בס"ד

דוח 3 רשתות- שכבת האפליקציה

חלק 1- פתיחת שרת אינטרנט-

שלב זה בעבודה התבקשנו לפתוח שרת אינטרנט. הורדנו שרת apache על המחשב שלנו בעזרת שורת הפקודה - sudo apt install apache2 אשר קיבלנו בעבודה. לאחר מכן ביצענו שינויי קל בשרת אשר קיבלנו כך שכאשר נפתח את השרת נראה את הדף האוטומטי שקיבלנו ובנוסף כיתוב שאנו הוספנו ביצענו את השינוי בכך שנכנסנו לקובץ אשר שומר את המידע על הדף שבשרת שהורדנו אשר נמצא בנתיב /var/www/html

עתה בשלב הראשון נטען את השרת שלנו במחשב עליו הוא מותקן בכך שבגוגל נכנס לipl המסמל localhost (127.0.0.1) וכך פתחנו את האתר הבא-

תוך כדי הפתיחה הפעלנו את תוכנת הוויר שארק אשר תסניף את החבילות שהועברו בעת הפתיחה

Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM TSval=1750889445 TSecr=0 WS=128 [SYN] 80 → 60342 74	TCP	127.0.0.1	127.0.0.1	0.000000000	1
Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM TSval=1750889445 TSecr=1750889445 WS=128 [SYN, ACK] 60342 → 80 74	TCP	127.0.0.1	127.0.0.1	0.000012477	2
Seq=1 Ack=1 Win=65536 Len=0 TSval=1750889445 TSecr=1750889445 [ACK] 80 → 60342 66	TCP	127.0.0.1	127.0.0.1	0.000021625	3
GET / HTTP/1.1 497	HTTP	127.0.0.1	127.0.0.1	0.854296630	4
Seq=1 Ack=432 Win=65152 Len=0 TSval=1750890299 TSecr=1750890299 [ACK] 60342 → 80 66	TCP	127.0.0.1	127.0.0.1	0.854319954	5
HTTP/1.1 200 OK (text/html) 3543	HTTP	127.0.0.1	127.0.0.1	0.855082173	6
Seq=432 Ack=3478 Win=62080 Len=0 TSval=1750890300 TSecr=1750890300 [ACK] 80 → 60342 66	TCP	127.0.0.1	127.0.0.1	0.855244581	7
GET /icons/ubuntu-logo.png HTTP/1.1 439	HTTP	127.0.0.1	127.0.0.1	1.086573585	8
HTTP/1.1 200 OK (PNG) 3673	HTTP	127.0.0.1	127.0.0.1	1.086728848	9
Seq=805 Ack=7085 Win=63104 Len=0 TSval=1750890531 TSecr=1750890531 [ACK] 80 → 60342 66	TCP	127.0.0.1	127.0.0.1	1.086734677	10
GET /favicon.ico HTTP/1.1 429	HTTP	127.0.0.1	127.0.0.1	1.117642544	11
HTTP/1.1 404 Not Found (text/html) 553	HTTP	127.0.0.1	127.0.0.1	1.117854926	12
Seq=1168 Ack=7572 Win=65536 Len=0 TSval=1750890613 TSecr=1750890562 [ACK] 80 → 60342 66	TCP	127.0.0.1	127.0.0.1	1.168305323	13
GET / HTTP/1.1 589	HTTP	127.0.0.1	127.0.0.1	3.344187642	14
HTTP/1.1 200 OK (text/html) 3542	HTTP	127.0.0.1	127.0.0.1	3.345543404	15
Seq=1691 Ack=11048 Win=63232 Len=0 TSval=1750892790 TSecr=1750892790 [ACK] 80 → 60342 66	TCP	127.0.0.1	127.0.0.1	3.345746951	16
Seq=1691 Ack=11048 Win=65536 Len=0 TSval=1750895482 TSecr=1750892790 [FIN, ACK] 80 → 60342 66	TCP	127.0.0.1	127.0.0.1	6.037812797	17
Seq=11048 Ack=1692 Win=65536 Len=0 TSval=1750895483 TSecr=1750895482 [FIN, ACK] 60342 → 80 66	TCP	127.0.0.1	127.0.0.1	6.038221911	18
Seq=1692 Ack=11049 Win=65536 Len=0 TSval=1750895484 TSecr=1750895483 [ACK] 80 → 60342 66	TCP	127.0.0.1	127.0.0.1	6.038970373	19

בס"ד

עתה נסביר על התעבורה שקרתה בעת פתיחת האתר.

תעבורה זאת תבוצע בעזרת פרוטוקל tpc בשכבת התעבורה ובפרוטוקול http בשכבת האפליקצייה. פרוטוקול tpc הוא פרוטוקול בשכבת התעבורה המשמש להעברת מידע. הוא נחשב לפרוטוקול אמין ביחס למתחרים בשל המנגנונים שבו הבודקים שמידע הועבר כראוי. פרוטוקול http הוא פרוטוקול בשכבת האפליקצייה אשר רץ על גבי פרוטוקול tcp בשכבת התעבורה (הוא מסתמך על אמינותו של tcp). פרוטוקול זה מאפשר לדפדפן להוריד מידע משרת אינטרנט ולהציג אותו בצורה ויזואלית (המידע מועבר בhtml)

ראשית ניתן לראות שכל החבילות נשלחו לאותו מקום כלומר localhost וזאת משום שפתחנו את האתר באותו מחשב שלנו. בעזרת הוויר שארק ניתן לראות שכל החבילות http הועברו על גבי אותו חיבור ולכן עבדנו על http בגרסה 1.1 (גם נראה בחבילות שכתוב שזו הייתה הגרסא שלנו).

ראשית מתקיים פתיחת חיבור tcp כפי שניתן לראות ב3 החבילות הראשונות בתהליך לחיצת הידיים המשולשת. (אין צורך להסביר מעבר משום שלא התבקשנו על כך בעבודה זו).

לאחר שנפתח חיבור ה tcp נשלח על גביו בקשת http אשר מבקשת את המידע של השרת apache שהורדנו ושינינו בהתאם על מנת לפתוח את הדף בדפדפן. הבקשה נמצאת בחבילה מספר 4 ונראית כך-

```
GET / HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:108.0) Gecko/20100101 Firefox/108.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
```

בבקשה זאת נפתח לנו חיבור HTTP אשר הוא מסוג get (כלומר אנו לא יכולים לשנות את המידע שישלח אלינו אלא רק לצפות בו) בבקשה זו אנו מבקשים מהשרת לקבל את דף האינטרנט השמור בשרת (בצורת html).
בנוסף ניתן לראות שעבדנו על http מסוג 1.1 כפי שכתוב בבקשה ומה שאנו מבקשים הוא את העמוד הדיפולטיבי שהגדרנו בשרת הapach ואת זאת אנו יודעים משום שהקובץ המבוקש הוא / אשר מייצג את עמוד הבית.

בחבילה מספר 5 אנו מקבלים חבילת אישור מן השרת אל הדפדפן שפותח את האתר. זוהי חבילת tcp עם דגל ack אשר אומרת שהשרת קיבל את בקשת הhttp של הדפדפן ונפתח לנו חיבור http.

```
Frame 5: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface lo, id 0
Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 80, Dst Port: 60342, Seq: 1, Ack: 432, Len: 0

0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ........ ......E·
0010  00 34 66 9d 40 00 40 06  d6 24 7f 00 00 01 7f 00   ·4f·@·@· ·$......
0020  00 01 00 50 eb b6 db 4e  a0 78 95 8b 2b a5 80 10   ···P···N ·x··+···
0030  01 fd fe 28 00 00 01 01  08 0a 68 5c 77 3b 68 5c   ···(···· ··h\w;h\
0040  77 3b                                              w;
```

בחבילה 6 אנו מקבלים את התשובה מן השרת והיא נראית כך-

```
HTTP/1.1 200 OK
Date: Wed, 11 Jan 2023 10:48:21 GMT
Server: Apache/2.4.52 (Ubuntu)
Last-Modified: Mon, 09 Jan 2023 13:44:14 GMT
ETag: "29c5-5f1d4f6861be5-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 3138
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html
```

כך נראה ה hedder של התשובה של השרת. ניתן לראות לפי ה ok שהבקשה הוצלחה משום שקיבלנו את המספר 200 אשר אומר שהתהליך עבר בהצלחה וכל המידע נמצא והועבר כראוי. המידע שמועבר לנו בחבילה זו הוא הרקע של העמוד והכיתוב בו אך התמונות אשר בעמוד יועברו בחבילות הבאות (ניתן לראות זאת בשל המידע המועבר ב html לאחר הhedder).

בחבילה מספר 7 אנו מקבלים חבילת אישור מן הדפדפן שמאשרת שהוא קיבל את המידע מן השרת. חבילה זו היא בעלת דגל ack דולק וניתן לראות שה seq number וה ack number השתנו בהתאם למידע שהועבר.

```
Frame 7: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface lo, id 0
Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 60342, Dst Port: 80, Seq: 432, Ack: 3478, Len: 0
```

```
0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ........ ......E.
0010  00 34 78 b1 40 00 40 06  c4 10 7f 00 00 01 7f 00   .4x.@.@. ........
0020  00 01 eb b6 00 50 95 8b  2b a5 db 4e ae 0d 80 10   .....P.. +..N....
0030  01 e5 fe 28 00 00 01 01  08 0a 68 5c 77 3c 68 5c   ...(.... ..h\w<h\
0040  77 3c                                              w<
```

עתה בחבילה מספר 8 הלקוח שולח לשרת בקשה לקבל תמונה כלשהי אשר אמורה להיות בדף האינטרנט הבקשה נראית כך-

```
GET /icons/ubuntu-logo.png HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:108.0) Gecko/20100101 Firefox/108.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://localhost/
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
```

ניתן לראות בבקשה זו שאנו מבקשים בקשת get בה אנו רוצים לקבל תמונה המייצגת את הלוגו של אובונטו אשר ידוע לנו ששמורה בתיקיית icons שבשרת שלנו. דף האינטרנט משתמש בתמונה זו ולכן מבקש אותה מן השרת.

בחבילה מספר 9 אנו רואים את החבילה בה השרת שלח אל הלקוח את התמונה המבוקשת-

```
HTTP/1.1 200 OK
Date: Wed, 11 Jan 2023 10:48:21 GMT
Server: Apache/2.4.52 (Ubuntu)
Last-Modified: Fri, 30 Sep 2022 04:09:50 GMT
ETag: "cfa-5e9dd2a489f80"
Accept-Ranges: bytes
Content-Length: 3322
Keep-Alive: timeout=5, max=99
Connection: Keep-Alive
Content-Type: image/png
```

כך נראה קובץ ה hedder של חבילת האישור (לאחר מכן יש מידע המייצג את התמונה) ניתן להבין מכך שהבקשה אושרה ושהעברנו את התמונה המבוקשת כראוי (קיבלנו 200 שהוא מייצג שהמידע הועבר בהצלחה).

בס"ד

בחבילה מספר 10 אנו מקבלים חבילת אישור מן הדפדפן על כך שהוא קיבל את החבילה עם התמונה בהצלחה.
חבילה זו היא חבילת tcp עם דגל ack דלוק-

```
Frame 10: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface lo, id 0
Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 60342, Dst Port: 80, Seq: 805, Ack: 7085, Len: 0

0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ........ ......E·
0010  00 34 78 b3 40 00 40 06  c4 0e 7f 00 00 01 7f 00   ·4x·@·@· ........
0020  00 01 eb b6 00 50 95 8b  2d 1a db 4e bc 24 80 10   ·····P·· -··N·$··
0030  01 ed fe 28 00 00 01 01  08 0a 68 5c 78 23 68 5c   ···(···· ··h\x#h\
0040  78 23                                              x#
```

עתה בחבילה מספר 11 ניתן לראות שהלקוח מבקש מן השרת קובץ נוסף עבור טעינת דף האינטרנט החבילה נראית כך-

```
....IEND.B`.GET /favicon.ico HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:108.0) Gecko/20100101 Firefox/108.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://localhost/
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
```

ניתן לראות בחבילה בקשת get עבור הקובץ favicon.ico אותו מבקש הלקוח מן השרת.

בחבילה מספר 12 אנו מקבלים את תשובתו של השרת-

```
HTTP/1.1 404 Not Found
Date: Wed, 11 Jan 2023 10:48:21 GMT
Server: Apache/2.4.52 (Ubuntu)
Content-Length: 271
Keep-Alive: timeout=5, max=98
Connection: Keep-Alive
Content-Type: text/html; charset=iso-8859-1
```

ניתן לראות בחבילת האישור מן השרת שהפעם בשונה מן החבילות הקודמות השרת לא מצא את הקובץ המבוקש ולכן קיבלנו הודעה עם מספר 404 אשר מסמן לנו שהשרת לא מצא את הקובץ המבוקש.

בס"ד

בחבילה מספר 13 הדפדפן שולח חבילת אישור על שקיבל את חבילת הhttp מן השרת (החבילה שאומרת שהקובץ לא נמצא)-

```
Frame 13: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface lo, id 0
Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 60342, Dst Port: 80, Seq: 1168, Ack: 7572, Len: 0
```

```
0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ........ ......E·
0010  00 34 78 b5 40 00 40 06  c4 0c 7f 00 00 01 7f 00   ·4x·@·@· ........
0020  00 01 eb b6 00 50 95 8b  2e 85 db 4e be 0b 80 10   ·····P·· ···N····
0030  02 00 fe 28 00 00 01 01  08 0a 68 5c 78 75 68 5c   ···(···· ··h\xuh\
0040  78 42                                              xB
```

עתה בשלב זה אנו שינינו מעט את הקובץ בו שמור המידע על השרת והוספנו כיתוב משלנו ועתה החלטנו לטעון מחדש את הדפדפן מה שהתקבל זה שבחבילה הבאה שהיא חבילה מספר 14 הדפדפן ביקש שוב מן השרת את אתר הבית-

```
GET / HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:108.0) Gecko/20100101 Firefox/108.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
If-Modified-Since: Mon, 09 Jan 2023 13:44:14 GMT
If-None-Match: "29c5-5f1d4f6861be5-gzip"
```

ניתן לראות שביקשנו בקשת get לאתר הבית השמור לנו.

בחבילה מספר 15 הלקוח מקבל את המידע מן השרת וכך טוען את המידע מחדש יחד עם השינויים ראש החבילה נראה כך-

```
HTTP/1.1 200 OK
Date: Wed, 11 Jan 2023 10:48:23 GMT
Server: Apache/2.4.52 (Ubuntu)
Last-Modified: Mon, 09 Jan 2023 13:44:14 GMT
ETag: "29c5-5f1d4f6861be5-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 3138
Keep-Alive: timeout=5, max=97
Connection: Keep-Alive
Content-Type: text/html
```

בס"ד

ניתן לראות בחבילת האישור שכל המידע הועבר כראוי (קיבלנו 200) בגוף החבילה יש לנו את המידע על האתר אשר מתואר בשפת html את התמונות שביקשנו בעבר אין צורך לבקש שוב משום שהם שמורות בcach של הדפדפן לכן אין צורך לפנות שוב לשרת ולבקשם.

בחבילה מספר 16 אנו נקבל חבילת אישור מן הדפדפן על כך שקיבלנו את המידע על טעינת האתר מחדש בחבילה הקודמת. החבילה היא חבילת tcp עם דגל ack דולק-

```
Frame 16: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface lo, id 0
Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 60342, Dst Port: 80, Seq: 1691, Ack: 11048, Len: 0
```

```
0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ........ ......E·
0010  00 34 78 b7 40 00 40 06  c4 0a 7f 00 00 01 7f 00   ·4x·@·@· ........
0020  00 01 eb b6 00 50 95 8b  30 90 db 4e cb 9f 80 10   ·····P·· 0··N····
0030  01 ee fe 28 00 00 01 01  08 0a 68 5c 80 f6 68 5c   ···(···· ··h\··h\
0040  80 f6                                              ··
```

עתה מה שקורה בשלב הבא הוא שאנו נסגור את הדפדפן וכך נסגור את חיבור ה tcp שפתחנו באותה דרך בה אנו סוגרים אותו תמיד.

ניתן לראות שראשית הדפדפן שולח לשרת הודעת fin ack המבשרת על סגירתו הסופית. ולאחר מכן השרת שולח הודעה כזו משלו הסוגרת את החיבור ולסיום הדפדפן שולח הודעת ack והחיבור נסגר לצמיתות. סיום החיבור נמצא בחבילות מספר 17-19.

בתעבורה שלנו בחלק זה התבצע חיבור http אחד ויחיד ניתן לדעת זאת לפי העובדה שידוע לנו שאנו עובדים על http 1.1 שהוא יכול להעביר מספר חבילות על חיבור אחד.

עתה בשלב הבא התבקשנו לבטל את מנגנון ה- segmentation offload בעזרת הפקודה אשר ניתנה לנו –

ethtool -K [interface] tx off sg off tso off כאשר [interface] הוא כרטיס הרשת עליו אנו עובדים esp03. מה שביטול מנגנון זה בעצם עושה הוא גורם לכך שכאשר חבילות הhttp הנשלחות ממחשב למחשב מתחלקות לכמה חבילות שונות אנו נוכל לראות זאת בתעבור. משום שבמצב הרגיל תהליך זה מתבצע על כרטיס הרשת עליו אנו עובדים וכך אנו לא יכולים לראות זאת בתעבורה הנקלטת בוויר שארק. לאחר הביטול נוכל לראות במפורש את פיצול החבילות.

```
ofek@ofek-VirtualBox:~$ sudo ethtool -K enp0s3 tx off sg off tso off
[sudo] password for ofek:
Actual changes:
tx-scatter-gather: off
tx-checksum-ip-generic: off
tx-generic-segmentation: off [not requested]
tx-tcp-segmentation: off
ofek@ofek-VirtualBox:~$
```

עתה נפתח את הדפדפן ממחשב חדש (אנו נשתמש במכונה וירטואלית נוספת) נפתח דפדפן ובו אנו נכתוב את כתובת הip של המחשב המקורי וכך בעצם נתחבר לשרת ה apache שהגדרנו מקודם במחשב הראשי ונפתח את העמוד בדפדפן שלנו במחשב החדש. עתה ננתח את התעבורה שהתקבלה בוויר שארק-

	Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3192731223 TSecr=0 WS=128 [SYN]	80 → 58992	74	TCP	10.0.2.4	10.0.2.5	0.000000000 1
Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=138094545 TSecr=3192731223 WS=128 [SYN, ACK]		58992 → 80	74	TCP	10.0.2.5	10.0.2.4	0.000095336 2
	Seq=1 Ack=1 Win=64256 Len=0 TSval=3192731224 TSecr=138094545 [ACK]	80 → 58992	66	TCP	10.0.2.4	10.0.2.5	0.000471913 3
	GET / HTTP/1.1		406	HTTP	10.0.2.4	10.0.2.5	0.001035719 4
	Seq=1 Ack=341 Win=64896 Len=0 TSval=138094546 TSecr=3192731224 [ACK]	58992 → 80	66	TCP	10.0.2.5	10.0.2.4	0.001074816 5
	HTTP/1.1 200 OK (text/html)		1514	HTTP	10.0.2.5	10.0.2.4	0.001983362 6
	Continuation		1514	HTTP	10.0.2.5	10.0.2.4	0.001986114 7
	Continuation		647	HTTP	10.0.2.5	10.0.2.4	0.001986583 8
	Seq=341 Ack=3478 Win=62592 Len=0 TSval=3192731226 TSecr=138094547 [ACK]	80 → 58992	66	TCP	10.0.2.4	10.0.2.5	0.002508576 9
	GET /icons/ubuntu-logo.png HTTP/1.1		364	HTTP	10.0.2.4	10.0.2.5	0.155303983 10
	HTTP/1.1 200 OK (PNG)[Malformed Packet]		1514	HTTP	10.0.2.5	10.0.2.4	0.155677590 11
	Continuation		1514	HTTP	10.0.2.5	10.0.2.4	0.155682287 12
	Continuation		777	HTTP	10.0.2.5	10.0.2.4	0.155682586 13
	Seq=639 Ack=7085 Win=62592 Len=0 TSval=3192731380 TSecr=138094701 [ACK]	80 → 58992	66	TCP	10.0.2.4	10.0.2.5	0.156321973 14
	GET /favicon.ico HTTP/1.1		354	HTTP	10.0.2.4	10.0.2.5	0.168510374 15
	HTTP/1.1 404 Not Found (text/html)		552	HTTP	10.0.2.5	10.0.2.4	0.169391968 16
	Seq=927 Ack=7571 Win=64128 Len=0 TSval=3192731394 TSecr=138094715 [ACK]	80 → 58992	66	TCP	10.0.2.4	10.0.2.5	0.170634321 17

ראשית נוצר חיבור tcp בין שני המחשבים שלנו. המחשב השני בו ניסינו לפתוח את השרת בדפדפן שלו הוא בעל ה ip 10.0.2.5 והוא פותח חיבור tcp עם המחשב בו אנו שמרנו את השרת בעל ה ip 10.0.2.4 פתיחה זו קוראת ב3 החבילות הראשונות בתהליך לחיצת הידיים המשולשת (syn synack ack) עליו הסברנו רבות בעבודות קודמות.

עתה לאחר שנפתח החיבור מה שקורה בחבילה מספר 4 הוא שהלקוח שלנו אשר הוא המחשב המקורי עם ה ip 10.0.2.5 שולח בקשת http אל השרת שלנו שנמצא ב ip 10.0.2.4 החבילה נראית כך-

```
GET / HTTP/1.1
Host: 10.0.2.4
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:103.0) Gecko/20100101 Firefox/103.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
```

ניתן לראות בבקשת http זו שביקשנו / כלומר את דף ברירת המחדל ובנוסף ביקשנו בעזרת get כלומר אנו לא יכולים לשנות את המידע שיתקבל. ידוע לנו שגודל קובץ הheader של בקשה הוא דינאמי ויכול להשתנות בהתאם לבקשה.

בס"ד

בחבילה מספר 5 אנו מקבלים חבילת אישור מן השרת אל הדפדפן המאשרת את פתיחת חיבור ה http שנוצר.
זוהי חבילת tcp בעלת דגל ack דלוק-

```
Frame 5: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface enp0s3, id 0
Ethernet II, Src: PcsCompu_16:89:3a (08:00:27:16:89:3a), Dst: PcsCompu_f5:94:fc (08:00:27:f5:94:fc)
Internet Protocol Version 4, Src: 10.0.2.4, Dst: 10.0.2.5
Transmission Control Protocol, Src Port: 80, Dst Port: 58992, Seq: 1, Ack: 341, Len: 0
```

```
0000  08 00 27 f5 94 fc 08 00  27 16 89 3a 08 00 45 00   ··'····· '··:··E·
0010  00 34 c7 a8 40 00 40 06  5b 13 0a 00 02 04 0a 00   ·4··@·@· [······ 
0020  02 05 00 50 e6 70 4b c7  c7 ee 41 03 55 4d 80 10   ···P·pK· ··A·UM··
0030  01 fb a7 3f 00 00 01 01  08 0a 08 3b 27 d2 be 4d   ···?···· ···;'··M
0040  36 58                                              6X
```

עתה בחבילה מספר 6 אנו נקבל את חבילת התשובה על הבקשה, ראש החבילה יראה כך-

```
HTTP/1.1 200 OK
Date: Mon, 09 Jan 2023 18:03:43 GMT
Server: Apache/2.4.52 (Ubuntu)
Last-Modified: Mon, 09 Jan 2023 13:44:14 GMT
ETag: "29c5-5f1d4f6861be5-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 3138
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html
```

קיבלנו מספר אישור 200 אשר אומר שהמידע הועבר בהצלחה ובגוף החבילה מועבר המידע על האתר בשפת html אשר הגיע מן השרת שלנו. המידע המועבר על ידי השרת גדול מאוד ולכן הוא מפוצל ל3 חבילות לפי ה mms החל מחבילה 6 הראשונה עד לחבילה 8 בהם הועבר כל המידע. (אנו יכולים לראות זאת בוויר שארק בזכות העבודה שביטלנו את מנגנון segmentation offload). כל חבילה מחולקת לפי הגודל שיכול להכנס ב mms ואז מוסיפים לה header מתאים משום שהheader הוא תלוי בגודל ההודעה.

HTTP/1.1 200 OK (text/html)	1514	HTTP	10.0.2.5	10.0.2.4	0.001983362 6
Continuation	1514	HTTP	10.0.2.5	10.0.2.4	0.001986114 7
Continuation	647	HTTP	10.0.2.5	10.0.2.4	0.001986583 8

בס"ד

בחבילה מספר 9 הדפדפן שולח אל השרת חבילת אישור המבשרת שהוא קיבל את המידע שהשרת שלח לו בהצלחה. זוהי חבילת tcp עם דגל ack דלוק והיא מבשרת שהלקוח קיבל את המידע הנצרך מהשרת-

```
Frame 9: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface enp0s3, id 0
Ethernet II, Src: PcsCompu_f5:94:fc (08:00:27:f5:94:fc), Dst: PcsCompu_16:89:3a (08:00:27:16:89:3a)
Internet Protocol Version 4, Src: 10.0.2.5, Dst: 10.0.2.4
Transmission Control Protocol, Src Port: 58992, Dst Port: 80, Seq: 341, Ack: 3478, Len: 0
```

```
0000  08 00 27 16 89 3a 08 00  27 f5 94 fc 08 00 45 00   ··'··:·· '·····E·
0010  00 34 69 b7 40 00 40 06  b9 04 0a 00 02 05 0a 00   ·4i·@·@· ........
0020  02 04 e6 70 00 50 41 03  55 4d 4b c7 d5 83 80 10   ···p·PA· UMK·····
0030  01 e9 99 b9 00 00 01 01  08 0a be 4d 36 5a 08 3b   ........ ···M6Z·;
0040  27 d3                                              '·
```

לאחר מכן בחבילה מספר 10 המחשב השני מבקש מן השרת תמונה של הסמל של אובונטו על מנת לפתוח את האתר כראוי. הבקשה נראית כך-

```
R..foa.my.?.....)..GET /icons/ubuntu-logo.png HTTP/1.1
Host: 10.0.2.4
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:103.0) Gecko/20100101 Firefox/103.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
Referer: http://10.0.2.4/
```

ניתן לראות בבקשה זו שהלקוח מבקש מן השרת שנמצא ב 10.0.2.4 תמונה המכונה Ubuntu-logo.png אשר שמורה (ככל הנראה אצל השרת).

עתה נראה מהי תשובתו של השרת על כך בחזרה אל הלקוח בחבילה מספר 11-

```
HTTP/1.1 200 OK
Date: Mon, 09 Jan 2023 18:03:43 GMT
Server: Apache/2.4.52 (Ubuntu)
Last-Modified: Fri, 30 Sep 2022 04:09:50 GMT
ETag: "cfa-5e9dd2a489f80"
Accept-Ranges: bytes
Content-Length: 3322
Keep-Alive: timeout=5, max=99
Connection: Keep-Alive
Content-Type: image/png
```

ניתן לראות שהשרת אישר את הבקשה והוא שולח בחזרה את התמונה אל הלקוח שלנו. בגוף ההודעה ישנו מידע המייצג את התמונה שהתבקשנו להעביר. מידע זה יפוצל ב3 חבילות החל מחבילה 11 (שבה נמצא הראש) עד לחבילה 13 וזאת בשל ה mms כל חבילה מחולקת לפי הגודל שיכול להכנס ב mms ואז מוסיפים לה header מתאים משום שהheader הוא תלוי בגודל ההודעה.

בס"ד

HTTP/1.1 200 OK (PNG)[Malformed Packet]	1514	HTTP	10.0.2.5	10.0.2.4	0.155677590	11
Continuation	1514	HTTP	10.0.2.5	10.0.2.4	0.155682287	12
Continuation	777	HTTP	10.0.2.5	10.0.2.4	0.155682586	13

בחבילה מספר 14 הדפדפן שולח אל השרת שלנו חבילת tcp אשר היא מאשרת שהוא קיבל את המידע http שהועבר אליו מן השרת. חבילה זו היא חבילת tcp עם דגל ackדלוק-

```
Frame 14: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface enp0s3, id 0
Ethernet II, Src: PcsCompu_f5:94:fc (08:00:27:f5:94:fc), Dst: PcsCompu_16:89:3a (08:00:27:16:89:3a)
Internet Protocol Version 4, Src: 10.0.2.5, Dst: 10.0.2.4
Transmission Control Protocol, Src Port: 58992, Dst Port: 80, Seq: 639, Ack: 7085, Len: 0
```

```
0000  08 00 27 16 89 3a 08 00  27 f5 94 fc 08 00 45 00   ··'··:·· '·····E·
0010  00 34 69 b9 40 00 40 06  b9 02 0a 00 02 05 0a 00   ·4i·@·@· ........
0020  02 04 e6 70 00 50 41 03  56 77 4b c7 e3 9a 80 10   ···p·PA· VwK·····
0030  01 e9 89 44 00 00 01 01  08 0a be 4d 36 f4 08 3b   ···D···· ···M6··;
0040  28 6d                                              (m
```

עתה בחבילה מספר 15 הלקוח יבקש מן השרת קובץ מסויים –

```
......T..GET /favicon.ico HTTP/1.1
Host: 10.0.2.4
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:103.0) Gecko/20100101 Firefox/103.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
Referer: http://10.0.2.4/
```

ניתן לראות שהלקוח ביקש מן השרת את הקובץ favicon.ico בבקשת get .

עתה בחבילה 16 נקבל את תשובת השרת על החבילה שהיא-

```
HTTP/1.1 404 Not Found
Date: Mon, 09 Jan 2023 18:03:43 GMT
Server: Apache/2.4.52 (Ubuntu)
Content-Length: 270
Keep-Alive: timeout=5, max=98
Connection: Keep-Alive
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL was not found on this server.</p>
<hr>
<address>Apache/2.4.52 (Ubuntu) Server at 10.0.2.4 Port 80</address>
</body></html>
```

בס"ד

ניתן לראות שקיבלנו הודעה עם מספר אישור 404 אשר אומר לנו שהקובץ לא נמצא בשרת ולכן לא החזרנו את הקובץ הנ"ל ולא נוכל להשתמש בו בדפדפן.

בחבילה מספר 17 אנו מקבלים מהדפדפן חבילת אישור על כך שהוא קיבל את חבילה 16 מן השרת (שאמרה לנו שהקובץ אינו נמצא) חבילה זו היא מסוג tcp עם דגל ack דלוק-

```
Frame 17: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface enp0s3, id 0
Ethernet II, Src: PcsCompu_f5:94:fc (08:00:27:f5:94:fc), Dst: PcsCompu_16:89:3a (08:00:27:16:89:3a)
Internet Protocol Version 4, Src: 10.0.2.5, Dst: 10.0.2.4
Transmission Control Protocol, Src Port: 58992, Dst Port: 80, Seq: 927, Ack: 7571, Len: 0
```

```
0000  08 00 27 16 89 3a 08 00  27 f5 94 fc 08 00 45 00   ··'··:·· '·····E·
0010  00 34 69 bb 40 00 40 06  b9 00 0a 00 02 05 0a 00   ·4i·@·@· ........
0020  02 04 e6 70 00 50 41 03  57 97 4b c7 e5 80 80 10   ···p·PA· W·K·····
0030  01 f5 86 16 00 00 01 01  08 0a be 4d 37 02 08 3b   ........ ···M7··;
0040  28 7b                                              ({
```

עתה השארנו את הקובץ פתוח ולכן אין לנו בתעבורה חבילות של סגירת חיבור ה tcp ופה נגמרת הנספת התעבורה שלנו

בשלב הבא נפתח את הדפדפן אך עתה במצב של גלישה בסתר (incognito) ונעשה את אותו הדבר כלומר נכתוב בשורת החיפוש את כתובת הip של המחשב עליו נמצא השרת 10.0.2.4. גלישה בסתר היא מצב בו אנו גולשים אך כל המידע שנשמר בעבר בגלישה בדפדפן זה(למשל cookies) שנאספו אינו נשמר ובנוסף החיפוש או פעולות הנמצאות בדפדפן במצב זה אינן נשמרות.

עתה נראה את הצילום מן הוויר שארק של החבילות שהועברו -

Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3192812652 TSecr=0 WS=128 [SYN] 80 → 58996	74	TCP	10.0.2.4	10.0.2.5	0.000000000	1
Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=138175971 TSecr=3192812652 WS=128 [SYN, ACK] 58996 → 80	74	TCP	10.0.2.5	10.0.2.4	0.000036901	2
Seq=1 Ack=1 Win=64256 Len=0 TSval=3192812652 TSecr=138175971 [ACK] 80 → 58996	66	TCP	10.0.2.4	10.0.2.5	0.000616387	3
GET / HTTP/1.1	406	HTTP	10.0.2.4	10.0.2.5	0.000616425	4
Seq=1 Ack=341 Win=64896 Len=0 TSval=138175972 TSecr=3192812652 [ACK] 58996 → 80	66	TCP	10.0.2.5	10.0.2.4	0.000642187	5
HTTP/1.1 200 OK (text/html)	1514	HTTP	10.0.2.5	10.0.2.4	0.001558308	6
Continuation	1514	HTTP	10.0.2.5	10.0.2.4	0.001562458	7
Continuation	647	HTTP	10.0.2.5	10.0.2.4	0.001562747	8
Seq=341 Ack=3478 Win=62592 Len=0 TSval=3192812654 TSecr=138175973 [ACK] 80 → 58996	66	TCP	10.0.2.4	10.0.2.5	0.002317505	9
GET /icons/ubuntu-logo.png HTTP/1.1	364	HTTP	10.0.2.4	10.0.2.5	0.237465389	10
HTTP/1.1 200 OK (PNG)[Malformed Packet]	1514	HTTP	10.0.2.5	10.0.2.4	0.237982034	11
Continuation	1514	HTTP	10.0.2.5	10.0.2.4	0.237985220	12
Continuation	777	HTTP	10.0.2.5	10.0.2.4	0.237985512	13
Seq=639 Ack=7085 Win=62592 Len=0 TSval=3192812891 TSecr=138176209 [ACK] 80 → 58996	66	TCP	10.0.2.4	10.0.2.5	0.239216949	14
GET /favicon.ico HTTP/1.1	354	HTTP	10.0.2.4	10.0.2.5	0.248726474	15
HTTP/1.1 404 Not Found (text/html)	552	HTTP	10.0.2.5	10.0.2.4	0.249158181	16
Seq=927 Ack=7571 Win=64128 Len=0 TSval=3192812901 TSecr=138176221 [ACK] 80 → 58996	66	TCP	10.0.2.4	10.0.2.5	0.249786507	17

ניתן לראות בבירור שהתעבורה שקרתה בעת פתיחת הדפדפן וחיבורו אל השרת מן המחשב האחר בדפדפן של גלישה בסתר זהה לפתיחה בדפדפן של גלישה רגילה. דבר זה קורה משום שכאשר גלשנו מדפדפן רגיל זוהי הפעם הראשונה שאנו מגיעים לאתר זה (השרת ששמור לנו במחשב במקורי) ולכן הפעולות שהדפדפן יבצע זהות למצב incognito משום שאין לו שום מידע על אתר זה.

חלק שני – שרת DNS אוטוריטטיבי

ראשית כל נבצע קונפיגורציה אל השרת ונגדיר את זה שכל פעם שיבקשו מהשרת DNS שלנו שאילתת DNS הוא יעביר אותה לשרת DNS העיקרי של גוגל 8.8.8.8 או לשרת המשני של גוגל 8.8.4.4

לאחר מכן גוגל ישמור אותה בcache שלו ויחזיר את התגובה לבקשת ה DNS למי ששאל אותו.

כלומר שרת הDNS שלנו הוא מעין Local resolver אשר מפנה את הבקשות לשרת של גוגל ונעזר בו.

נגדיר בקובץ resolve.conf :

```
nameserver 10.0.2.4
options edns0 trust-ad
search .
```

בקובץ named.conf.options:

```
  GNU nano 6.2
options {
        directory "/var/cache/bind";

        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.

        forwarders {
         8.8.8.8;
         8.8.4.4;
        };

        //========================================================================
        // If BIND logs error messages about the root key being expired,
        // you will need to update your keys.  See https://www.isc.org/bind-keys
        //========================================================================
        dnssec-validation no;

        listen-on-v6 { any; };
};
```

כעת נרצה לערוך בדיקות לשרת שלנו , נבצע nslookup לאתר hello.com , ניתן להשתמש גם בתוכנה dig המאפשרת לראות פרטים על השאילתת DNS.

נסניף את התעבורה באמצעות wireshark תוך סינון שנראה חבילות של החבילות שנראה רק בפרוטוקול DNS .

כעת נבצע את שאילתת הDNS בפעם הראשונה באמצעות nslookup hello.com כפי שניתן לראות בתמונה הבאה:

```
ariel@ariel-VirtualBox:~$ nslookup hello.com
Server:         10.0.2.4
Address:        10.0.2.4#53

Non-authoritative answer:
Name:   hello.com
Address: 216.239.36.21
Name:   hello.com
Address: 216.239.34.21
Name:   hello.com
Address: 216.239.32.21
Name:   hello.com
Address: 216.239.38.21
Name:   hello.com
Address: 2001:4860:4802:38::15
Name:   hello.com
Address: 2001:4860:4802:36::15
Name:   hello.com
Address: 2001:4860:4802:32::15
Name:   hello.com
Address: 2001:4860:4802:34::15
```

ראשית כל השרת פונה אל הלוקאל ריסולבר בחבילה הראשונה ומבצע שאילתה רקורסיבית .

כפי שניתן לראות בתמונה הבאה כי הדגל דלוק !!!

```
+ Frame 85: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface any, id 0
+ Linux cooked capture v1
+ Internet Protocol Version 4, Src: 10.0.2.4, Dst: 10.0.2.4
+ User Datagram Protocol, Src Port: 48737, Dst Port: 53
- Domain Name System (query)
    Transaction ID: 0x5db7
  - Flags: 0x0100 Standard query
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... .0.. .... = Z: reserved (0)
      .... .... ...0 .... = Non-authenticated data: Unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  + Queries
    [Response In: 88]
```

לאחר מכן ,

ניתן לראות כי אנו מבצעים את השאילתה מהשרת שלנו בכתובת 10.0.2.4 ומקבלים תשובה שהיא לא איטרטיבית כלומר יתכן שכאשר פנינו לגוגל לגוגל כבר היה בcache את הרשומה של hello.com ב cache שלו.

ניתן לראות כי יש OPT בחבילה כלומר זה נמצא ב cache של גוגל-

86	6.159689320	10.0.2.4	8.8.4.4	DNS	94 Standard query 0x2fd1 A hello.com OPT

```
+ Frame 85: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface any, id 0
+ Linux cooked capture v1
+ Internet Protocol Version 4, Src: 10.0.2.4, Dst: 10.0.2.4
+ User Datagram Protocol, Src Port: 48737, Dst Port: 53
- Domain Name System (query)
    Transaction ID: 0x5db7
  - Flags: 0x0100 Standard query
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... .0.. .... = Z: reserved (0)
      .... .... ...0 .... = Non-authenticated data: Unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  + Queries
    [Response In: 88]
```

כעת נראה שאכן השרת שלנו מפנה אותנו לשרת של גוגל על מנת לקבל את כתובת הip של האתר hello.com .

dns

No.	Time	Source	Destination	Protocol	Length	Info
85	6.158469392	10.0.2.4	10.0.2.4	DNS	71	Standard query 0x5db7 A hello.com
86	6.159689320	10.0.2.4	8.8.4.4	DNS	94	Standard query 0x2fd1 A hello.com OPT
87	6.208447320	8.8.4.4	10.0.2.4	DNS	146	Standard query response 0x2fd1 A hello.com A 216.239.32.21 A 216.239.34.21 A 216.239.36.21 A 216.239.38.21 OPT
88	6.208754767	10.0.2.4	10.0.2.4	DNS	135	Standard query response 0x5db7 A hello.com A 216.239.36.21 A 216.239.34.21 A 216.239.32.21 A 216.239.38.21
89	6.209057607	10.0.2.4	10.0.2.4	DNS	71	Standard query 0x2124 AAAA hello.com
90	6.209274401	10.0.2.4	8.8.4.4	DNS	94	Standard query 0x31c3 AAAA hello.com OPT
91	6.268847475	8.8.4.4	10.0.2.4	DNS	194	Standard query response 0x31c3 AAAA hello.com AAAA 2001:4860:4802:32::15 AAAA 2001:4860:4802:38::15 AAAA 2001:4860:4802:36::15 AAAA 2001:4860:4802:34::15 OPT
92	6.269939004	10.0.2.4	10.0.2.4	DNS	183	Standard query response 0x2124 AAAA hello.com AAAA 2001:4860:4802:38::15 AAAA 2001:4860:4802:32::15 AAAA 2001:4860:4802:34::15 AAAA 2001:4860:4802:36::15

ניתן לראות בחבילה מספר 2 כלומר בשורה הראשונה שאנו מבצעים standard query כלומר שאילתת DNS סטנדרטית משרת הDNS שלנו באייפי 10.0.2.4 לשרת הDNS הראשי של גוגל כלומר אנו מעבירים אליו את השאלה מכיוון שהגדרנו זאת ב forwarders.

```
+ Frame 5: 92 bytes on wire (736 bits), 92 bytes captured (736 bits) on interface enp0s3, id 0
+ Ethernet II, Src: PcsCompu_16:89:3a (08:00:27:16:89:3a), Dst: RealtekU_12:35:00 (52:54:00:12:35:00)
+ Internet Protocol Version 4, Src: 10.0.2.4, Dst: 8.8.8.8
+ User Datagram Protocol, Src Port: 52576, Dst Port: 53
- Domain Name System (query)
    Transaction ID: 0xb16d
  + Flags: 0x0110 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 1
  - Queries
    - hello.com: type A, class IN
        Name: hello.com
        [Name Length: 9]
        [Label Count: 2]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
  + Additional records
    [Response In: 6]
```

כלומר אנחנו מבקשים ממנו את ה ipv4 כי ניתן לראות כי ה type הינו A באשר הns = hello.com

לאחר מכן בחבילה מספר3 שרת הDNS הראשי של גוגל מחזיר לנו תגובה אל ה local resolver לשאילתת הDNS

```
+ User Datagram Protocol, Src Port: 53, Dst Port: 52576
- Domain Name System (response)
    Transaction ID: 0xb16d
  + Flags: 0x8190 Standard query response, No error
    Questions: 1
    Answer RRs: 4
    Authority RRs: 0
    Additional RRs: 1
  + Queries
  - Answers
    + hello.com: type A, class IN, addr 216.239.36.21
    - hello.com: type A, class IN, addr 216.239.38.21
        Name: hello.com
        Type: A (Host Address) (1)
        Class: IN (0x0001)
        Time to live: 14400 (4 hours)
        Data length: 4
        Address: 216.239.38.21
    + hello.com: type A, class IN, addr 216.239.32.21
    + hello.com: type A, class IN, addr 216.239.34.21
  + Additional records
```

באשר ישנן 2 רשומות אשר ממפות בין הדומיין hello.com לבין כתובת הIpv4

אחת מהן ל ip : 216.239.36.21 והשנייה לIp 216.239.38.21.

בנוסף לכך ניתן לראות כי ה ttl – time to live שמוקצה לרשומות ששרת ה DNS של גוגל הוא 4 שעות כלומר 14,000 שניות ולכן רשומות אלו ישמרו ב Cache של שרת הDNS שלנו למשך הזמן הזה.

לאחר מכן בחבילה מספר 4 אנו שולחים את התגובה מה local resolver כלומר משרת הDNS שיצרנו ללקוח עצמו .

חבילות 5-8 גם מתארות את תהליך השאילתה רק כאשר הפעם נמפה בין הדומיין לבין IPV6 כפי שניתן לראות מכיוון שסוג הרשומות הינו AAAA.

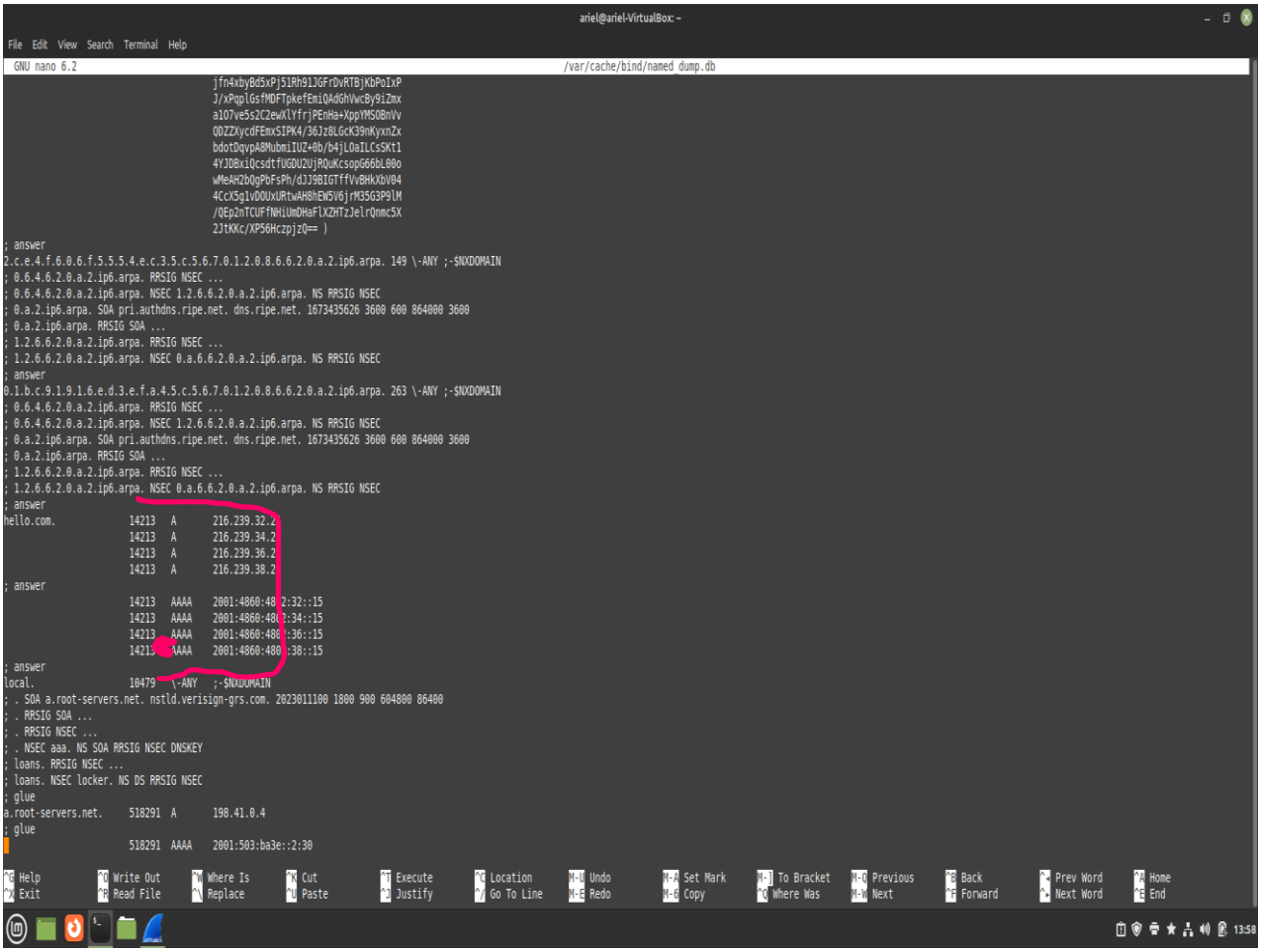
כעת כאשר אנו נבקש שוב פעם את הדומיין hello.com ונסניף את התעבורה באמצעות wireshark נקבל כי כאשר נסנן לפי dns.

נקבל כי אין חבילות כלל מכיוון שהשרת שלנו בכלל לא יפנה אל השרת של גוגל זאת מכיוון שראשית הוא בודק ב local cache ובודק האם קיימת רשומה שממפה .

מכיוון שהTTL בחיפוש הראשון היה 4 שעות ולא עברו 4 שעות מתקיים כי השרת שלנו שבכתובת 10.0.2.4 שלף את הרשומה מהcache המקומי .

בתמונה הבאה ניתן לראות את הcache שלנו לאחר שהרצנו את הפקודה :

sudo rndc dumpdb -cache



ניתן לראות כי הרשומות של hello.com שמורות לנו בcache וכאשר נבצע שוב nslookup hello.com הוא כלל לא יפנה אל השרת אלה ישר ימצא את הרשומות בcache וישתמש בהן.

כעת נרצה להגדיר את שרת הDNS שהתקנו על המכונה הוירטואלית שלנו להיות שרת אוטורטיבי :

נוסיף אל הקובץ : named.conf.local את ה zone הבא :

```
GNU nano 6.2                    /etc/bind/name
zone "biu.ac.il" {
        type master;
        file"/etc/bind/db.biu.ac.il";
};
```

כלומר אנו מגדירים לשרת קובץ שיכיל את הניהול של הדומיין : biu.ac.il והקובץ הנ"ל יהיה הקובץ

/etc/bind/db.biu.ac.il כעת נייצר את הzone file ואת הרשומות המבוקשות בתרגיל כפי שניתן לראות בתמונה הבאה:

```
  GNU nano 6.2                    /etc/bind/db.biu.ac.il
; BIND data file for the biu.ac.il  interface
;
$TTL    604800
@       IN      SOA     biu.ac.il. root.biu.ac.il. (
                            34          ; Serial
                            604800      ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@           IN      NS      ns
@           IN      A       10.0.2.4
ns          IN      A       10.0.2.4
www         IN      A       10.0.2.4
@           IN      MX      10   mail
mail        IN      A       10.0.2.4
```

בתמונה הבאה ניתן לראות את כלל השאילות שביצענו , ננתח כל אחת מן השאילות הנ"ל :

```
ariel@ariel-VirtualBox:~$ nslookup ns.biu.ac.il
Server:         10.0.2.4
Address:        10.0.2.4#53

Name:   ns.biu.ac.il
Address: 10.0.2.4

ariel@ariel-VirtualBox:~$ nslookup biu.ac.il
Server:         10.0.2.4
Address:        10.0.2.4#53

Name:   biu.ac.il
Address: 10.0.2.4

ariel@ariel-VirtualBox:~$ nslookup www.biu.ac.il
Server:         10.0.2.4
Address:        10.0.2.4#53

Name:   www.biu.ac.il
Address: 10.0.2.4

ariel@ariel-VirtualBox:~$ nslookup mail.biu.ac.il
Server:         10.0.2.4
Address:        10.0.2.4#53

Name:   mail.biu.ac.il
Address: 10.0.2.4
```

שאילתה מספר 1: (nslookup – ns.biu.ac.il)

http || dns || tcp

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.4	10.0.2.4	DNS	74	Standard query 0x03bd A ns.biu.ac.il
2	0.000204708	10.0.2.4	10.0.2.4	DNS	90	Standard query response 0x03bd A ns.biu.ac.il A 10.0.2.4
3	0.000428698	10.0.2.4	10.0.2.4	DNS	74	Standard query 0x0685 AAAA ns.biu.ac.il
4	0.000579106	10.0.2.4	10.0.2.4	DNS	115	Standard query response 0x0685 AAAA ns.biu.ac.il SOA biu.ac.il

נשים לב כי כי בחבילה הראשונה אנו שואלים את שרת הDNS שלנו שהוא גם בip 10.0.2.4 מה הip של הדומיין ns.biu.ac.il כלומר מה הip של שרת השמות של biu.ac.il .

נשים לב כי אנו פונים אל שרת הDNS המקומי ולא אל שרת ה dns של בר אילן!

לאחר מכן שרת ה DNS מחזיר בתשובה את ה ip של ns.biu.ac.il מכיוון שרשומה זו קיימת אצלו ב zone file שהגדרנו לעיל כפי שניתן לראות כי השרת אחראי על הדומיין ולכן הדגל דלוק ולכן הוא מחזיר את הרשומה שממפה בין הדומיין שביקשנו לבין הip של הדומיין .

```
+ Linux cooked capture v1
+ Internet Protocol Version 4, Src: 10.0.2.4, Dst: 10.0.2.4
- User Datagram Protocol, Src Port: 53, Dst Port: 35754
    Source Port: 53
    Destination Port: 35754
    Length: 54
    Checksum: 0x184f [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
  + [Timestamps]
    UDP payload (46 bytes)
- Domain Name System (response)
    Transaction ID: 0x03bd
  - Flags: 0x8580 Standard query response, No error
      1... .... .... .... = Response: Message is a response
      .000 0... .... .... = Opcode: Standard query (0)
      .... .1.. .... .... = Authoritative: Server is an authority for domain
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... 1... .... = Recursion available: Server can do recursive queries
      .... .... .0.. .... = Z: reserved (0)
      .... .... ..0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
      .... .... ...0 .... = Non-authenticated data: Unacceptable
      .... .... .... 0000 = Reply code: No error (0)
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
```

בחבילות 3 ו4 אנו שוב מבצעים שאילתה רק שהפעם נבקש את כתובת ה ip של הדומיין בגרסה 6.

ולכן נבקש רשומה מ type = AAAA .

כעת השרת החזיר בתשובה כי ns.biu.ac.il כי השם הדומיין של שרת השמות שהיה המקור העיקרי של המידע ל zone file הינו biu.ac.il כפי שניתן לראות בתמונה הבאה כי הmname = biu.ac.il

```
    UDP payload (71 bytes)
  Domain Name System (response)
    Transaction ID: 0x0685
    Flags: 0x8580 Standard query response, No error
      1... .... .... .... = Response: Message is a response
      .000 0... .... .... = Opcode: Standard query (0)
      .... .1.. .... .... = Authoritative: Server is an authority for domain
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... 1... .... = Recursion available: Server can do recursive queries
      .... .... .0.. .... = Z: reserved (0)
      .... .... ..0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
      .... .... ...0 .... = Non-authenticated data: Unacceptable
      .... .... .... 0000 = Reply code: No error (0)
    Questions: 1
    Answer RRs: 0
    Authority RRs: 1
    Additional RRs: 0
    Queries
      ns.biu.ac.il: type AAAA, class IN
    Authoritative nameservers
      biu.ac.il: type SOA, class IN, mname biu.ac.il
    [Request In: 3]
    [Time: 0.000150408 seconds]
```

שאילתה מספר 2(nslookup biu.ac.il)

	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.4	10.0.2.4	DNS	71	Standard query 0x10c2 A biu.ac.il
2	0.000656892	10.0.2.4	10.0.2.4	DNS	87	Standard query response 0x10c2 A biu.ac.il A 10.0.2.4
3	0.000942700	10.0.2.4	10.0.2.4	DNS	71	Standard query 0x3544 AAAA biu.ac.il
4	0.001143479	10.0.2.4	10.0.2.4	DNS	112	Standard query response 0x3544 AAAA biu.ac.il SOA biu.ac.il

בדומה לשאילתה מספר 1 , בהתחלה אנו נבקש משרת הDNS שהגדרנו את השאילתה :

? A biu.ac.il כלומר האם אתה יודע מה הipv4 של הדומיין biu.ac.il ובתגובה לכך השרת עונה

A biu.ac.il A 10.0.2.4 כלומר האייפי של הדומיין המבוקש הינו 10.0.2.4.

בחבילות 3 ו4 אנו שוב מבצעים שאילתה רק שהפעם נבקש את כתובת ה ip של הדומיין בגרסה 6.

ולכן נבקש רשומה מ type = AAAA .

כעת השרת החזיר בתשובה כי ns.biu.ac.il כי השם הדומיין של שרת השמות שהיה המקור העיקרי של המידע ל zone file הינו biu.ac.il כפי שניתן לראות בתמונה הבאה כי הmname = biu.ac.il

כמו בדומה לשאילתה 1 , אך אנו לא מתמקדים בכתובות בגרסה 6 ולא נעמיק בכך.

שאילתה 3(nslookup www.biu.ac.il):

בשאילתה זו נפנה לאתר האינטרנט בדומיין www.biu.ac.il נשים לב כי בשרת DNS שלנו הגדרנו zone file אשר מכיל את הרשומה הבאה:

```
www        IN      A        10.0.2.4
```

כלומר כאשר כתוב www בסופו של דבר אנו נוסיף נקודה בסוף והנקודה תתחלף לbiu.ac.il ז"א שנקבל www.biu.ac.il זוהי רשומה מסוג A אשר ממפה מהדומיין הנ"ל אל כתובת האייפי 10.0.2.4 כלומר כתובת הip של המחשב שלנו .

Source	Destination	Protocol	Length	Info
10.0.2.4	10.0.2.4	DNS	75	Standard query 0x5f4f A www.biu.ac.il
10.0.2.4	10.0.2.4	DNS	91	Standard query response 0x5f4f A www.biu.ac.il A 10.0.2.4
10.0.2.4	10.0.2.4	DNS	75	Standard query 0xcbde AAAA www.biu.ac.il
10.0.2.4	10.0.2.4	DNS	116	Standard query response 0xcbde AAAA www.biu.ac.il SOA biu.ac.il

בדומה לשאילתות הקודמות ניתן לראות כי המחשב שלנו מבקש משרת הDNS שנמצא על אותה כתובת Ip את הכתובת של הדומיין www.biu.ac.il באמצעות שאילתת DNS :

? www.biu.ac.il A ולאחר מכן שרת ה DNS הולך ובודק בzone file ורואה כי יש לו רשומה אשר ממפה בין הדומיין הנ"ל לבין כתובת ה ipv4 הבאה 10.0.2.4 ולכן מחזיר תשובה זו למחשב שלנו.

שתי הבקשות האחרות הן ב iPV6 ולא נתעמק בהן גם .

כעת נעבור אל השאילתה האחרונה שמוצאת את כתובת הדואר של דומיין מסוים.

<u>שאילתה רביעית (nslookup mail.biu.ac.il) –</u>

כעת נבצע שאילתת DNS על הדומיין mail.biu.ac.il .

Source	Destination	Protocol	Length	Info
10.0.2.4	10.0.2.4	DNS	76	Standard query 0xac78 A mail.biu.ac.il
10.0.2.4	10.0.2.4	DNS	92	Standard query response 0xac78 A mail.biu.ac.il A 10.0.2.4
10.0.2.4	10.0.2.4	DNS	76	Standard query 0x9869 AAAA mail.biu.ac.il
10.0.2.4	10.0.2.4	DNS	117	Standard query response 0x9869 AAAA mail.biu.ac.il SOA biu.ac.il

כפי שניתן לראות אנו שואלים שאילתת DNS סטנדרטית ומעוניינים לדעת מה היא כתובת ה ipv4 של שרת הדואר של biu.ac.il . כעת נשים לב לפרט קצת שונה כאן, כאשר שרת הDNS שלנו מסתכל בתוך הzone file הוא רואה את הרשומה הבאה:

```
@          IN     MX       10   mail
mail       IN     A        10.0.2.4
```

כאשר לרשומת ה MX יש גם תכונה נוספת שנקראת priority כלומר כאשר נבקש את שרת המייל אז יש לרשומה הזאת קדימות על רשומה למשל עם priority = 20.

לכן השרת שלנו מחזיר אל המחשב שלנו את הכתובת האייפי של שרת הדואר של בר אילן שהיא 10.0.2.4.

לסיכום :

בעצם כל מה שעשינו פה זה להקים שרת DNS לוקאלי על המחשב שלנו , כאשר השרת שלנו לא פונה אל שרת ה dns של בר אילן אלא משתמש בzone file שנמצא לוקאלית בתיקייה.

בתוך ה zone file ישנן רשימות מסוג : SOA,A,AAAA,MX.

רשימת SOA = Start Of authority כלומר ההתחלה של השרת האוטוריטטיבי שלנו שבה ישנם שדות כמו TTL – שמסמן כמה זמן לכל רשומה יש שהיא תחייה בcache וכו…

לאחר מכן ביצענו שאילתות DNS לשרת והראנו כי אכן אנו לא פונים לשרתים של בר אילן אלה לשרת האוטוריטטיבי שהגדרנו.

בס"ד

חלק 3 - שרת Proxy

בחלק זה נרצה להתקין שרת פרוקסי על אחת מן המכונות הוירטואליות שלנו.

שרת פרוקסי (שרת מתווך) הוא שרת אשר מהווה שמתווך בין לקוח מסויים לבין שרתים אחרים שונים. מה שקורה בפועל הוא שלקוח פונה לשרת הפרוקסי עם בקשה כלשהי, שרת הפרוקסי בודק אם הוא יכול בעצמו לענות על הבקשה (אם המידע המתאים נמצא ב cache שלו) ובמידה ולא יכול הוא פותח חיבור נפרד ופונה לשרת הרגיל. לאחר שהשרת הרגיל מחזיר לשרת הפרוקסי את התשובה שרת הפרוקסי מוריד את המידע לcache שלו ואז מחזיר את המידע ללקוח מן הcache היתרון העיקרי בעבודה עם שרת פרוקסי הוא שכאשר לקוחות נוספים מבקשים את אותו מידע מן השרת אז שרת הפרוקסי יוכל להחזיר אותו ישירות מן ה cache.

נתקין שרת פרוקסי על המכונה הוירטואלית באשר כתובת הipv4 של שרת הפרוקסי תהיינה 10.0.2.4 וכתובת ה ip של הלקוח שיפנה אל השרת שלנו הינה 10.0.2.5. (נתקין את שרת הפרוקסי על המכונה הראשונה עליה התקנו את שרת ה dns)

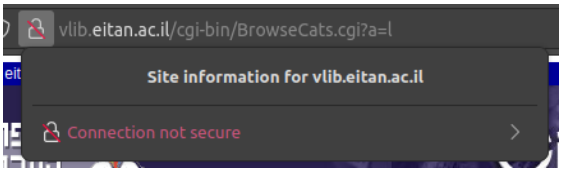
נתקין את שרת הפרוקסי בעזרת הפקודה שקיבלנו בעבודה- sudo apt install squid. ולאחר מכן נריץ את שרת הפרוקסי בעזרת הפקודה- sudo service squid start.

בשלב הבא נכנס להגדרות הדפדפן שלנו ונגדיר את שרת הפרוקסי שלנו כשרת הפרוקסי של הדפדפן (נזין את כתובת ה ip והפורט שלו). בנוסף נשנה בקובץ squid.conf את השורה http_access deny all אל השורה http_access allow all על מנת שנוכל לקבל את כל עמודי ה http. (את אותם פעולות ביצענו במחשב השני שבו הלקוח שיפנה אל השרת נמצא).

כעת לאחר שהתקנו את שרת הפרוקסי שמאזין לפורט 3128 ונמצא בכתובת ה ip– 10.0.2.4.

נתחבר אל האתר הבא מהלקוח :vlib.eitan.ac.il שהוא אתר http כמבוקש בהגדרת התרגיל

ניתן לראות גם כי החיבור עם אתר זה לא מאובטח כי הוא לא ב https כפי שניתן לראות בתמונה הבאה :



כעת נסביר את התהליך בקצרה ולאחר מכן נדגים אותו באמצעות Wireshark .

כאשר הלקוח יתחבר הוא ראשית כל יפתח חיבור TCP עם שרת הפרוקסי שלנו כפי שניתן לראות בתמונה הבאה:

5	1.260742256	10.0.2.5	10.0.2.4	TCP	74 49856 → 3128 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=…
6	1.260782051	10.0.2.4	10.0.2.5	TCP	74 3128 → 49856 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460…
7	1.261531191	10.0.2.5	10.0.2.4	TCP	66 49856 → 3128 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=4039302…

ניתן לראות כי אכן מתרחש תהליך לחיצת הידיים עם שרת הפרוקסי שמאזין לפורט 3128 כאשר הלקוח מתחבר אל השרת.

וכאשר הוא מחזיר ACK על הACK של השרת הוא שולח גם בקשת HTTP על מנת לקבל את האתר

שהלקוח ביקש כפי שניתן לראות בתמונה הבאה:

8	1.261531285	10.0.2.5	10.0.2.4	HTTP	273 CONNECT vlib.eitan.ac.il:443 HTTP/1.1

לאחר מכן השרת פונה אל ה ל local resolver על מנת למצוא את כתובת הIp של הדומיין (vlib.eitan.ac.il) ,אך נציין כאן כי ניקינו את הcache לפני ההסנפה ולכן הכתובת של הdomain המבוקש לא נמצאת בcache. ולכן ה local resolver יפנה אל השרת הראשי של גוגל על מנת לקבל את כתובת האייפי של הדומיין!

נשים לב כי על מנת לקבל את האייפי שרת הפרוקסי פותח חיבור נוסף תחת פרוטוקול התעבורה UDP ומעליו בשכבת האפליקציה פרוטוקול הDNS שבאמצעותו נוכל לקבל את כתובת הרשת כפי שניתן לראות בתמונה הבאה:

10	1.269227778	10.0.2.4	8.8.8.8	DNS	99 Standard query 0xbcae A vlib.eitan.ac.il OPT
11	1.269308733	10.0.2.4	8.8.8.8	DNS	99 Standard query 0x00d4 AAAA vlib.eitan.ac.il OPT
12	1.376858244	8.8.8.8	10.0.2.4	DNS	103 Standard query response 0xbcae A vlib.eitan.ac.il A 199.203.54.24 OPT
13	1.377719944	10.0.2.4	199.203.54.24	TCP	74 53420 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2254991478 TSecr=0 WS=128
14	1.386858693	8.8.8.8	10.0.2.4	DNS	134 Standard query response 0x00d4 AAAA vlib.eitan.ac.il SOA eitan.ac.il OPT

בשורה האפורה שרת הפרוקסי כבר מתחיל ליצור חיבור TCP עם האתר בכתובת 199.203.54.24

ופותח חיבור TCP נוסף !!! משום מה השרת פרוקסי מקבל RESET מהאתר המבוקש מה שגורם לכך ששאר החיבורים נכשלים ובמהלך בקשות הHTTP אנו שולחים משרת הפרוקסי ללקוח הראשון 503 כלומר שאתר האינטרנט לא זמין.

לאחר מכן הלקוח מנסה שוב ופותח חיבור TCP נוסף על מנת לקבל את אתר האינטרנט כפי שניתן לראות בתמונה הבאה:

10.0.2.5	10.0.2.4	TCP	74 49858 → 3128 [SYN] Seq=0 Win=6
10.0.2.4	10.0.2.5	TCP	74 3128 → 49858 [SYN, ACK] Seq=0
10.0.2.5	10.0.2.4	TCP	66 49858 → 3128 [ACK] Seq=1 Ack=1

כעת לשרת פרוקסי יש כבר את הכתובת האייפי של אתר האינטרנט מכיוון שהוא מצא אותה בשאילתת הDNS שהראנו למעלה .

לכן כעת כאשר הלקוח שולח אל השרת בקשה GET לאתר ושרת הפרוקסי מאשר כי הוא קיבל בקשה זו .

לאחר מכן שרת הפרוקסי פותח חיבור TCP נוסף מול האתר שנמצא בכתובת שציינו לעיל על מנת להוריד אליו את האתר כפי שניתן לראות בתמונה הבאה:

No.	Time	Source	Destination	Protocol	Length	Info
31	3.535514452	10.0.2.4	199.203.54.24	TCP	74	55040 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2254993636 TSecr=0 WS=128
32	3.557859991	199.203.54.24	10.0.2.4	TCP	60	80 → 55040 [SYN, ACK] Seq=0 Ack=1 Win=32768 Len=0 MSS=1460
33	3.557925257	10.0.2.4	199.203.54.24	TCP	54	55040 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
35	3.558740763	10.0.2.4	199.203.54.24	HTTP	499	GET / HTTP/1.1
36	3.581142464	199.203.54.24	10.0.2.4	HTTP	725	HTTP/1.1 200 OK (text/html)
37	3.581189033	10.0.2.4	199.203.54.24	TCP	54	55040 → 80 [ACK] Seq=446 Ack=672 Win=63745 Len=0
205	4.053384948	10.0.2.4	199.203.54.24	HTTP	455	GET /favicon.ico HTTP/1.1
206	4.078588434	199.203.54.24	10.0.2.4	HTTP	1514	HTTP/1.1 200 OK (image/x-icon)
207	4.078627708	10.0.2.4	199.203.54.24	TCP	54	55040 → 80 [ACK] Seq=847 Ack=2132 Win=63745 Len=0
208	4.078942438	199.203.54.24	10.0.2.4	HTTP	400	Continuation
209	4.078951800	10.0.2.4	199.203.54.24	TCP	54	55040 → 80 [ACK] Seq=847 Ack=2478 Win=63399 Len=0
230	6.269254155	10.0.2.4	199.203.54.24	HTTP	520	GET /cgi-bin/BrowseCats.cgi?a=l HTTP/1.1
232	6.412048798	199.203.54.24	10.0.2.4	HTTP	1514	HTTP/1.1 200 OK
233	6.412091069	10.0.2.4	199.203.54.24	TCP	54	55040 → 80 [ACK] Seq=1313 Ack=3938 Win=63399 Len=0
234	6.412578832	199.203.54.24	10.0.2.4	HTTP	1354	Continuation
235	6.412589993	10.0.2.4	199.203.54.24	TCP	54	55040 → 80 [ACK] Seq=1313 Ack=5238 Win=63399 Len=0
240	6.414461457	199.203.54.24	10.0.2.4	HTTP	1434	Continuation
241	6.414470859	10.0.2.4	199.203.54.24	TCP	54	55040 → 80 [ACK] Seq=1313 Ack=6618 Win=63399 Len=0
244	6.416139067	199.203.54.24	10.0.2.4	HTTP	1258	Continuation
245	6.416149620	10.0.2.4	199.203.54.24	TCP	54	55040 → 80 [ACK] Seq=1313 Ack=7822 Win=63399 Len=0
248	6.418206081	199.203.54.24	10.0.2.4	HTTP	1434	Continuation
249	6.418217306	10.0.2.4	199.203.54.24	TCP	54	55040 → 80 [ACK] Seq=1313 Ack=9202 Win=63399 Len=0
252	6.431192651	199.203.54.24	10.0.2.4	HTTP	519	Continuation
253	6.431218852	10.0.2.4	199.203.54.24	TCP	54	55040 → 80 [ACK] Seq=1313 Ack=9667 Win=63399 Len=0

בס"ד

נציין כי שרת הפרוקסי פותח מספר חיבורי TCP!

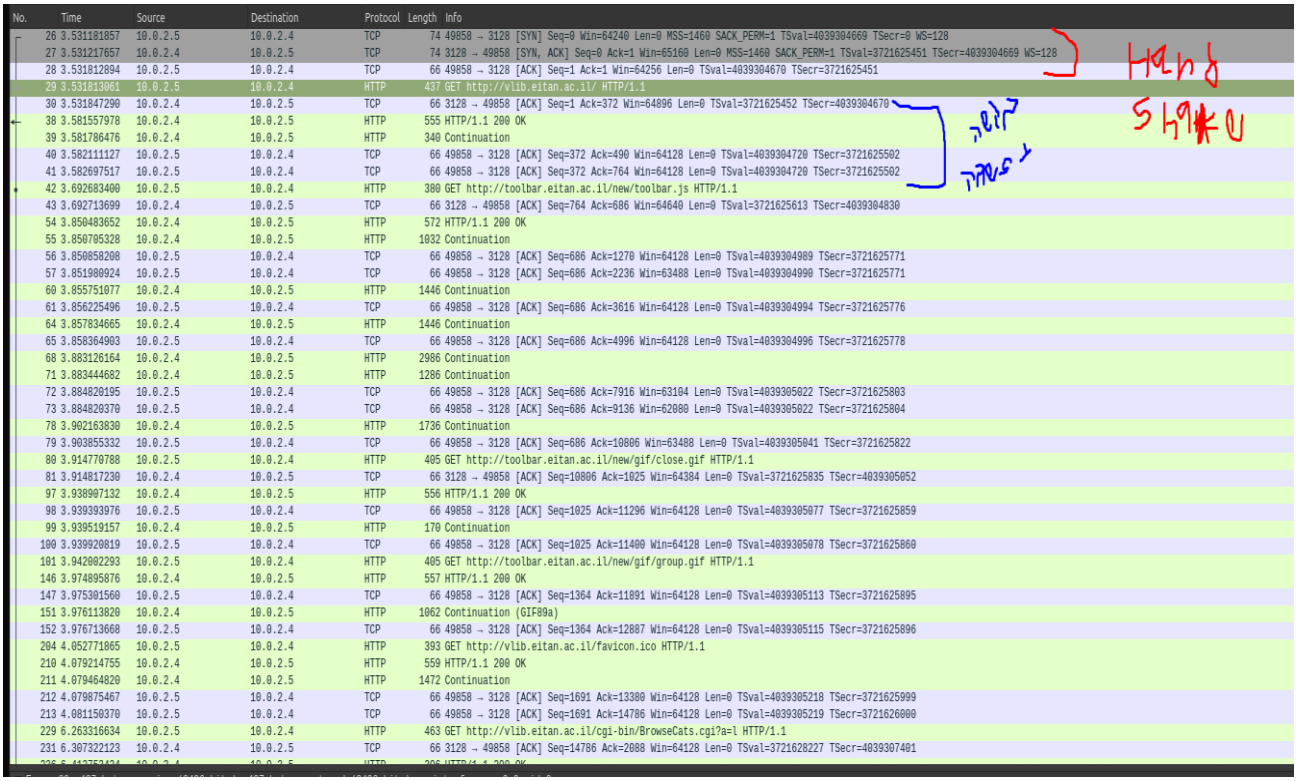
זאת על מנת להביא את את כל הקבצים של האתר קבצי הjs,html,css וכו שאחראים על נראות האתר ועל פעילות האתר.

נשים לב כי התהליך שמתרחש הוא שהלקוח מבקש קובץ מהשרת פרוקסי ואז השרת פרוקסי מבקש את הקובץ מהשרת שבכתובת של האתר ולאחר מכן שהאתר נותן את הקובץ לשרת הפרוקסי שרת הפרוקסי מחזיר את הקובץ אל הלקוח החזרה זו מתבצעת על חיבור ה TCP שנוצר בהתחלה בניהם .

כלומר שרת הפרוקסי יפתח חיבור TCP על מנת לקבל את הקובץ מאתר האינטרנט .

בנוסף נשים לב כי לאחר ששרת הפרוקסי מחזיר את הקובץ אל הלקוח שבכתובת 10.0.2.5 הלקוח מפרק שוב את הקובץ ואז רואה שבפנים צריך לבקש קבצים מסויימים ולכן הוא עושה עוד בקשת GET HTTP אל שרת הפרוקסי על אותו חיבור TCP ואז השרת מחזיר לו שוב וכך הלאה..

ניתן לראות זאת בתמונה הבאה :



No.	Time	Source	Destination	Protocol	Length	Info
26	3.531181857	10.0.2.5	10.0.2.4	TCP	74	49858 → 3128 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=4039304669 TSecr=0 WS=128
27	3.531217657	10.0.2.4	10.0.2.5	TCP	74	3128 → 49858 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=3721625451 TSecr=4039304669 WS=128
28	3.531812894	10.0.2.5	10.0.2.4	TCP	66	49858 → 3128 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=4039304670 TSecr=3721625451
29	3.531813061	10.0.2.5	10.0.2.4	HTTP	437	GET http://vlib.eitan.ac.il/ HTTP/1.1
30	3.531847290	10.0.2.4	10.0.2.5	TCP	66	3128 → 49858 [ACK] Seq=1 Ack=372 Win=64896 Len=0 TSval=3721625452 TSecr=4039304670
38	3.581557978	10.0.2.4	10.0.2.5	HTTP	555	HTTP/1.1 200 OK
39	3.581786476	10.0.2.4	10.0.2.5	HTTP	340	Continuation
40	3.582111127	10.0.2.5	10.0.2.4	TCP	66	49858 → 3128 [ACK] Seq=372 Ack=490 Win=64128 Len=0 TSval=4039304720 TSecr=3721625502
41	3.582697517	10.0.2.5	10.0.2.4	TCP	66	49858 → 3128 [ACK] Seq=372 Ack=764 Win=64128 Len=0 TSval=4039304720 TSecr=3721625502
42	3.692683400	10.0.2.5	10.0.2.4	HTTP	380	GET http://toolbar.eitan.ac.il/new/toolbar.js HTTP/1.1
43	3.692713699	10.0.2.4	10.0.2.5	TCP	66	3128 → 49858 [ACK] Seq=764 Ack=686 Win=64640 Len=0 TSval=3721625613 TSecr=4039304830
54	3.850483652	10.0.2.4	10.0.2.5	HTTP	572	HTTP/1.1 200 OK
55	3.850705328	10.0.2.4	10.0.2.5	HTTP	1032	Continuation
56	3.850858208	10.0.2.5	10.0.2.4	TCP	66	49858 → 3128 [ACK] Seq=686 Ack=1270 Win=64128 Len=0 TSval=4039304989 TSecr=3721625771
57	3.851980924	10.0.2.5	10.0.2.4	TCP	66	49858 → 3128 [ACK] Seq=686 Ack=2236 Win=63488 Len=0 TSval=4039304990 TSecr=3721625771
60	3.855751077	10.0.2.4	10.0.2.5	HTTP	1446	Continuation
61	3.856225496	10.0.2.5	10.0.2.4	TCP	66	49858 → 3128 [ACK] Seq=686 Ack=3616 Win=64128 Len=0 TSval=4039304994 TSecr=3721625776
64	3.857834665	10.0.2.4	10.0.2.5	HTTP	1446	Continuation
65	3.858364903	10.0.2.5	10.0.2.4	TCP	66	49858 → 3128 [ACK] Seq=686 Ack=4996 Win=64128 Len=0 TSval=4039304996 TSecr=3721625778
68	3.883126164	10.0.2.4	10.0.2.5	HTTP	2986	Continuation
71	3.883444682	10.0.2.4	10.0.2.5	HTTP	1286	Continuation
72	3.884820195	10.0.2.5	10.0.2.4	TCP	66	49858 → 3128 [ACK] Seq=686 Ack=7916 Win=63104 Len=0 TSval=4039305022 TSecr=3721625803
73	3.884820370	10.0.2.5	10.0.2.4	TCP	66	49858 → 3128 [ACK] Seq=686 Ack=9136 Win=62080 Len=0 TSval=4039305022 TSecr=3721625804
78	3.902163830	10.0.2.4	10.0.2.5	HTTP	1736	Continuation
79	3.903855332	10.0.2.5	10.0.2.4	TCP	66	49858 → 3128 [ACK] Seq=686 Ack=10806 Win=63488 Len=0 TSval=4039305041 TSecr=3721625822
80	3.914770788	10.0.2.5	10.0.2.4	HTTP	405	GET http://toolbar.eitan.ac.il/new/gif/close.gif HTTP/1.1
81	3.914817230	10.0.2.4	10.0.2.5	TCP	66	3128 → 49858 [ACK] Seq=10806 Ack=1025 Win=64384 Len=0 TSval=3721625835 TSecr=4039305052
97	3.938907132	10.0.2.4	10.0.2.5	HTTP	556	HTTP/1.1 200 OK
98	3.939393976	10.0.2.5	10.0.2.4	TCP	66	49858 → 3128 [ACK] Seq=1025 Ack=11296 Win=64128 Len=0 TSval=4039305077 TSecr=3721625859
99	3.939519157	10.0.2.4	10.0.2.5	HTTP	170	Continuation
100	3.939920819	10.0.2.5	10.0.2.4	TCP	66	49858 → 3128 [ACK] Seq=1025 Ack=11400 Win=64128 Len=0 TSval=4039305078 TSecr=3721625860
101	3.942002293	10.0.2.5	10.0.2.4	HTTP	405	GET http://toolbar.eitan.ac.il/new/gif/group.gif HTTP/1.1
146	3.974895876	10.0.2.4	10.0.2.5	HTTP	557	HTTP/1.1 200 OK
147	3.975301560	10.0.2.5	10.0.2.4	TCP	66	49858 → 3128 [ACK] Seq=1364 Ack=11891 Win=64128 Len=0 TSval=4039305113 TSecr=3721625895
151	3.976113820	10.0.2.4	10.0.2.5	HTTP	1062	Continuation (GIF89a)
152	3.976713668	10.0.2.5	10.0.2.4	TCP	66	49858 → 3128 [ACK] Seq=1364 Ack=12887 Win=64128 Len=0 TSval=4039305115 TSecr=3721625896
204	4.052771865	10.0.2.5	10.0.2.4	HTTP	393	GET http://vlib.eitan.ac.il/favicon.ico HTTP/1.1
210	4.079214755	10.0.2.4	10.0.2.5	HTTP	559	HTTP/1.1 200 OK
211	4.079464820	10.0.2.4	10.0.2.5	HTTP	1472	Continuation
212	4.079875467	10.0.2.5	10.0.2.4	TCP	66	49858 → 3128 [ACK] Seq=1691 Ack=13380 Win=64128 Len=0 TSval=4039305218 TSecr=3721625999
213	4.081150370	10.0.2.5	10.0.2.4	TCP	66	49858 → 3128 [ACK] Seq=1691 Ack=14786 Win=64128 Len=0 TSval=4039305219 TSecr=3721626000
229	6.263316634	10.0.2.5	10.0.2.4	HTTP	463	GET http://vlib.eitan.ac.il/cgi-bin/BrowseCats.cgi?a=l HTTP/1.1
231	6.307322123	10.0.2.4	10.0.2.5	TCP	66	3128 → 49858 [ACK] Seq=14786 Ack=2088 Win=64128 Len=0 TSval=3721628227 TSecr=4039307401

בנוסף לכך נראה גם כי שרת הפרוקסי פותח מספר חיבורי TCP מול השרת של אתר האינטרנט כי ניתן לראות כי מתבצע תהליך לחיצת הידיים מול השרת של אתר האיטרנט וכי ישנם כמה פורטים כלומר כמה סוקטים שהשרת פתח כלומר כמה חיבורים שונים ששרת הפרוקסי פתח מול שרת האתר על מנת לקבל את הקבצים הדרושים של האתר כפי שניתן לראות בתמונה הבאה::

No.	Time	Source	Destination	Protocol	Length	Info
453	7.477897746	10.0.2.4	199.203.54.24	TCP	74	39976 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2254997578 TSecr=0 WS=12
454	7.520711144	8.8.8.8	10.0.2.4	DNS	124	Standard query response 0xe817 A a1887.dscq.akamai.net A 104.77.202.64 A 81.218.31.145 OPT
455	7.520712260	8.8.8.8	10.0.2.4	DNS	148	Standard query response 0x11bc AAAA a1887.dscq.akamai.net AAAA 2001:4cd0:dc00:1::684d:ca23
456	7.521523265	10.0.2.4	104.77.202.64	TCP	74	58902 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3012749591 TSecr=0 WS=12
457	7.542546606	199.203.54.24	10.0.2.4	HTTP	493	HTTP/1.1 200 OK (GIF89a)
458	7.542547038	199.203.54.24	10.0.2.4	TCP	60	80 → 39962 [SYN, ACK] Seq=0 Ack=1 Win=32768 Len=0 MSS=1460
459	7.542547062	199.203.54.24	10.0.2.4	TCP	60	80 → 39976 [SYN, ACK] Seq=0 Ack=1 Win=32768 Len=0 MSS=1460
460	7.542547085	104.77.202.64	10.0.2.4	TCP	60	80 → 58902 [SYN, ACK] Seq=0 Ack=1 Win=32768 Len=0 MSS=1460
461	7.542584014	10.0.2.4	199.203.54.24	TCP	54	39958 → 80 [ACK] Seq=796 Ack=14629 Win=63480 Len=0
462	7.542696712	10.0.2.4	199.203.54.24	TCP	54	39962 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
463	7.542746096	10.0.2.4	199.203.54.24	TCP	54	39976 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
464	7.542794861	10.0.2.4	104.77.202.64	TCP	54	58902 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
465	7.543103987	10.0.2.4	10.0.2.5	HTTP	555	HTTP/1.1 200 OK
466	7.543358874	10.0.2.4	199.203.54.24	HTTP	466	GET /images/logo-back.gif HTTP/1.1
467	7.543591516	10.0.2.4	10.0.2.5	HTTP	109	Continuation (GIF89a)
468	7.543657726	10.0.2.4	199.203.54.24	HTTP	469	GET /images/eitan_square.gif HTTP/1.1
469	7.543666517	10.0.2.5	10.0.2.4	TCP	66	49870 → 3128 [ACK] Seq=997 Ack=16396 Win=64128 Len=0 TSval=4039308682 TSecr=3721629463
470	7.543704205	10.0.2.4	104.77.202.64	HTTP	450	POST / HTTP/1.1
471	7.544147936	10.0.2.5	10.0.2.4	TCP	66	49870 → 3128 [ACK] Seq=997 Ack=16439 Win=64128 Len=0 TSval=4039308682 TSecr=3721629464
472	7.544219749	10.0.2.4	104.77.202.64	HTTP	139	Continuation
473	7.544459150	104.77.202.64	10.0.2.4	TCP	60	80 → 58902 [ACK] Seq=1 Ack=482 Win=32287 Len=0
474	7.613178505	199.203.54.24	10.0.2.4	HTTP	1514	HTTP/1.1 200 OK (GIF89a)[Malformed Packet]
475	7.613209833	10.0.2.4	199.203.54.24	TCP	54	39950 → 80 [ACK] Seq=803 Ack=8821 Win=62780 Len=0
476	7.613526032	199.203.54.24	10.0.2.4	HTTP	1361	Continuation
477	7.613533538	10.0.2.4	199.203.54.24	TCP	54	39950 → 80 [ACK] Seq=803 Ack=10128 Win=62780 Len=0
478	7.613739848	10.0.2.4	10.0.2.5	HTTP	558	HTTP/1.1 200 OK
479	7.613879888	10.0.2.4	10.0.2.5	HTTP	2434	Continuation
480	7.613945638	199.203.54.24	10.0.2.4	HTTP	1514	HTTP/1.1 200 OK (JPEG JFIF image)[Malformed Packet]
481	7.613951369	10.0.2.4	199.203.54.24	TCP	54	39944 → 80 [ACK] Seq=816 Ack=9383 Win=63480 Len=0
482	7.614255363	10.0.2.4	10.0.2.5	HTTP	561	HTTP/1.1 200 OK
483	7.614332494	10.0.2.4	10.0.2.5	HTTP	1124	Continuation
484	7.614521974	199.203.54.24	10.0.2.4	HTTP	1354	Continuation
485	7.614526848	10.0.2.4	199.203.54.24	TCP	54	39944 → 80 [ACK] Seq=816 Ack=10683 Win=63480 Len=0
486	7.614789098	10.0.2.4	10.0.2.5	HTTP	1366	Continuation