצה ללקוח.

לסיום, נדגים באמצעות wireshark:

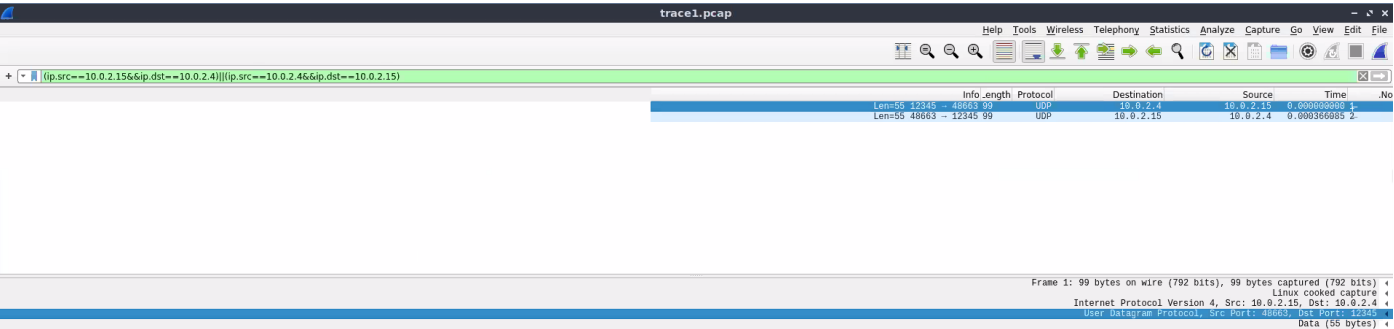


ניתן לראות שהסנפנו 2 חבילות –

הראשונה מהלקוח לשרת, ניתן לראות זאת ע"י ש ip.src שווה ל10.0.2.15 שזה ip של הלקוח וip.dst שווה ל10.0.2.4 שזה ip של השרת.

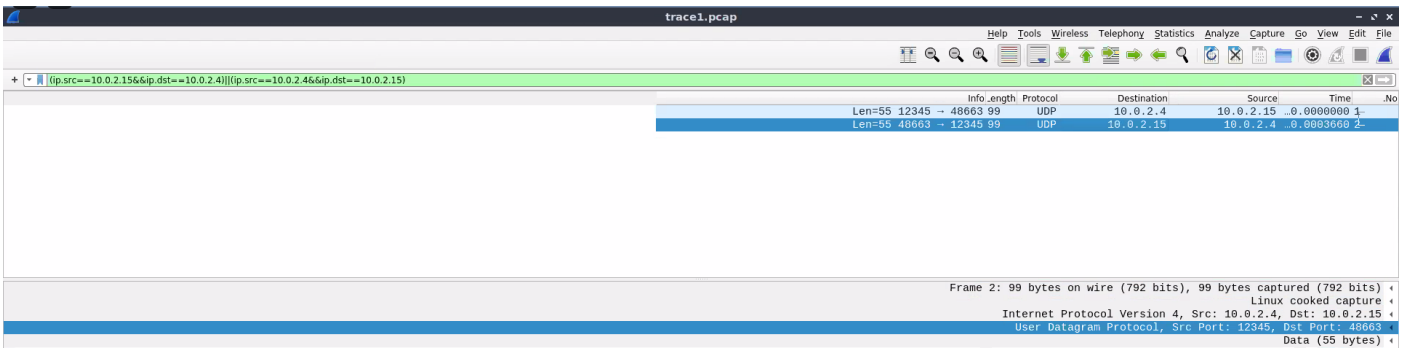
השנייה מהשרת ללקוח, ניתן לראות זאת ע"י ש ip.src שווה ל10.0.2.4 שזה ip של השרת וip.dst שווה ל10.0.2.15 שזה ip של הלקוח.

נביט בחבילה הראשונה (מהלקוח לשרת):



ניתן לראות בשכבת התעבורה המסומנת שאכן החבילה נשלחה מהפורט של הלקוח (שמערכת ההפעלה הקצתה) לפורט של השרת שקבענו להיות 12345.

נביט בחבילה השנייה (מהשרת ללקוח):



The screenshot shows a Wireshark capture of a packet named 'trace1.pcap'. The packet list pane shows a single packet, Frame 2, with a length of 55 bytes. The packet details pane shows the following structure:

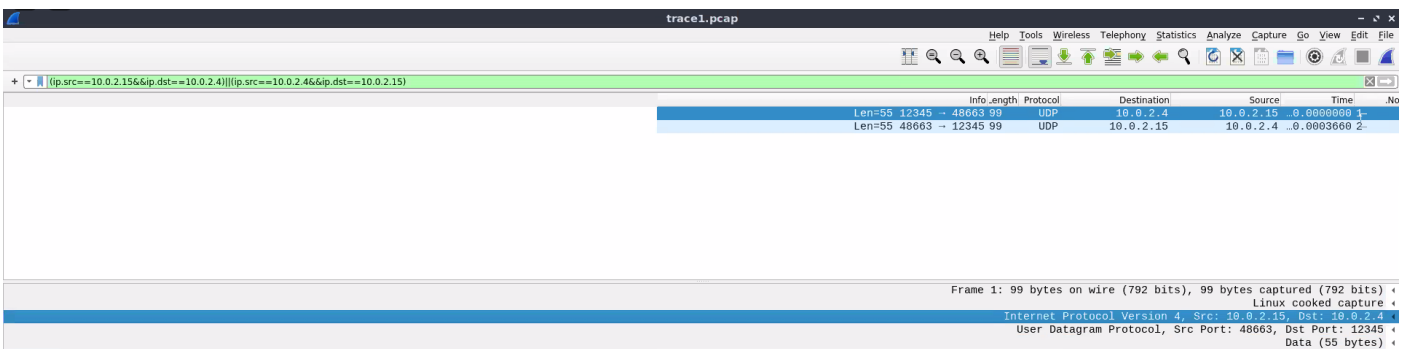
Info	Length	Source	Destination	Protocol
Ethernet II, Src: Linux cooked capture, Dst: 10.0.2.4	14	10.0.2.15	10.0.2.4	UDP
User Datagram Protocol, Src Port: 48663, Dst Port: 12345	99	10.0.2.15	10.0.2.4	UDP
Data (55 bytes)	55			

ניתן לראות בשכבת התעבורה המסומנת שאכן החבילה נשלחה מהפורט של השרת שקבענו להיות 12345 לפורט של הלקוח (שמערכת ההפעלה הקצתה).

## חלק א – סעיף 4

**\*\* הראנו שימוש בפקודה ifconfig כבר בסעיף 2.**

נביט בחבילה הראשונה (מהלקוח לשרת):

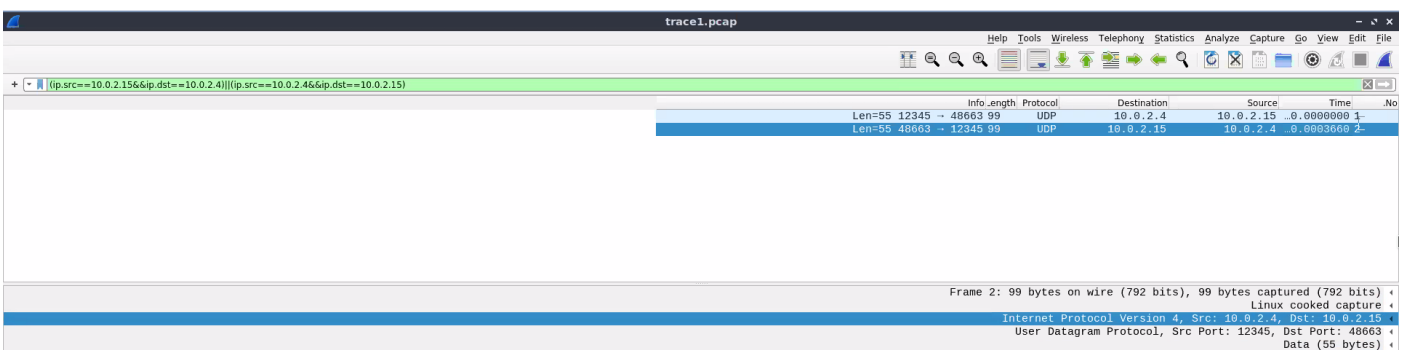


The screenshot shows a Wireshark capture of a packet named 'trace1.pcap'. The packet list pane shows a single packet, Frame 1, with a length of 55 bytes. The packet details pane shows the following structure:

Info	Length	Source	Destination	Protocol
Ethernet II, Src: Linux cooked capture, Dst: 10.0.2.15	14	10.0.2.4	10.0.2.15	UDP
User Datagram Protocol, Src Port: 48663, Dst Port: 12345	99	10.0.2.4	10.0.2.15	UDP
Data (55 bytes)	55			

ניתן לראות בשכבת הרשת המסומנת שאכן החבילה נשלחה מקו של הלקוח (הראנו בסעיף 2 שמדובר בקו שמספרו 10.0.2.15) לקו של השרת (הראנו בסעיף 2 שמדובר בקו שמספרו 10.0.2.4).

נביט בחבילה השנייה (מהשרת ללקוח):



The screenshot shows a Wireshark capture of a packet named 'trace1.pcap'. The packet list pane shows a single packet, Frame 2, with a length of 55 bytes. The packet details pane shows the following structure:

Info	Length	Source	Destination	Protocol
Ethernet II, Src: Linux cooked capture, Dst: 10.0.2.4	14	10.0.2.15	10.0.2.4	UDP
User Datagram Protocol, Src Port: 48663, Dst Port: 12345	99	10.0.2.15	10.0.2.4	UDP
Data (55 bytes)	55			

ניתן לראות בשכבת הרשת המסומנת שאכן החבילה נשלחה מקו של השרת (הראנו בסעיף 2 שמדובר בקו שמספרו 10.0.2.4) לקו של הלקוח (הראנו בסעיף 2 שמדובר בקו שמספרו 10.0.2.15).

## חלק ב –

להלן דוגמת פלט:

```
elad@elad-virtualbox:~/Desktop$ python3 client.py 10.0.2.4 12345
mail.biu.ac.il
1.2.3.5
www.biu.ac.il
1.2.3.4
mail.google.co.il
9.9.9.9
```

תפסנו את התעבורה באמצעות wireshark וקיבלנו את 8 החבילות הבאות:

	Info_length	Protocol	Destination	Source	Time	.No
Len=14	12345 → 34396 62	UDP	10.0.2.4	10.0.2.15	0.000000000	1
Len=26	34396 → 12345 70	UDP	10.0.2.15	10.0.2.4	0.000137755	2
Len=13	12345 → 34396 62	UDP	10.0.2.4	10.0.2.15	28.758868085	7
Len=25	34396 → 12345 69	UDP	10.0.2.15	10.0.2.4	28.759010534	8
Len=17	12345 → 34396 62	UDP	10.0.2.4	10.0.2.15	47.560258799	13
Len=17	55555 → 12345 61	UDP	127.0.0.1	127.0.0.1	47.560407004	14
Len=29	12345 → 55555 73	UDP	127.0.0.1	127.0.0.1	47.560481158	15
Len=29	34396 → 12345 73	UDP	10.0.2.15	10.0.2.4	47.560847449	16

2 החבילות הראשונות (1 ו-2) מתייחסות לבקשה הראשונה של הלקוח לשרת ("mail.biu.ac.il").

נפרט את המעבר של חבילה 1 בשבבות השונות:

	Info_length	Protocol	Destination	Source	Time	.No
Len=14	12345 → 34396 62	UDP	10.0.2.4	10.0.2.15	0.000000000	1
Len=26	34396 → 12345 70	UDP	10.0.2.15	10.0.2.4	0.000137755	2
Len=13	12345 → 34396 62	UDP	10.0.2.4	10.0.2.15	28.758868085	7
Len=25	34396 → 12345 69	UDP	10.0.2.15	10.0.2.4	28.759010534	8
Len=17	12345 → 34396 62	UDP	10.0.2.4	10.0.2.15	47.560258799	13
Len=17	55555 → 12345 61	UDP	127.0.0.1	127.0.0.1	47.560407004	14
Len=29	12345 → 55555 73	UDP	127.0.0.1	127.0.0.1	47.560481158	15
Len=29	34396 → 12345 73	UDP	10.0.2.15	10.0.2.4	47.560847449	16

Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface any, id 0  
 Linux cooked capture  
 Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.0.2.4  
 User Datagram Protocol, Src Port: 34396, Dst Port: 12345  
 Data (14 bytes)  
 VSS Monitoring Ethernet trailer, Source Port: 0

```

0000  00 00 00 01 00 06 08 00 27 ee ac 9d 00 00 08 00  .....
0010  45 00 00 2a e5 2a 40 00 40 11 3d 86 0a 00 02 0f  E..*.*@. @.=...
0020  0a 00 02 04 86 5c 30 39 00 16 c7 78 6d 61 69 6c  ....\09 ...xmail
0030  2e 62 69 75 2e 61 63 2e 69 6c 00 00 00 00 00  .biu.ac.il....

```

שבבת האפליקציה של הלקוח מעבירה את datan שהיא הכתובת mail.biu.ac.il בבתים (14 בתים) לשבבת התעבורה של הלקוח -

Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.0.2.4	
User Datagram Protocol, Src Port: 34396, Dst Port: 12345	
Data (14 bytes)	
Data: 6d61696c2e6269752e61632e696c	
[Length: 14]	
VSS Monitoring Ethernet trailer, Source Port: 0	

```

0000  00 00 00 01 00 06 08 00 27 ee ac 9d 00 00 08 00  .....
0010  45 00 00 2a e5 2a 40 00 40 11 3d 86 0a 00 02 0f  E..*.*@. @.=...
0020  0a 00 02 04 86 5c 30 39 00 16 c7 78 6d 61 69 6c  ....\09 ...xmail
0030  2e 62 69 75 2e 61 63 2e 69 6c 00 00 00 00 00  .biu.ac.il....

```

לאחר מכן שבבת התעבורה של הלקוח מוסיפה header שכולל פורט מקור (34396 - הפורט של הלקוח) ופורט יעד (12345 - הפורט של השרת) ומעבירה את הSegment שנוצר לשבבת הרשת של הלקוח -

Linux cooked capture	
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.0.2.4	
User Datagram Protocol, Src Port: 34396, Dst Port: 12345	
Source Port: 34396	
Destination Port: 12345	
Length: 22	
Checksum: 0xc778 [unverified]	
[Checksum Status: Unverified]	
[Stream index: 0]	
[Timestamps]	
Data (14 bytes)	
VSS Monitoring Ethernet trailer, Source Port: 0	

```

0000  00 00 00 01 00 06 08 00 27 ee ac 9d 00 00 08 00  .....
0010  45 00 00 2a e5 2a 40 00 40 11 3d 86 0a 00 02 0f  E..*.*@. @.=...
0020  0a 00 02 04 86 5c 30 39 00 16 c7 78 6d 61 69 6c  ....\09 ...xmail
0030  2e 62 69 75 2e 61 63 2e 69 6c 00 00 00 00 00  .biu.ac.il....

```

לאחר מכן שבבת הרשת של הלקוח מוסיפה header שכולל IP מקור (10.0.2.15 - IP של הלקוח) ויעד (10.0.2.4 - IP של הלקוח) ומעבירה את הpacket שנוצר לשבבת הערוץ של הלקוח -

Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface any, id 0		
Linux cooked capture		
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.0.2.4		
Version: 4 = .... 0100		
Header Length: 20 bytes (5) = 0101 ....		
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)		
Total Length: 42		
Identification: 0xe52a (58666)		
Flags: 0x4000, Don't fragment		
Fragment offset: 0		
Time to live: 64		
Protocol: UDP (17)		
Header checksum: 0x3d86 [validation disabled]		
[Header checksum status: Unverified]		
Source: 10.0.2.15		
Destination: 10.0.2.4		
User Datagram Protocol, Src Port: 34396, Dst Port: 12345		
Data (14 bytes)		
VSS Monitoring Ethernet trailer, Source Port: 0		
0000	00 00 00 01 00 06 08 00 27 ee ac 9d 00 00 08 00	.....'.....
0010	45 00 00 2a e5 2a 40 00 40 11 3d 86 0a 00 02 0f	E..*.*@. @.=.....
0020	0a 00 02 04 86 5c 30 39 00 16 c7 78 6d 61 69 6c	.....\09 ...xmail
0030	2e 62 69 75 2e 61 63 2e 69 6c 00 00 00 00	.biu.ac. il.....

ניתן לשים לב שבשכבת הרשת ישנו שימוש בפרוטוקול UDP כפי שהוגדר בסוקטים בקוד של הלקוח ובקוד של השרת.

לאחר מכן שכבת הערוץ של הלקוח מוסיפה header שכולל mac מקור (27:08:00:00:ac:ee - כתובת הmac של הלקוח) mac יעד (שאינו מופיע בפירוט מכיוון שבmode של any, תוכנת wireshark מספקת fake link-layer headers ולא את המקור) ומעבירה את frame שנוצר לשכבה הפיזית של הלקוח -

Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface any, id 0									
Linux cooked capture									
Packet type: Unicast to us (0)									
Link-layer address type: 1									
Link-layer address length: 6									
Source: PcsCompu_ee:ac:9d (08:00:27:ee:ac:9d)									
Unused: 0000									
Protocol: IPv4 (0x0800)									
Padding: 0000									
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.0.2.4									
User Datagram Protocol, Src Port: 34396, Dst Port: 12345									
Data (14 bytes)									
VSS Monitoring Ethernet trailer, Source Port: 0									
0000	00 00 00 01 00 06 08 00 27 ee ac 9d 00 00 08 00	.....'.....							
0010	45 00 00 2a e5 2a 40 00 40 11 3d 86 0a 00 02 0f	E..*.*@. @.=.....							
0020	0a 00 02 04 86 5c 30 39 00 16 c7 78 6d 61 69 6c	.....\09 ...xmail							
0030	2e 62 69 75 2e 61 63 2e 69 6c 00 00 00 00 00	.biu.ac. il.....							

השכבה הפיזית של הלקוח כוללת את כל המידע בבתיים (62 בתיים), וזהו המידע שמועבר לשכבה הפיזית של השרת -

Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface any, id 0		Linux cooked capture
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.0.2.4		User Datagram Protocol, Src Port: 34396, Dst Port: 12345
Data (14 bytes)		VSS Monitoring Ethernet trailer, Source Port: 0
0000	00 00 00 01 00 06 08 00 27 ee ac 9d 00 00 08 00	.....'.....
0010	45 00 00 2a e5 2a 40 00 40 11 3d 86 0a 00 02 0f	E..*.*@. @.=.....
0020	0a 00 02 04 86 5c 30 39 00 16 c7 78 6d 61 69 6c	.....\09 ...xmail
0030	2e 62 69 75 2e 61 63 2e 69 6c 00 00 00 00	.biu.ac. il.....

לסיום, מתבצע תהליך הפוך של העברה ופירוק המידע מהשכבה הפיזית של השרת לשכבת האפליקציה של השרת.

כעת, המעבר של חבילה 2 בשכבות השונות דומה למעבר של חבילה 1, אך ניתן לראות שהמידע (Data) המועבר מכיל את תשובת השרת ללקוח שהיא "1.2.3.5,240,mail.biu.ac.il". לאחר מכן הלקוח לוקח מתוך התשובה את כתובת הIP ומדפיס אותה.

		Info_length	Protocol	Destination	Source	Time	.No
	Len=14	12345 → 34396	62	UDP	10.0.2.4	10.0.2.15 0.000000000	1
	Len=26	34396 → 12345	70	UDP	10.0.2.15	10.0.2.4 0.000137755	2
	Len=13	12345 → 34396	62	UDP	10.0.2.4	10.0.2.15 28.758868085	7
	Len=25	34396 → 12345	69	UDP	10.0.2.15	10.0.2.4 28.759010534	8
	Len=17	12345 → 34396	62	UDP	10.0.2.4	10.0.2.15 47.560258799	13
	Len=17	55555 → 12345	61	UDP	127.0.0.1	127.0.0.1 47.560407004	14
	Len=29	12345 → 55555	73	UDP	127.0.0.1	127.0.0.1 47.560481158	15
	Len=29	34396 → 12345	73	UDP	10.0.2.15	10.0.2.4 47.560847449	16

Frame 2: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface any, id 0
Linux cooked capture
Internet Protocol Version 4, Src: 10.0.2.4, Dst: 10.0.2.15
User Datagram Protocol, Src Port: 12345, Dst Port: 34396
Data (26 bytes)

0000	00 04 00 01 00 06 08 00 27 6e 78 c7 00 00 08 00	.....'nx.....
0010	45 00 00 36 4c 5c 40 00 40 11 d6 48 0a 00 02 04	E..6L\@. @...H....
0020	0a 00 02 0f 30 39 86 5c 00 22 18 46 6d 61 69 6c	....09\ "Fmail
0030	2e 62 69 75 2e 61 63 2e 69 6c 2c 31 2e 32 2e 33	.biu.ac.il,1.2.3
0040	2e 35 2c 32 34 30	.5,240

2 החבילות הבאות (7 ו-8) מתייחסות לבקשה השנייה של הלקוח לשרת ("www.biu.ac.il"). המעבר בשכבות מתבצע בדיוק באותו אופן כפי שהסברנו למעלה, ההבדל היחיד הוא המידע שנשלח והתשובה של השרת שהיא "www.biu.ac.il,1.2.3.4,180".

4 החבילות האחרונות (13-16) מתייחסות לבקשה השלישית של הלקוח לשרת ("mail.google.co.il"). המעבר בשכבות מתבצע בדיוק באותו אופן כפי שהסברנו למעלה, ההבדלים הפעם הם שהתשובה לבקשת הלקוח לא נמצאת בקובץ של השרת הרגיל והוא צריך לפנות לשרת האב כדי לקבל את התשובה ממנו.

בחבילה 13 הלקוח פונה לשרת הרגיל עם הבקשה. ניתן לראות שהחבילה נשלחת מ-IP של הלקוח שמספרו 10.0.2.15 ל-IP של השרת הרגיל שמספרו 10.0.2.4 ושהתוכן שלה הוא התוכן של Data המסומן -

		Info_length	Protocol	Destination	Source	Time	.No
	Len=14	12345 → 34396	62	UDP	10.0.2.4	10.0.2.15 0.000000000	1
	Len=26	34396 → 12345	70	UDP	10.0.2.15	10.0.2.4 0.000137755	2
	Len=13	12345 → 34396	62	UDP	10.0.2.4	10.0.2.15 28.758868085	7
	Len=25	34396 → 12345	69	UDP	10.0.2.15	10.0.2.4 28.759010534	8
	Len=17	12345 → 34396	62	UDP	10.0.2.4	10.0.2.15 47.560258799	13
	Len=17	55555 → 12345	61	UDP	127.0.0.1	127.0.0.1 47.560407004	14
	Len=29	12345 → 55555	73	UDP	127.0.0.1	127.0.0.1 47.560481158	15
	Len=29	34396 → 12345	73	UDP	10.0.2.15	10.0.2.4 47.560847449	16

Data (17 bytes)
Data: 6d61696c2e676f6f676c652e636f2e696c
[Length: 17]

0000	00 00 00 01 00 06 08 00 27 ee ac 9d 00 00 08 00	.....'.....
0010	45 00 00 2d f4 d5 40 00 40 11 2d d8 0a 00 02 0f	E....@. @.....
0020	0a 00 02 04 86 5c 30 39 00 19 f1 fb 6d 61 69 6c	....\09 ....mail
0030	2e 67 6f 6f 67 6c 65 2e 63 6f 2e 69 6c 00	.google.co.il

בחבילה 14 השרת הרגיל פונה לשרת האב עם הבקשה. במקרה זה מכיוון שהתקשורת נעשית באותה מכוונה, האב והבן מתקשרים ביניהם על localhost, כלומר לכתובת 127.0.0.1. ניתן לראות שהחבילה נשלחת מה-port של השרת הרגיל שמספרו 12345 ל-port של שרת האב שמספרו 55555 ושהתוכן שלה הוא התוכן של Data המסומן -

		Info_length	Protocol	Destination	Source	Time	.No
	Len=14	12345 → 34396	62	UDP	10.0.2.4	10.0.2.15 0.000000000	1
	Len=26	34396 → 12345	70	UDP	10.0.2.15	10.0.2.4 0.000137755	2
	Len=13	12345 → 34396	62	UDP	10.0.2.4	10.0.2.15 28.758868085	7
	Len=25	34396 → 12345	69	UDP	10.0.2.15	10.0.2.4 28.759010534	8
	Len=17	12345 → 34396	62	UDP	10.0.2.4	10.0.2.15 47.560258799	13
	Len=17	55555 → 12345	61	UDP	127.0.0.1	127.0.0.1 47.560407004	14
	Len=29	12345 → 55555	73	UDP	127.0.0.1	127.0.0.1 47.560481158	15
	Len=29	34396 → 12345	73	UDP	10.0.2.15	10.0.2.4 47.560847449	16

Data (17 bytes)
Data: 6d61696c2e676f6f676c652e636f2e696c
[Length: 17]

0000	00 00 03 04 00 06 00 00 00 00 00 00 00 08 00	.....+@.....
0010	45 00 00 2d 2b b4 40 00 40 11 11 0a 7f 00 00 01	E...+@. @.....
0020	7f 00 00 01 30 39 d9 03 00 19 fe 2c 6d 61 69 6c	....09...mail
0030	2e 67 6f 6f 67 6c 65 2e 63 6f 2e 69 6c	.google.co.il

בחבילה 15 שרת האב מחזיר את התשובה "9.9.9.9,240,mail.google.co.il" לשרת הרגיל. ניתן לראות שהחבילה נשלחת מה-port של שרת האב שמספרו 55555 ל-port של השרת הרגיל שמספרו 12345 ושהתוכן שלה הוא התוכן של Data המסומן -



		Info_length	Protocol	Destination	Source	Time	.No
	Len=14	12345 → 34396 62	UDP	10.0.2.4	10.0.2.15	0.000000000	1
	Len=26	34396 → 12345 70	UDP	10.0.2.15	10.0.2.4	0.000137755	2
	Len=13	12345 → 34396 62	UDP	10.0.2.4	10.0.2.15	28.758868085	7
	Len=25	34396 → 12345 69	UDP	10.0.2.15	10.0.2.4	28.759010534	8
	Len=17	12345 → 34396 62	UDP	10.0.2.4	10.0.2.15	47.560258799	13
	Len=17	55555 → 12345 61	UDP	127.0.0.1	127.0.0.1	47.560407004	14
	Len=29	12345 → 55555 73	UDP	127.0.0.1	127.0.0.1	47.560481158	15
	Len=29	34396 → 12345 73	UDP	10.0.2.15	10.0.2.4	47.560847449	16

		Data (29 bytes)
		...Data: 6d61696c2e676f6f676c652e636f2e696c2c392e392e392e
		[Length: 29]
0000	00 00 03 04 00 06 00 00 00 00 00 00 00 08 00	.....
0010	45 00 00 39 2b b5 40 00 40 11 10 fd 7f 00 00 01	E..9+.@. @.....
0020	7f 00 00 01 d9 03 30 39 00 25 fe 38 6d 61 69 6c	.....09.%8mail
0030	2e 67 6f 6f 67 6c 65 2e 63 6f 2e 69 6c 2c 39 2e	.google.co.il,9.
0040	39 2e 39 2e 39 2c 32 34 30	9.9.9,24 0

בחבילה 16 השרת הרגיל מחזיר את התשובה ללקוח. ניתן לראות שהחבילה נשלחת מIP של השרת הרגיל שמספרו 10.0.2.4 לIP של הלקוח שמספרו 10.0.2.15 ושהתוכן שלה הוא התוכן של Data המסומן -

		Info_length	Protocol	Destination	Source	Time	.No
	Len=14	12345 → 34396 62	UDP	10.0.2.4	10.0.2.15	0.000000000	1
	Len=26	34396 → 12345 70	UDP	10.0.2.15	10.0.2.4	0.000137755	2
	Len=13	12345 → 34396 62	UDP	10.0.2.4	10.0.2.15	28.758868085	7
	Len=25	34396 → 12345 69	UDP	10.0.2.15	10.0.2.4	28.759010534	8
	Len=17	12345 → 34396 62	UDP	10.0.2.4	10.0.2.15	47.560258799	13
	Len=17	55555 → 12345 61	UDP	127.0.0.1	127.0.0.1	47.560407004	14
	Len=29	12345 → 55555 73	UDP	127.0.0.1	127.0.0.1	47.560481158	15
	Len=29	34396 → 12345 73	UDP	10.0.2.15	10.0.2.4	47.560847449	16

		Data (29 bytes)
		...Data: 6d61696c2e676f6f676c652e636f2e696c2c392e392e392e
		[Length: 29]
0000	00 04 00 01 00 06 08 00 27 6e 78 c7 00 00 08 00	.....'nx.....
0010	45 00 00 39 6c 27 40 00 40 11 b6 7a 0a 00 02 04	E..01'@. @..z.....
0020	0a 00 02 0f 30 39 86 5c 00 25 18 49 6d 61 69 6c	....09.\.%Imail
0030	2e 67 6f 6f 67 6c 65 2e 63 6f 2e 69 6c 2c 39 2e	.google.co.il,9.
0040	39 2e 39 2e 39 2c 32 34 30	9.9.9,24 0