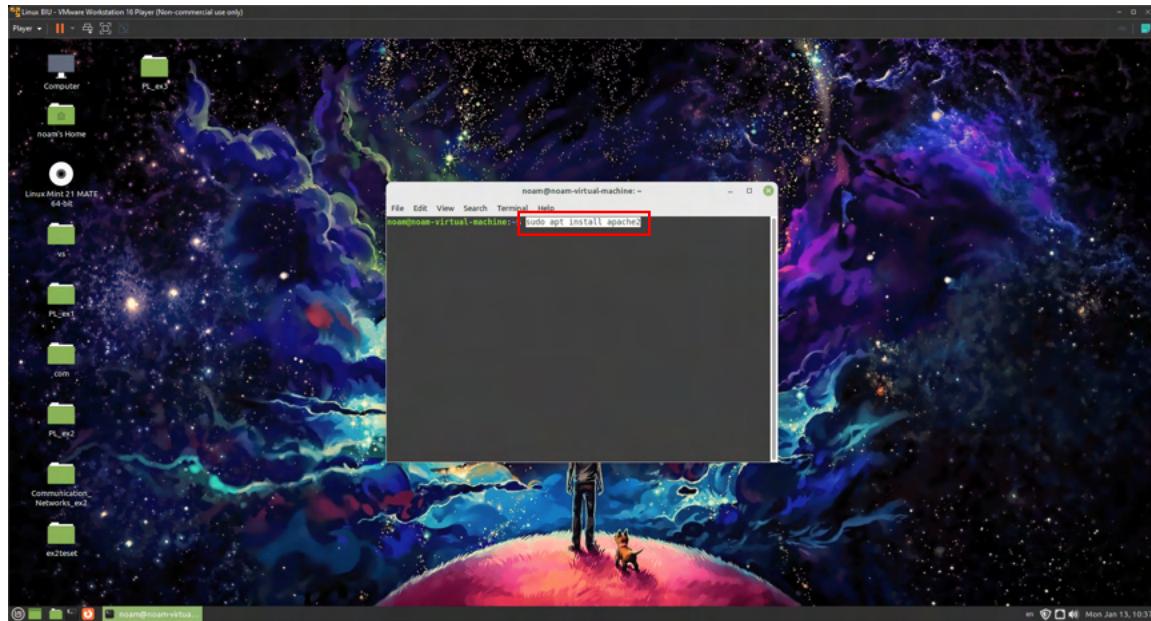


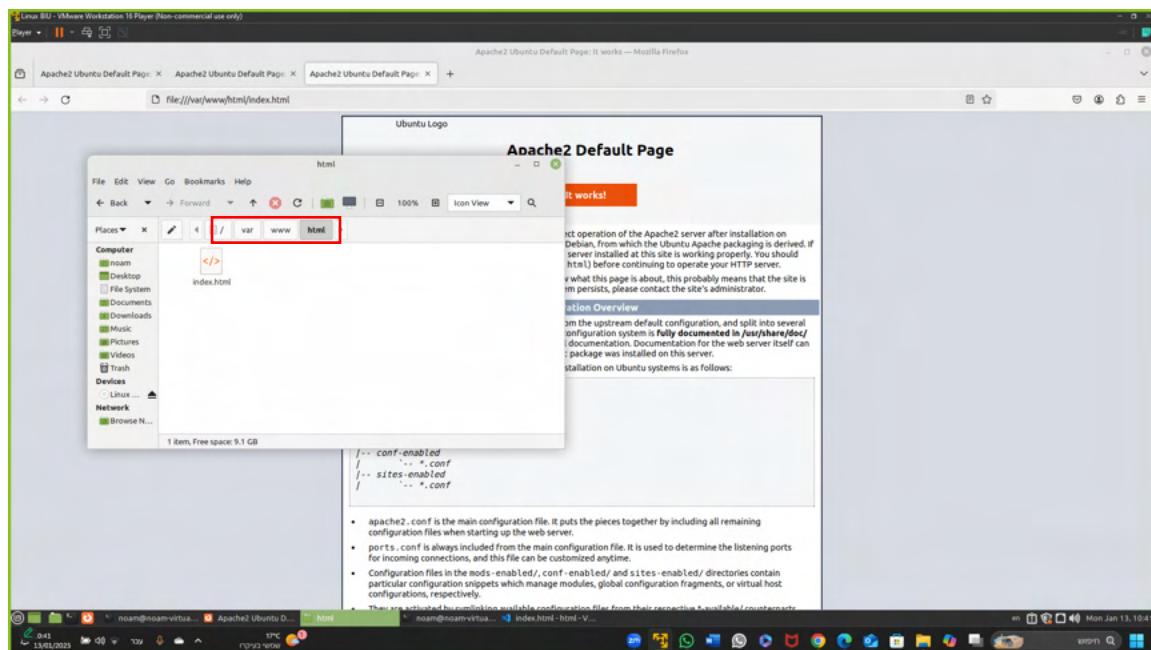
## מטלה מס' 3 – אילאי בן יהושע ונועם ליבוביץ

### חלק ראשון

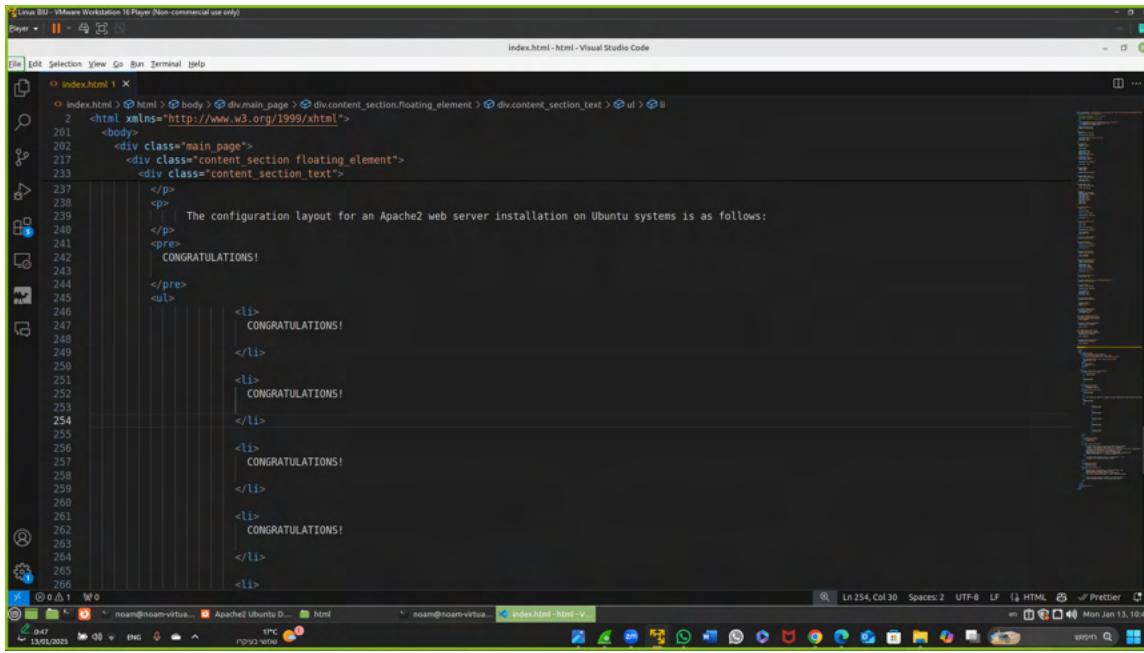
נתקין את שרת האינטרנט – apache.



נראה את קובץ html הדיפולטיבי בנתיב – .var/www/html –



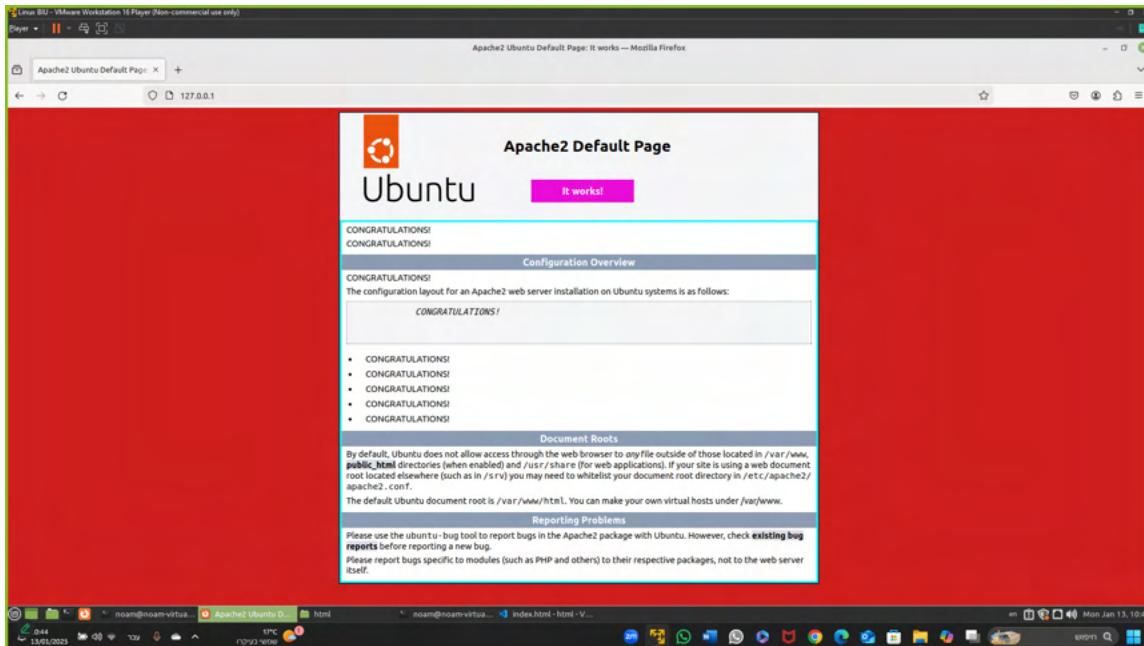
נפתח אותו בערוך הטקסטים vscode ונבצע בו מס' שינויים גרפיים וטקסטואליים.



The screenshot shows the Visual Studio Code interface with the file 'index.html' open. The code contains several identical entries: 'CONGRATULATIONS!' repeated five times, each enclosed in a list item (`<li>`). The code is as follows:

```
1 <html xmlns="http://www.w3.org/1999/xhtml">
2   <body>
3     <div class="main_page">
4       <div class="content_section_floating_element">
5         <div class="content_section_text">
6           <p>
7             The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:
8           </p>
9           <pre>
10          CONGRATULATIONS!
11        </pre>
12        <ul>
13          <li>
14            CONGRATULATIONS!
15          </li>
16          <li>
17            CONGRATULATIONS!
18          </li>
19          <li>
20            CONGRATULATIONS!
21          </li>
22          <li>
23            CONGRATULATIONS!
24          </li>
25        <ul>
26      </div>
27    </div>
28  </body>
29</html>
```

נשים לי כי השינויים מוצגים כתוב דף הדיפולטיי אותו מחזיר השרת לאחר רענון של העמוד.



נפתח את תוכנת wireshark (מעכשיו תקרא בהמשך הטופס 'כרייש'), ונסניף בעזרתיה את התעבורה. נסנן כך שיזגגו חבילות הרלוונטיות לפורטוקול TCP ולפורט 80 עליו רץ שרת האינטרנט שלנו.

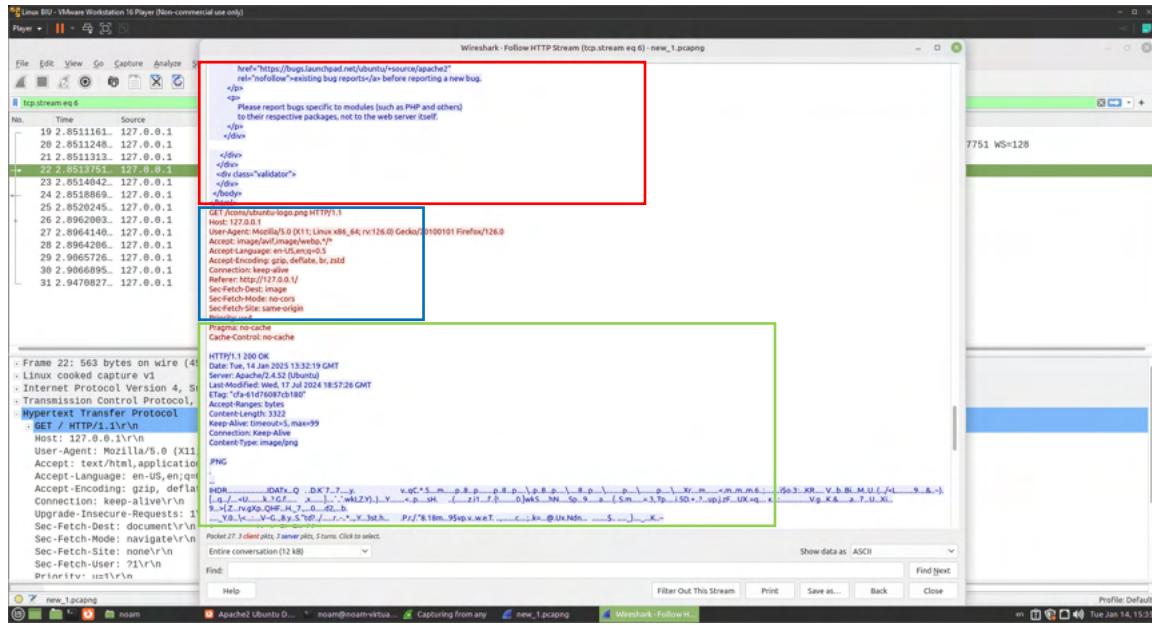
The screenshots show the following sequence of events:

- Frame 22:** 563 bytes on wire (4504 bits), 563 bytes captured (4504 bits) on interface any, id 0
  - Linux cooked capture v1
  - Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.1
  - Transmission Control Protocol, Src Port: 39876, Dst Port: 80, Seq: 1, Ack: 1, Len: 495
    - HTTP/1.1 200 OK [text/html]
- Frame 23:** 7026 bytes on wire (56648 bits), 7026 bytes captured (56648 bits) on interface any, id 1
  - User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:12.0) Gecko/20100101 Firefox/12.0
  - Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,\*/\*;q=0.8
  - 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
  - DNT: 1
- Frame 27:** 39876 bytes on wire (31904 bits), 39876 bytes captured (31904 bits) on interface any, id 2
  - HTTP/1.1 200 OK [text/html]
  - Content-encoding: gzip
  - Content-length: 2080
  - File data: 7026 bytes
  - Line-based Text data: text/html (316 lines)
 

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<!-->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Apache2 Ubuntu Default Page: It works</title>
<style type="text/css" media="screen"></style>
<!-->
<margin: 0px 0px 0px 0px;-->
```

סביר את הפלואו של הדברים בתמונות לעיל. תחילת אנו רואים לחיצת ידיים של get http&syn&ack-&ack לשרת. לאחר מכן הלקוח שולח בקשה http מסוג לשרת על מנת לקבל את "דף הבית" ("הלו הוא"). השרת מחזיר ללקוח ack כדי לתת חיוי שacky בקשה זו נקלטת אכן, ובוודעת הקובבת לנו הוא מחזיר לו את העמוד כאשר סטטוס ההודעה הינו ok, כלומר הכל טוב. מה שקרה לאחר מכן הלקוח החל לקרוא את העמוד וראה שהוא זמין למשאב נוסף – ubuntu-logo.png, אז הוא ביצע על גבי אותו חיבור (שכן אנחנו

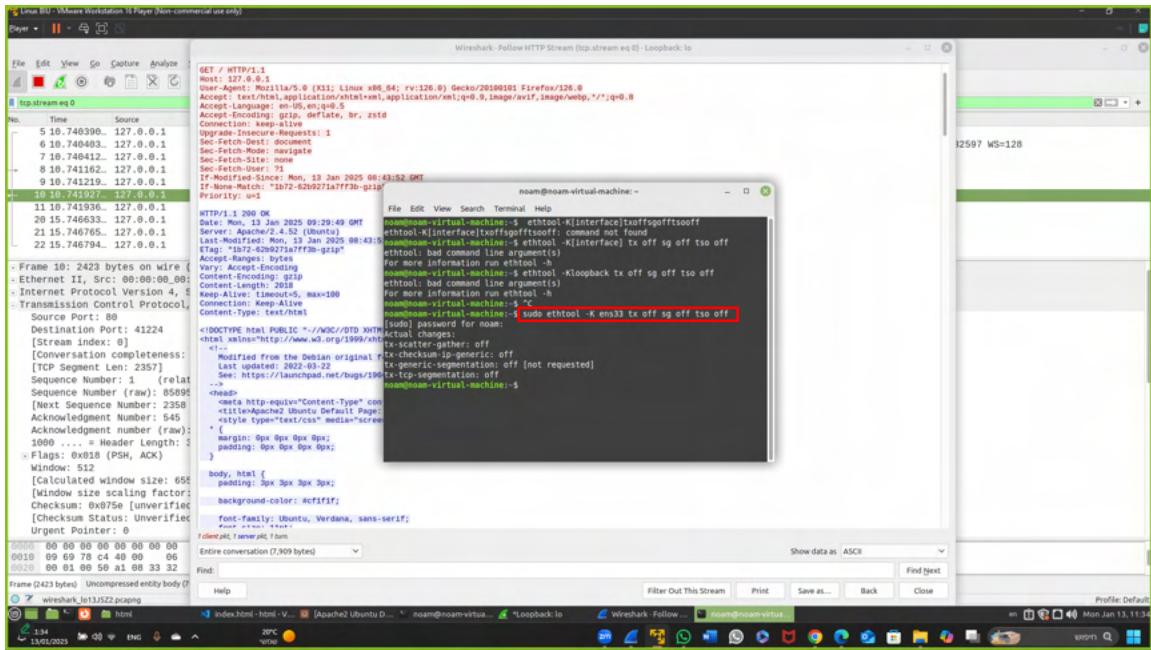
בגרסה 1.1 והוֹא (persistent) בקשה נוספת מהשרת ע"מ קיבל את התמונה הדרישה. השרת משיב בAO 200 עם תוכן התמונה, ואז הלוקוח מבצע בקשה נוספת (שוב על גבי אותו חיבור). עברו משאב מסוג favicon, אך הפעם השרת לא מוצאתו ועונה ללקוח בFOUND 404.



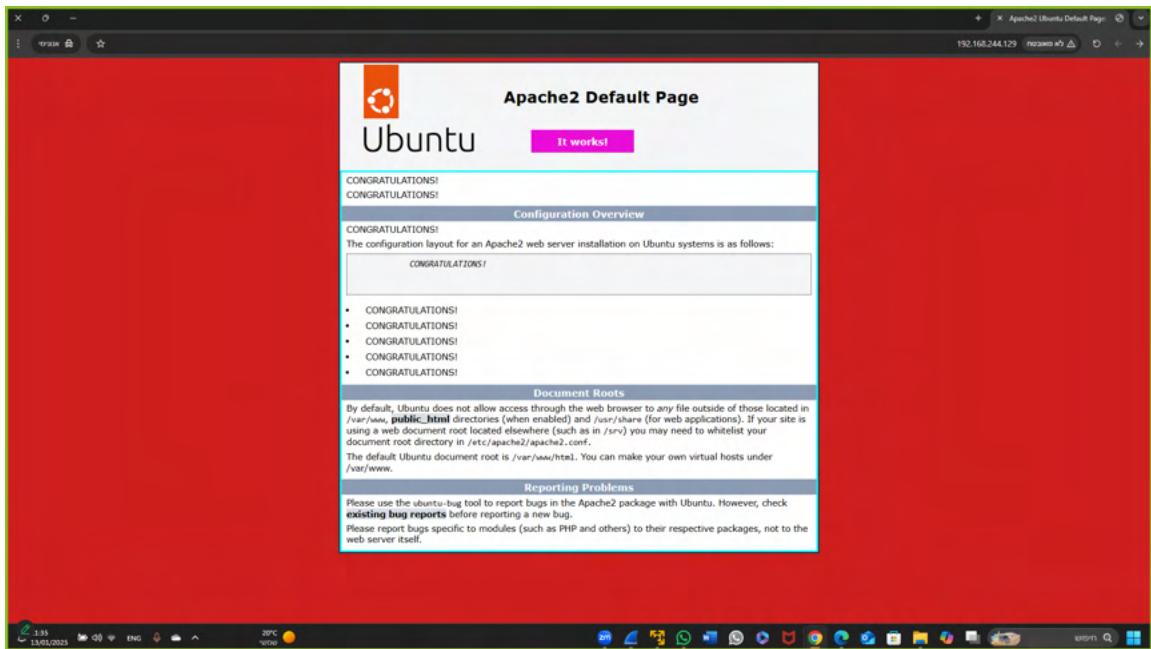
התמונה לעיל אנו רואים את הפלואו של תקשורת/http שקרתה, כאשר **באדום** אנו רואים את סוף תוכן העמוד שהשרת החזיר בבקשת הראשונה ללקוח, **בכחול** אנו רואים את הבקשת הנוספת של הלוקוח עבורי התמונה, **וביירח** את התשובה של השרת עם הקידוד שלה.

נסכם – זה "c" התקשרות לכל אורכה הتبכעה על גבי חיבור ייחד שהוקם בין השרת ללקוח, אשר דרכו עברו 3 בקשות. הראשונה החזירה קובץ html, השנייה החזירה תוכן **מיוחד** שר תמונה בפורמט jpg, והשלישית החזירה header ללא תוכן, שכן המשאב לא נמצא ע"י השרת. נשים לב שאורכי התכנים שמועברים בהודעות מצוינים בהאדרים, כך שלדוגמא בתמונה הראשונה אנו רואים כי אורך התוכן (לאחר דחיסתו) הינו 2018, וזה כמובן ק"יימ" בכל אחת מההודעות.

## כutt נבטל את מנגנון segmentation offload

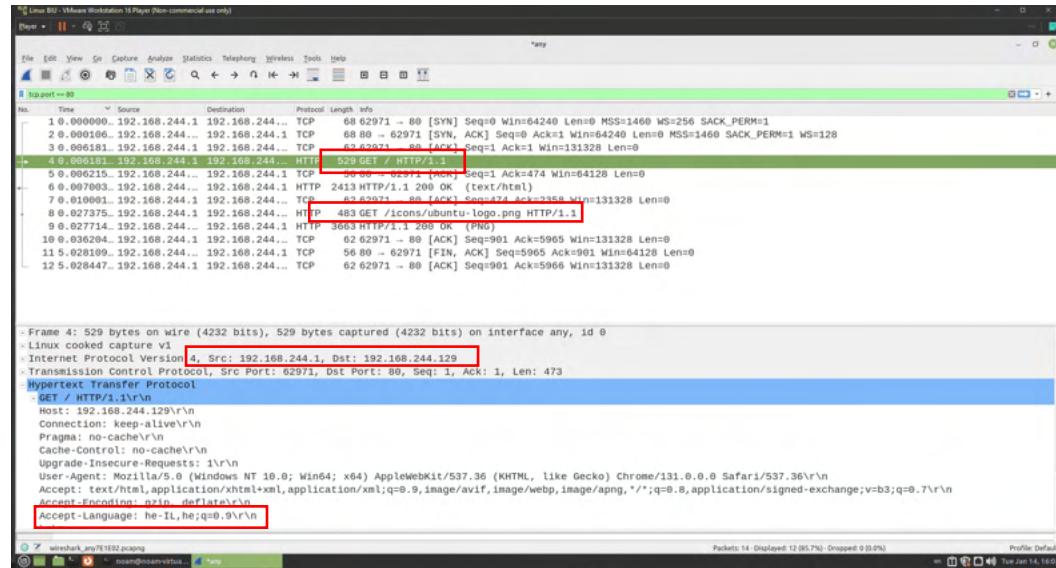


נגלש לשרת מהמחשב **השי** דרך מצב incognito וכן דרך מצב דפדפן 'רגיל'. ראשית נראה שהה אקע עובד.

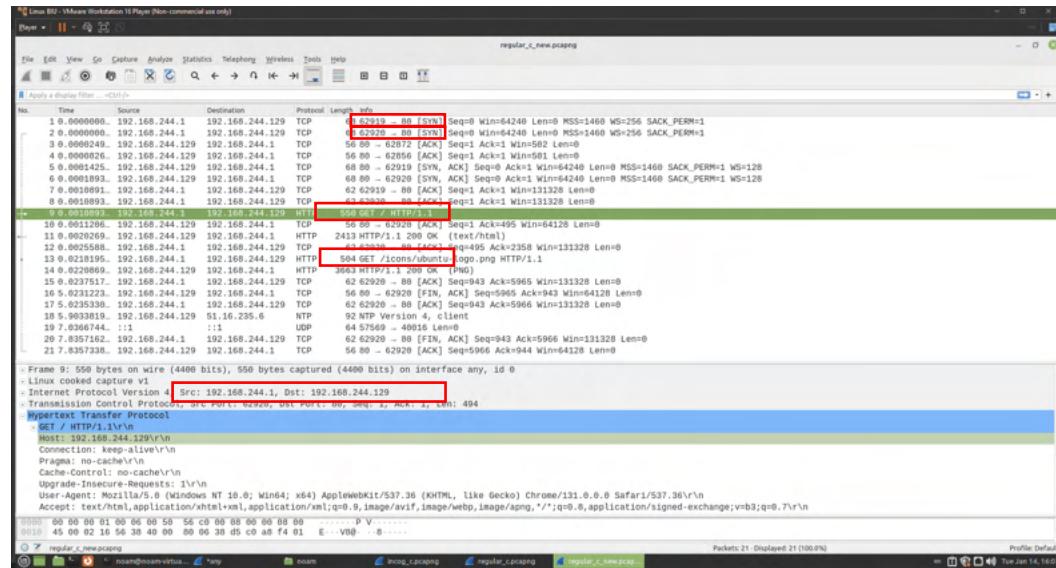


עובד.

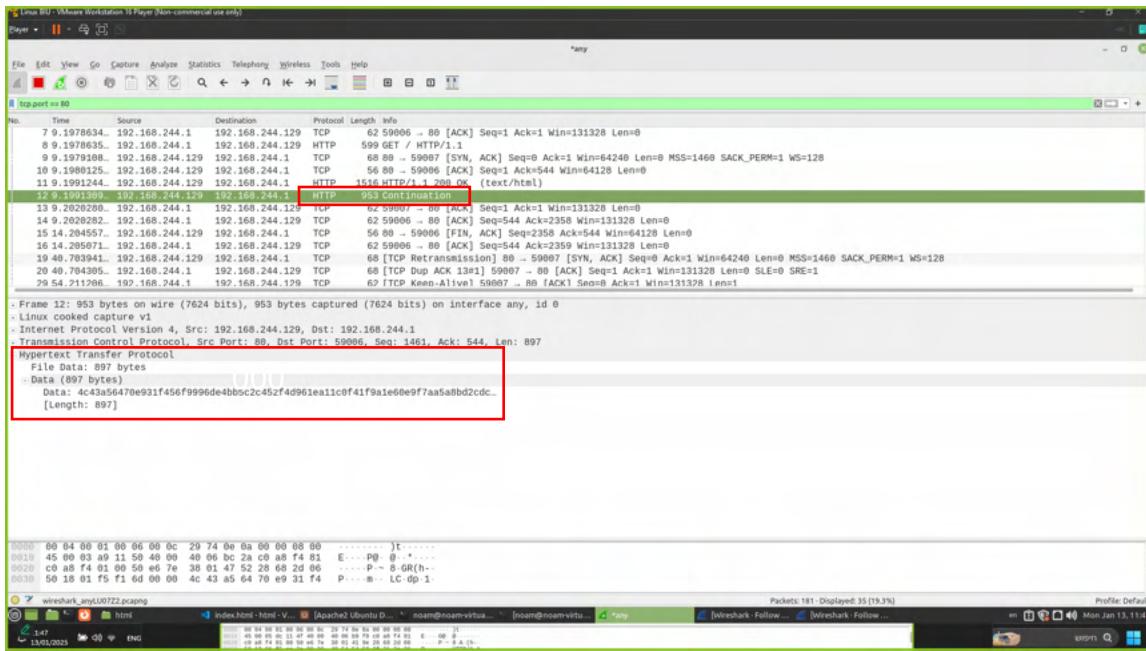
נבחן בתבורה דרך הכריש. נתחל דוקא בגלישה דרך מצב incognito. נשים לב שקיבלנו משחו מאוד זהה לגלישה דרך המחשב עליו רץ השירות. גם פה ישן 2 בקשות אל השירות, הראשונה עברו עמוד html הדיפולטי של השירות, והשנייה עברו התמונה שהעמוד מציג (בדומה למה שראינו קודם לכן). אולם נשים לב שהפעם לא מטבחעת פניה שלישית עברו משאב החוסר, favicon, אלא יש לאחר קבלת התמונה מהשרת הלקוח סוגר את החיבור מול השירות. נשים לב שהכל מתבצע על גבי חיבור יחיד בין הלקוח לשרת. הבדל טריומיAli הוא שהפעם התקשרות מטבחעת בין 2 כתובות זו וזו הגינוי שכן השירות והלקוח רצים על 2 מחשבים שונים), בשונה מהדוג' הקודמת. שם השתמשנו ב-127.0.0.1.1.



כעת נבחן במס' שינויים כאשר ניגש לשרת דרך הדפסן במצב הרגיל. תחילתה נציג כי גם הוא מבקש 2 משאבים בלבד, אך הפעם הוא פותח חיבור נוסף, עם זאת הוא לא משתמש בו כלל לאורך הפלאו של התקשרות. ראיינו כבר בתרגיל קודם כי זהו מנגן על מנת שלדפסן יהיה מס' חיבורים זמינים בעת הצורך על מנת לבצע מס' משאבים במקביל. דבר מינורי שראיינו הוא שבheader של השפה בסיטו incognito יש רק עברית, ואילו ברגיל מצויין שם גם כן אנגלית – en. מעבר לכך לא הבחנו בהבדלים נוספים בין מצב הדרפסן.



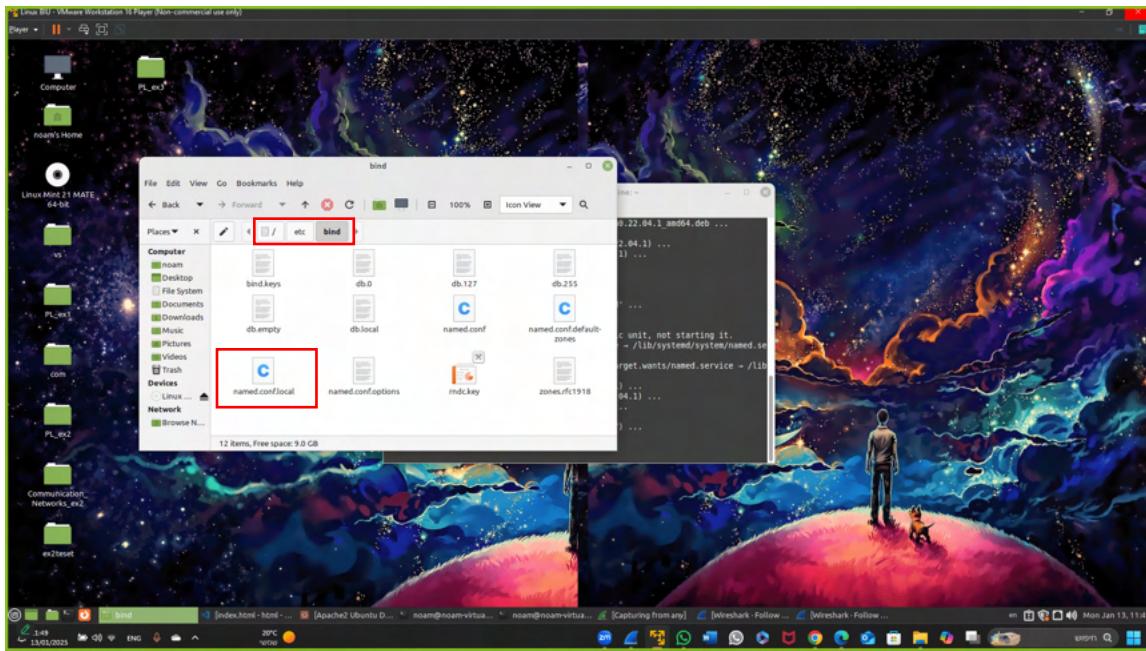
נבטל בהגדורות את הkonfiguracija לגביה איחוד חבילות tcp תחת http ו נראה שאק עכשו אנו רואים את שכבת האפליקציה פר-חביבה, שלא כמו שהוא קודם לכן.



עד כאן חלק ראשון

## חלק שני

לאחר שהתקנו את שרת DNS, נמצא את הקובץ `.named.conf.options`, נמצא את הקובץ



נגדיר את שרת DNS שלנו כך שכל פעם שתגיע אלינו שאלת DNS, הוא יעביר אותה לשרת DNS של גוגל (8.8.8.8 או 8.8.4.4). לאחר מכן, קיבל את התשובה חזרה מגוגל, ישמר אותה בקاش ואז יחזיר את התשובה למי ששאל אותה (עובד כמו ריזולבר).

A screenshot of a terminal window displaying the contents of the 'named.conf.options' file. The file is located at '/etc/bind/named.conf.options'. The code is as follows:

```
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.

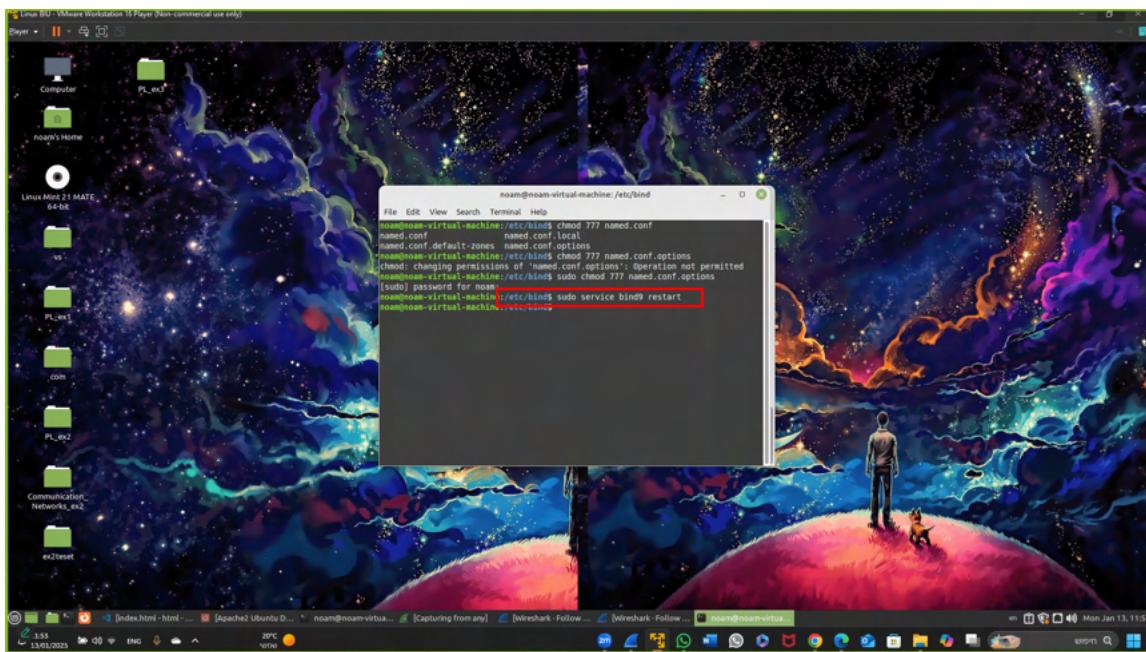
    forwarders {
        8.8.8.8;
        8.8.4.4;
    };

    // 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;

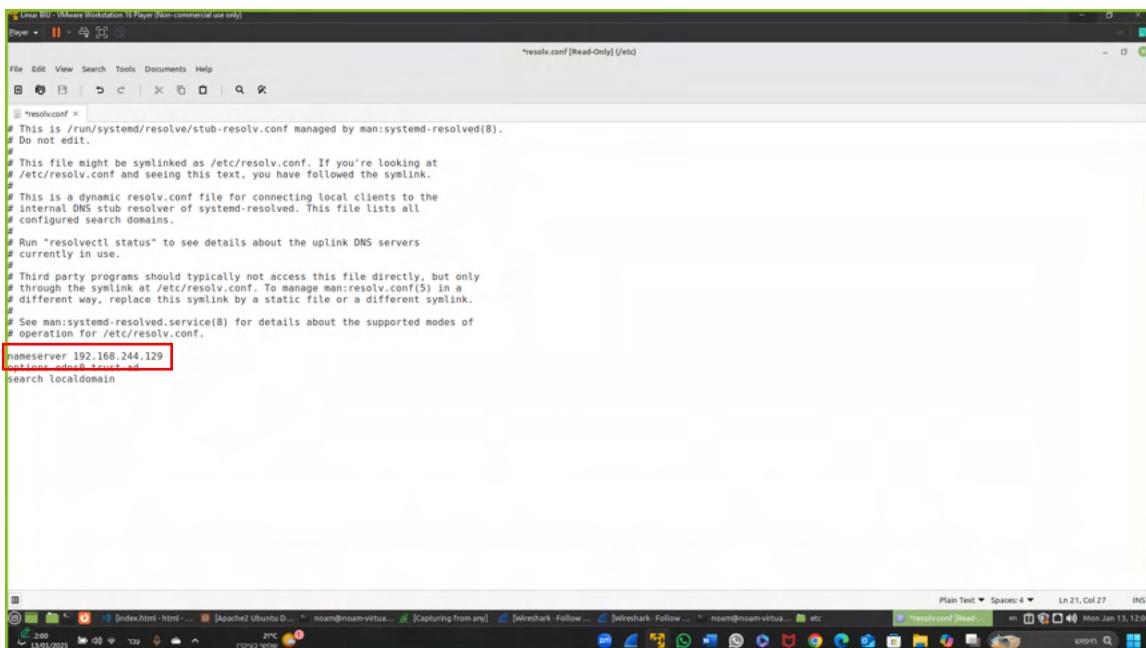
    listen-on-v6 { any; };
};
```

The lines 'forwarders' and 'dnssec-validation no;' are highlighted with red boxes.

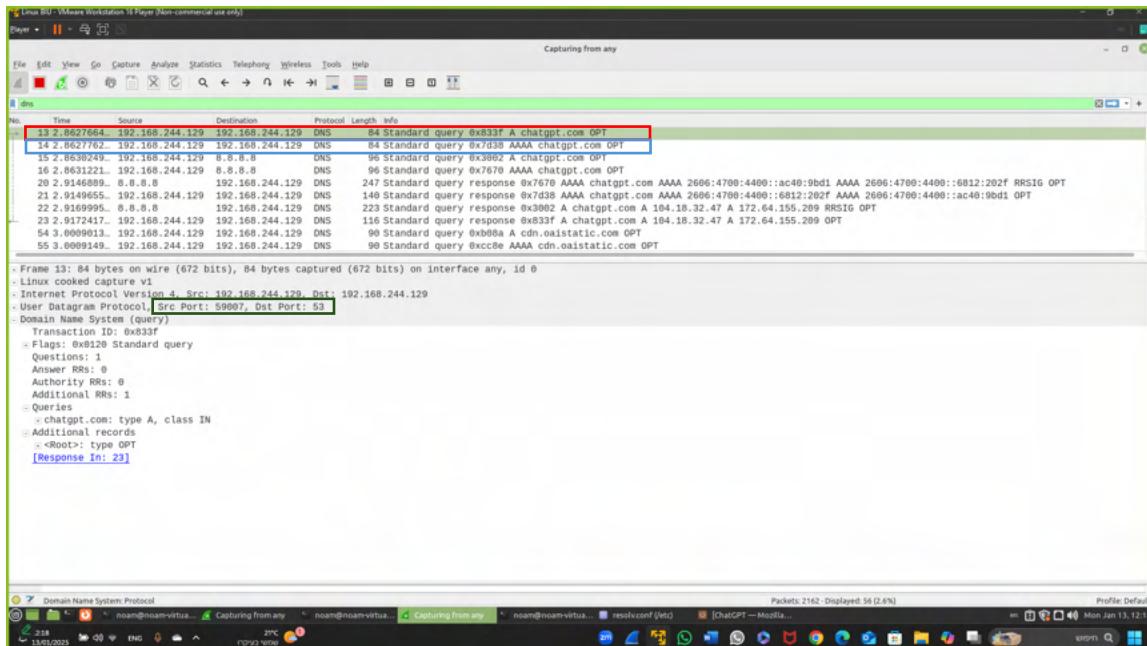
נשמר את השינויים ונתחל את השרת כך שהשינויים יכנסו לתוכף.



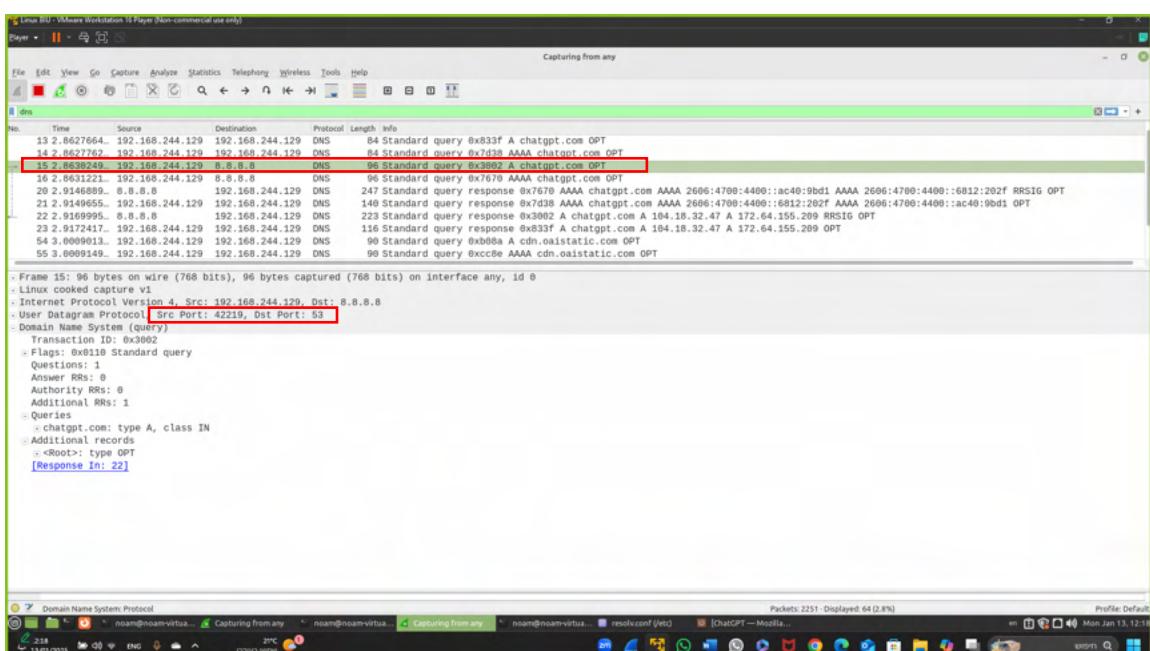
כעת, נגדיר את המחשב לעבור מול שרת DNS שהגדכנו באופן דיפולטיבי.



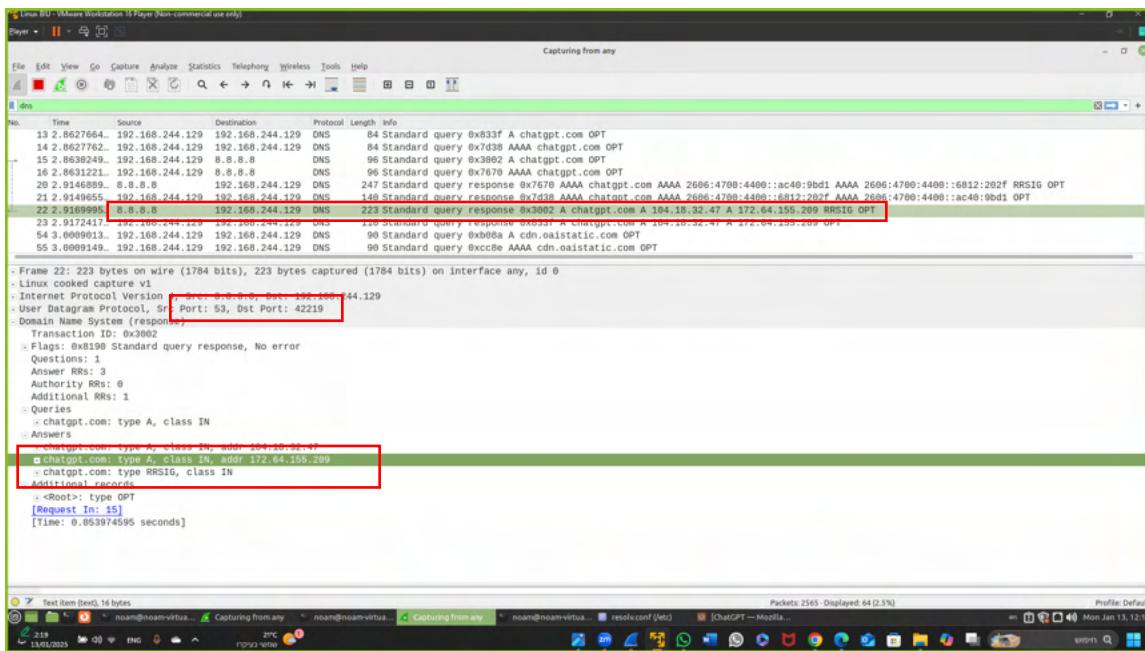
מבצע פעולה המצריכה ביצוע DNS, נגלוש לאתר של **GPT**. ניתן לראות בכריש את בקשת DNS של הלוקו (הלווא הוא אנחנו), כאשר הלוקו מבקש את כתובות הIp של האתר פעם אחת בפורמט של 6vk4. **באחד** ומיד **לאח"כ** פעם נוספת בפורמט של 6vk4. **בכחו**. נשים לב מבון שפורט היעד הוא 53, הרו שהוא מוקצה לשאלות DNS. (הערה – נשים לב שבכל פעם מעכשו בהקשרים של DNS תמיד תבוצען 2 שאלות – 1 עבור 6vk4 ומיד לאח"כ תישלח שאלתה עבור 6vk4). אנו נתיחס בהסבירים לפלאו של 6vk4, 6vk4 שקול בהקשר של ההסבירים שנთאר).



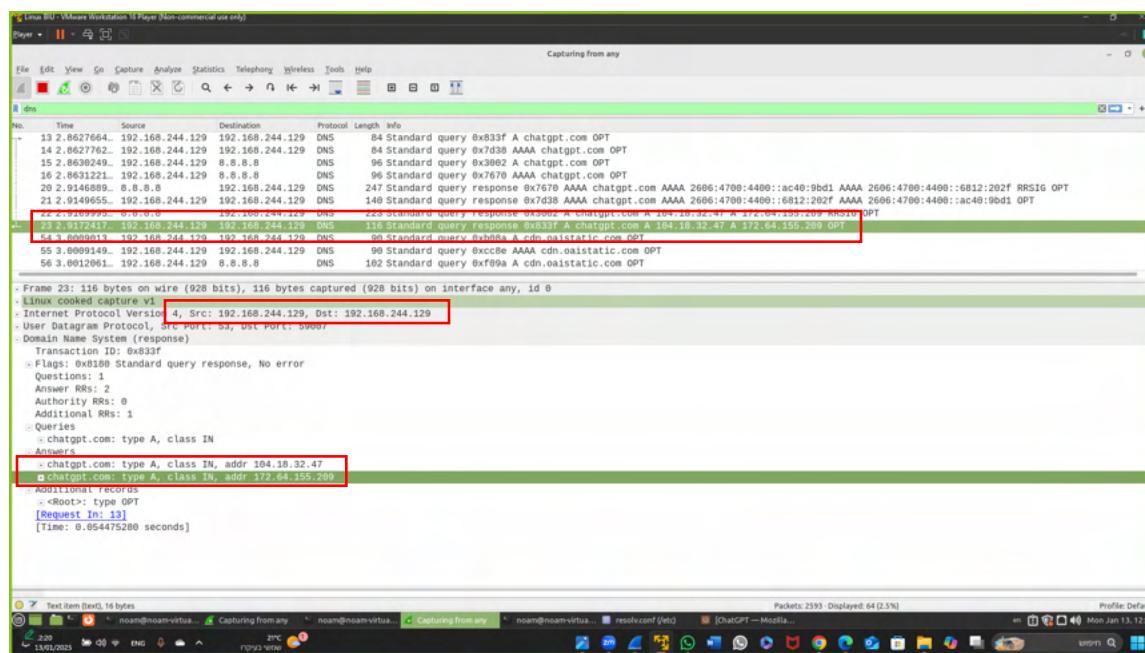
**8.8.8.8** נשים לב שאכן לאחר שרתת DNS שלנו מקבל את הבקשה הוא נעזר בשרת של גול (בעזרת חיבור נפרד מזה שהוא לו עם הלוקו, שכן הוא פונה למפורט 42219 למפורט 53 בשרת של גול) כדי להחזיר את התשובה (האתר שביקשנו). השרת שלנו כאמור מבקש זאת בעזרת 4IPV ולאחר מכן בעזרת IPV6.



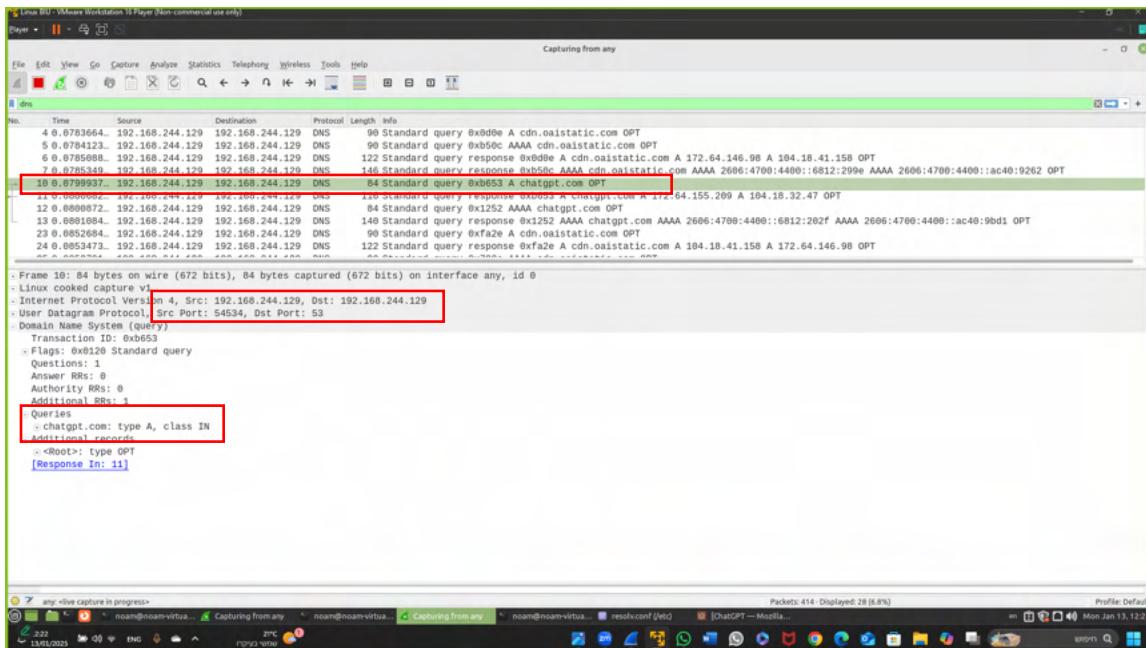
כאן, השרת של גוגל מוחזיר אל השרת שלנו את התשובה עם כתובת IP של האתר שביקשנו. זהו הכתובת הסופית שכן השרת של גוגל שלח לנו רשומה מסוג A.



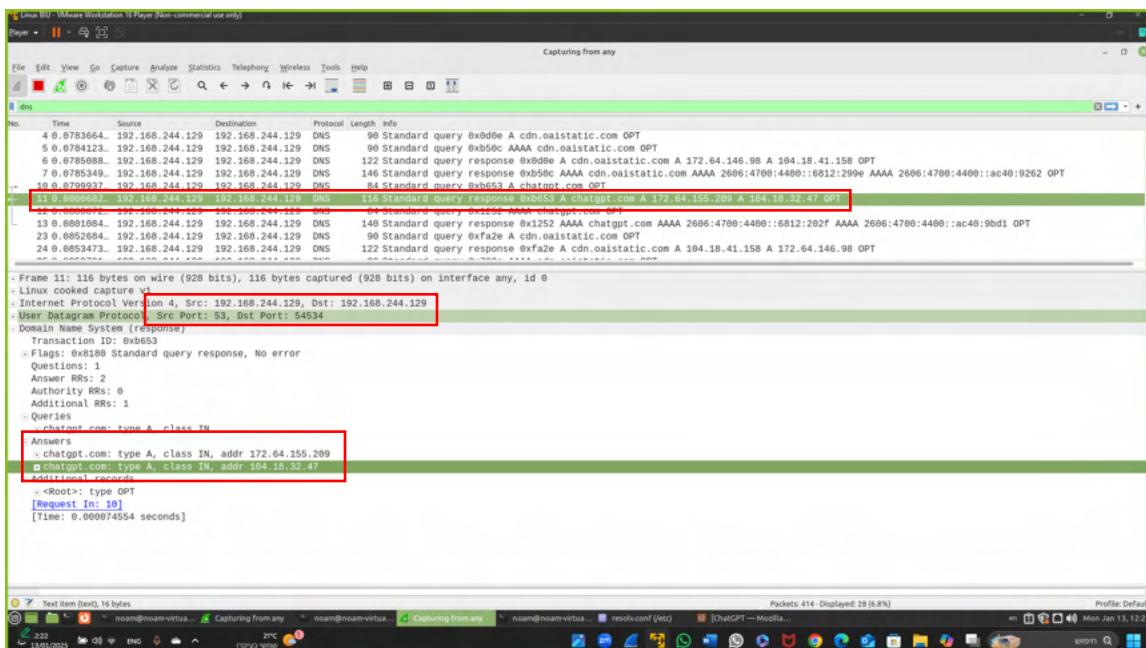
השרת שלנו, שלח אלינו (להלן) בחזרה את התשובה שקיבל מהשרת של גוגל. נראה כי בתשובה שקיבלנו, קיבלנו שתי אופציות לIP של האתר שביקשנו, יכול להיות מכמה סיבות שאת חלון הזכרנו בהרצאה למשל חלוקת עומסים, או חברה חיצונית המספקת שירות DNS.



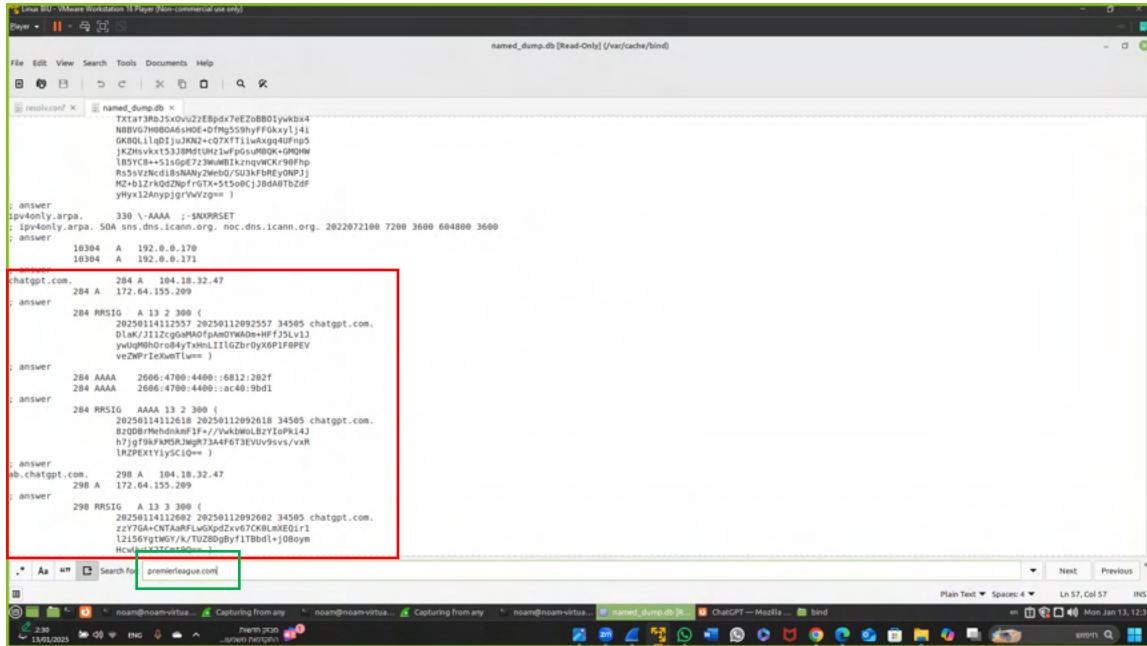
כעת, ננסה לגשת שוב פעם אל האתר (מתוך תקוויה שכתובתו תהיה כבר בקאס').



נשים לב שבהודעה העוקבת מיד השרת החזיר ללקוח את התשובה (ה- `dst` וה- `src` הטעפכו). בשונה מפעם קודמת לא היה שלב ביןים של פניה לשרת DNS של גוגל זאת כתוצאה מהעובדת שהשרת שלנו שמר את הכתובת כבר בקאס של עוד מהבקשתה הקודמת, ולכן אין לנו צורך בבקשתם החיצונית שוב. סה"כ קיבלנו באמצעות את 2 הכתובות של האתר שבקישמו (בדיקה כמו הניסיון הקודם).



ניתן את הרשומות השמורות בקובץ בשם named\_dump.db (באמצעות ביצוע הפקודה sudo rndc dumpdb -cache -cache), ונכנס אל הקובץ שיצרנו. נבחן כי אכן הדומיין (האתר של צאט PGT) שchipshno נמצא בו.



```

Linux BU - VMware Workstation 14 Player (Non-commercial use only)
File Edit View Search Tools Documents Help
named_dump.db [Read-Only] (/var/cache/bind)

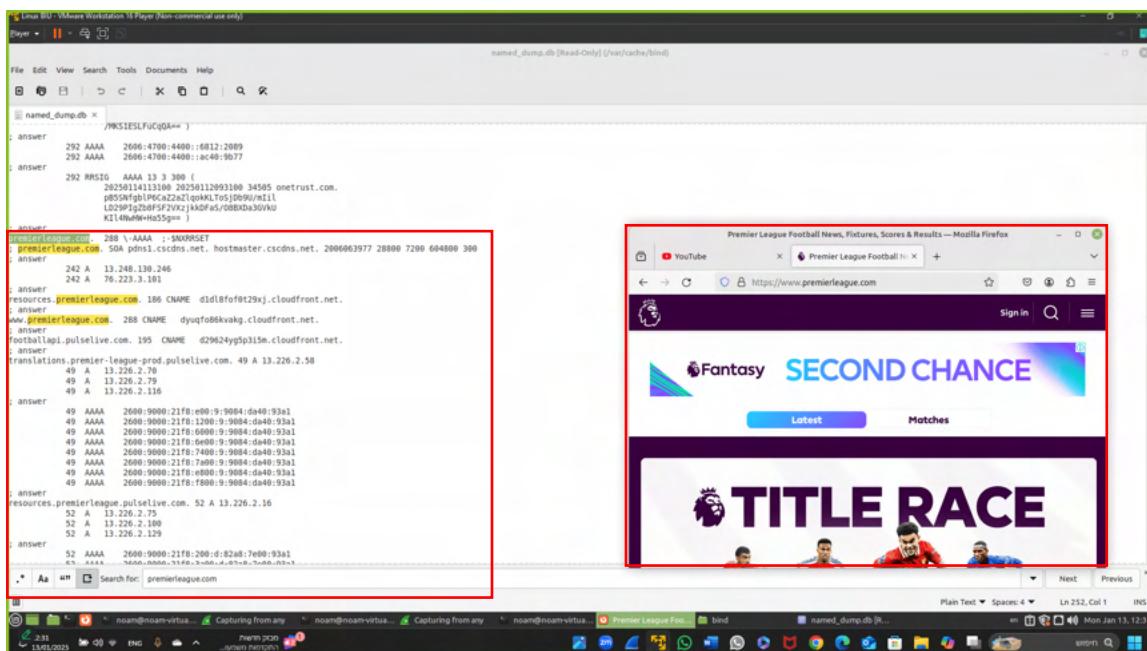
; search for: premierleague.com
TXXB0V7H0B0A6H0B7D7E20BBD1YwKkx4
NBBV07H0B0A6H0B7D7E20BBD1YwKkx4
GKBR0L1lDp1juXN0X+D07KXTtluuAxqg4UfUp5
JKZHiVxkvt5J3B9e1UH2lWPF6s5M0Dk-GH0W
L0BbPmIa34Xjv9X6hCQr7OzV3F6d70yq8nmp
R55y52v2nd1s5MANy2Web0z5D3KFbR5y0NP2f
M2-b1ZrKd0ZnpfrGTX+515oOCj2BdA#0Tb2df
yHpx12AnypjgrVvVzq==

; answer
; ip4only.arpa.      330 \.AAAAA. : $NOVERSE
; ip4only.arpa. 50A.sns.dns.icann.org.noc.dns.icann.org. 2022072100 7200 3600 604800 3600
; answer
    10304 A   192.0.0.178
    10304 A   192.0.0.171

chatgpt.com. 284 A   104.18.32.47
    284 A   172.64.155.209
; answer
    284 RRSIG A 13 2 300 (
        20250114112557 20250112092557 34505 chatgpt.com.
        DlAkJ11Zcg0MaOfpAm0YWA0s-HFf15LV1
        ywQghBn0ro84yTxHn11L6Zbr0yXkPfI0EV
        vEc2Hr1OkmttLunm )
; answer
    284 AAAA 2606:4700:4400::812:102f
    284 AAAA 2606:4700:4400::c40:9bd1
; answer
    284 RRSIG AAAA 13 2 300 (
        20250114112613 20250112092603 34505 chatgpt.com.
        B22304C97547FUoipdZxe67CK8axEQi1
        H1gj9M4hMSRJeqg444F673EVu9ssv/xAr
        lR2PEXTYlySCIo== )
; answer
ab.chatgpt.com. 298 A   104.18.32.47
    298 A   172.64.155.209
; answer
    298 RRSIG A 13 3 300 (
        20250114112604 20250112092603 34505 chatgpt.com.
        227475C97547FUoipdZxe67CK8axEQi1
        L2156YgTWQVx/TUZ80p8yITBbd1+j08yn
        HCwvksx24s6
; answer
; Aa ;w Search for: premierleague.com

```

נודא שכן הדומיין שמחפשים נכנס אל הקוש. נשים לב כי בתחתית המסךChipshno את כתובת האתר של הליגה האנגלית ווידנו שהוא לא קיים כרגע בקשר. כתע נגלוש אל אתר הליגה האנגלית ולאחר מכן ניצא שוב את הקובץ ע"פ הפקודה שציינו קודם. ונגלה שכן הדומיין נשמר בקשר של השירות שלנו.



```

Linux BU - VMware Workstation 14 Player (Non-commercial use only)
File Edit View Search Tools Documents Help
named_dump.db [Read-Only] (/var/cache/bind)

; search for: premierleague.com
/WKS1ESLFuQzQ==

; answer
    292 AAAA 2606:4700:4400::6812:9077
    292 AAAA 2606:4700:4400::ac40:9077
; answer
    292 RRSIG AAAA 13 3 300 (
        20250114113108 20250112093108 34505 onetrust.com.
        p85SMfbglPMCn2azLqokULTo5D9WU/m1L
        LD29P1gZBfSPV92jkK0fAvs5OBXDa30VKU
        K214mWuHnSgus )
; answer
premierleague.com. 288 \.AAAAA. : $NOVERSE
premierleague.com. 50A.pdns1.cscdns.net. hostmaster.cscdns.net. 2006063977 28800 7200 604800 300
; answer
    242 A   13.246.130.246
    242 A   76.223.3.101
; answer
resources.premierleague.com. 186 CNAME d1d18f0f0t29xj.cloudfront.net.
; answer
www.premierleague.com. 288 CNAME dyugfo8kvakg.cloudfront.net.
; answer
FootballLapi.pulselive.com. 195 CNAME d219824yp3p15m.cloudfront.net.
; answer
translations.premier-league-prod.pulselive.com. 49 A 13.226.2.58
    49 A   13.226.2.78
    49 A   13.226.2.79
    49 A   13.226.2.116
; answer
    49 AAAA 2606:9000:217B:e09:9:9084:da40:93a1
    49 AAAA 2606:9000:217B:120B:9:9084:da40:93a1
    49 AAAA 2606:9000:217B:6e98:9:9084:da40:93a1
    49 AAAA 2606:9000:217B:6e99:9:9084:da40:93a1
    49 AAAA 2606:9000:217B:740B:9:9084:da40:93a1
    49 AAAA 2606:9000:217B:7a0B:9:9084:da40:93a1
    49 AAAA 2606:9000:217B:7a0C:9:9084:da40:93a1
    49 AAAA 2606:9000:217B:780B:9:9084:da40:93a1
; answer
resources.premierleague.pulselive.com. 52 A 13.226.2.16
    52 A   13.226.2.75
    52 A   13.226.2.100
    52 A   13.226.2.129
; answer
    52 AAAA 2606:9000:217B:200:d:82d8:7e00:93a1
    52 AAAA 2606:9000:217B:200:d:82d8:7e00:93a1
    52 AAAA 2606:9000:217B:200:d:82d8:7e00:93a1
    52 AAAA 2606:9000:217B:200:d:82d8:7e00:93a1
    52 AAAA 2606:9000:217B:200:d:82d8:7e00:93a1
; answer
; Aa ;w Search for: premierleague.com

```

כעת, נגדיר את שרת DNS שלנו להיות אוטורטיבי. נכנס אל הקובץ `/etc/bind/named.conf.local` ונגדיר לשרת קובץ שיכיל את ניהול של הדומיין `biu.ac.il` (במלבן האדום). ניצור את הקובץ בעזרת תבנית קיימת שנמצאת ב `/etc/bind/db.local` ולכן נריץ את הפקודה שבמלבן הכהול אשר תיצור לנו עותק של התבנית הדרישה.

```

Linux BU - VMware Workstation 10 Player (Non-commercial use only)
File Edit View Search Tools Documents Help
named_dump.db named.conf.local (/etc/bind)
File Edit View Search Terminal Help
noam@noam-virtual-machine: /etc
[noam@noam-virtual-machine: /etc]$ sudo cp /etc/bind/db.local /etc/bind/db.biu.ac.il
[sudo] password for noam:
noam@noam-virtual-machine: /etc$ sudo chmod 777 resolv.conf
noam@noam-virtual-machine: /etc$ sudo systemctl restart systemd-resolved
noam@noam-virtual-machine: /etc$ sudo rm /var/cache/bind/*
noam@noam-virtual-machine: /etc$ sudo rm /etc/named/*.cache
noam@noam-virtual-machine: /etc$ sudo cp /etc/bind/db.local /etc/bind/db.biu.ac.il
noam@noam-virtual-machine: /etc$ 

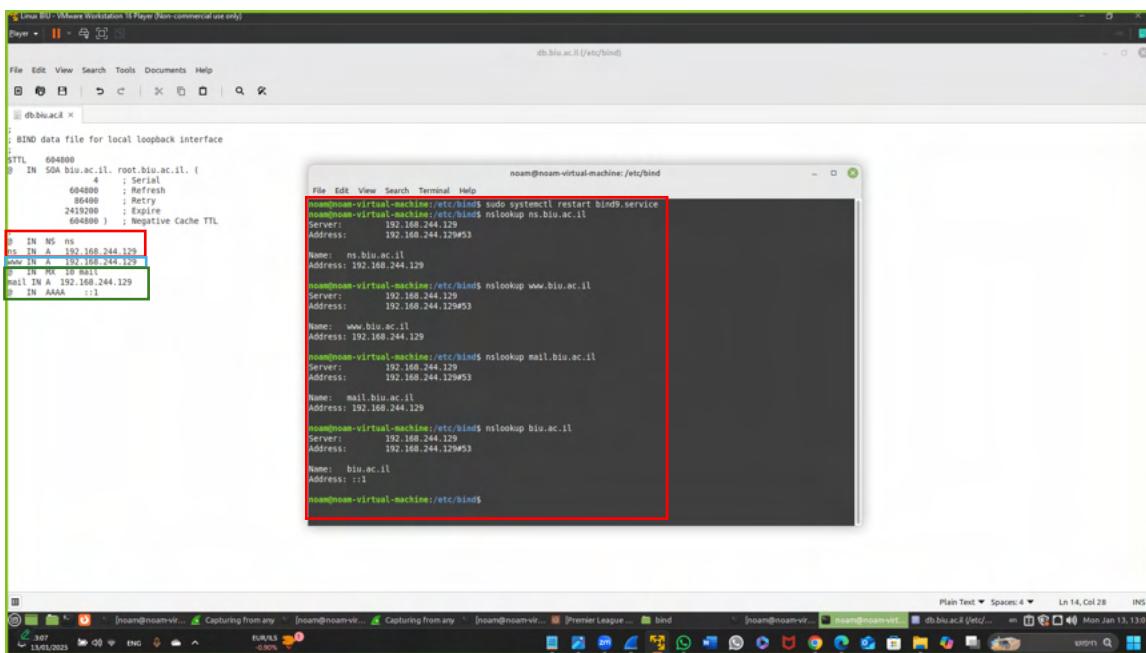
```

נכנס אל הקובץ שיצרנו ונגדיר את כל רשומות DNS הבסיסיות לשרת שאחראי על דומיין ציר לדעת לענות עליו. נבחר שבס כל מקום שבו רשום `@` - הוא יוחלף אוטומטית בדומיין `biu.ac.il`, מה שמאפשר כתיבה נקייה יותר ומעט מתחכמת.

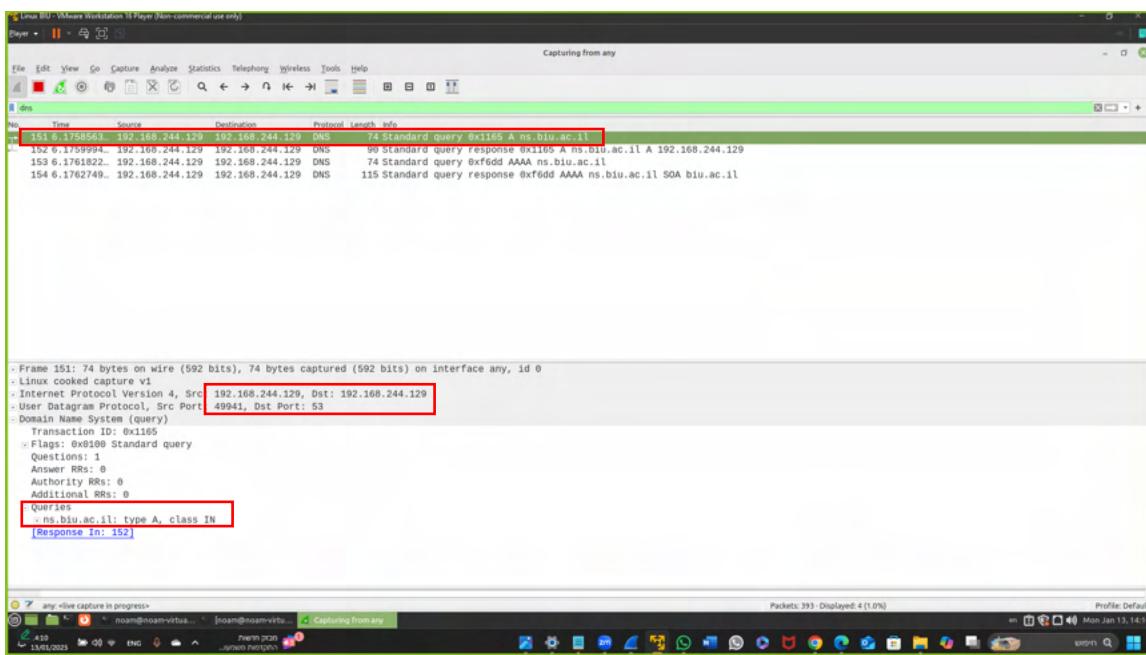
**נשנה את רשומת NS כך שתתמפה בין הדומיין `www.biu.ac.il` לכתובת IP של המחשב שלנו (במלבן הכהול).**

ניצור רשומת A שתתמפה בין `www.biu.ac.il` לכתובת IP של המחשב שלנו (במלבן הכהול).  
ניצור רשומת MX (מייל) ועל כך נמפה בין `www.biu.ac.il` ל- `mail.biu.ac.il`, בנוסף לרשותת A  
- `www.biu.ac.il` (במלבן הירוק) התשוויר גם היא ל-IP של המחשב שלנו.

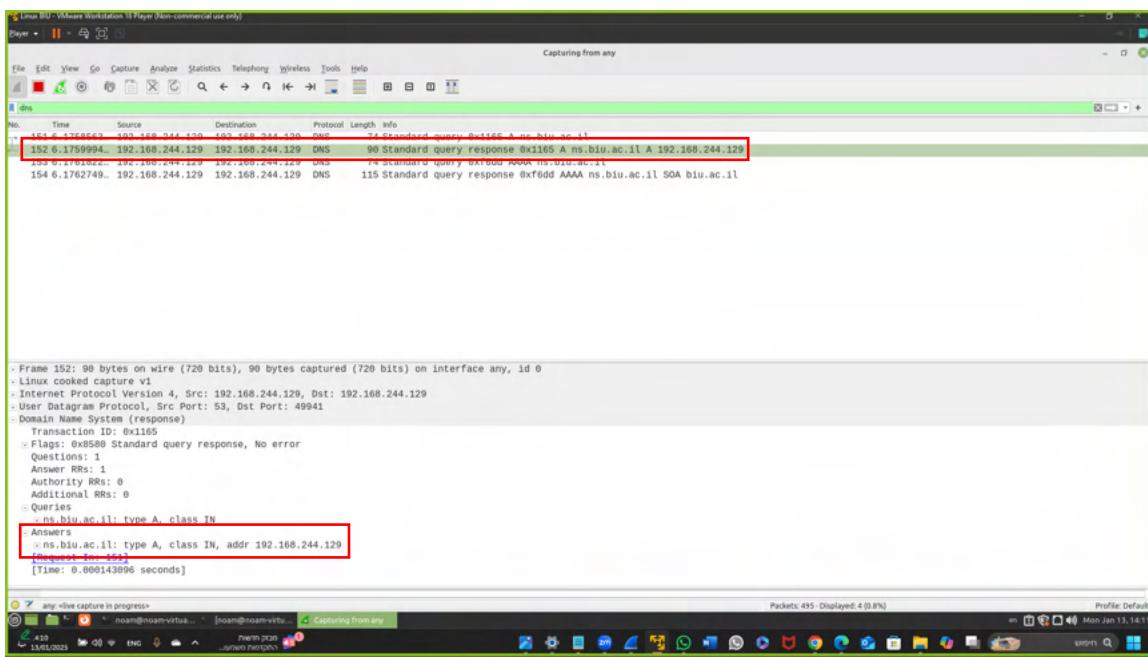
עליה את ערך `serial` בcards החדשניים יכנסו לתוקף ונשמר אותו בעזרת הפקודה – `sudo systemctl restart bind9.service`. כעת נבצע שאילתות DNS מול השרת שלנו, ונודע שאחנהן אכן מקבלים את הכתובות שהגדרכנו.



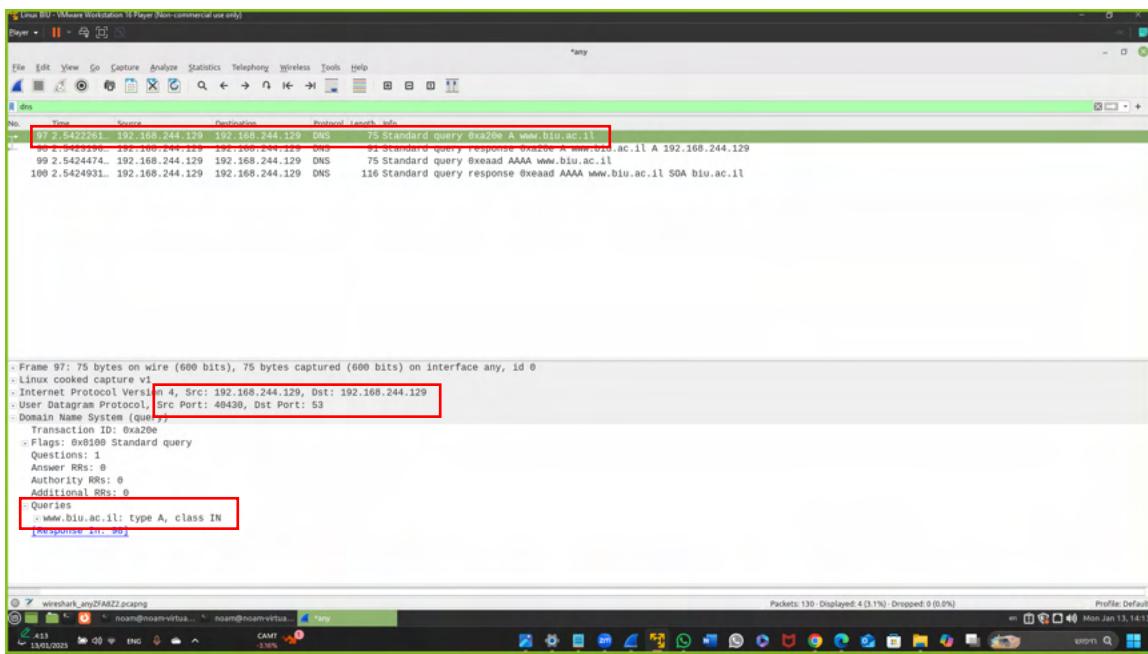
נתבונן בתיעוד השאלות שבירצנו דרך הクリש. ראשית, שאלת DNS בצד עברו il.biu.ac.il. נראה בクリש שהליקוט של לשורת שולן בקשה למצוא ע"י הפקודה nslookup ns.biu.ac.il. נראה שלשורת שולן בקשה למצוא את כתובות התחנה.



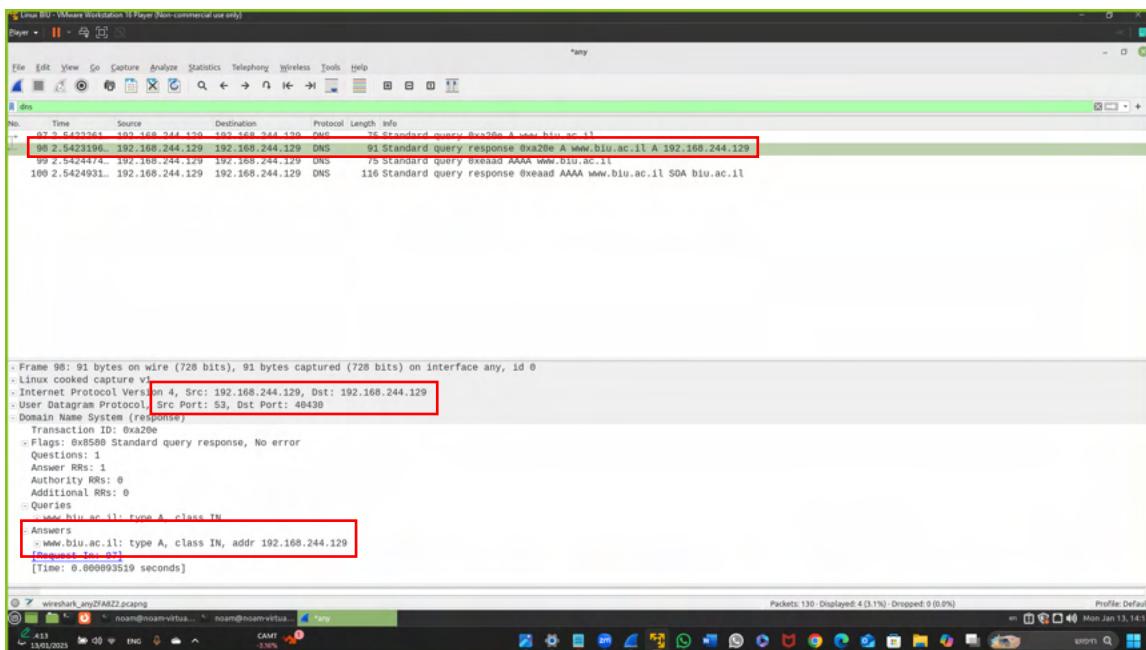
השורת מוחזיר לו את התשובה ששומרה לו בקובץ שיצרנו קודם, ככלומר הוא מוחזיר לו את IP של הדומיין ns.biu.ac.il או כפי שהגדכנו. (כפי אמרנו בהתחלה, מתבצעת מיד פניה גם כן עבור ה6עווון, וגם היא מקבלת מענה מהשרת שהגדכנו)



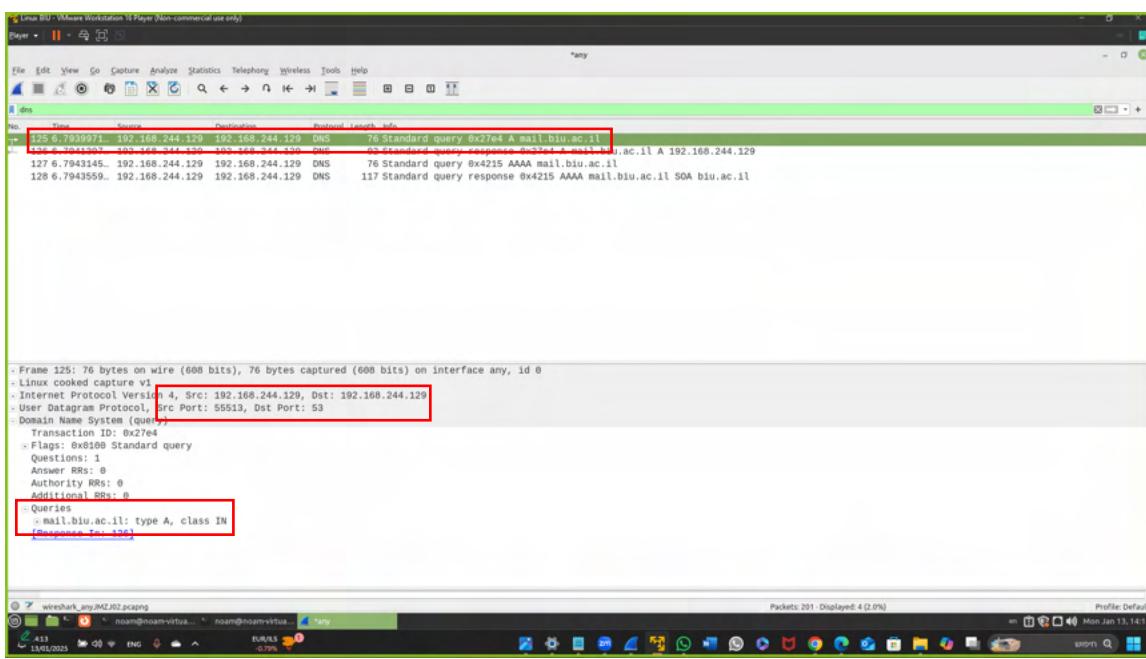
כעת נבצע שאלחת DNS אחרת בעזרת nslookup. נבקש את הIP של [www.biu.ac.il](http://www.biu.ac.il) מהשרת שלנו.



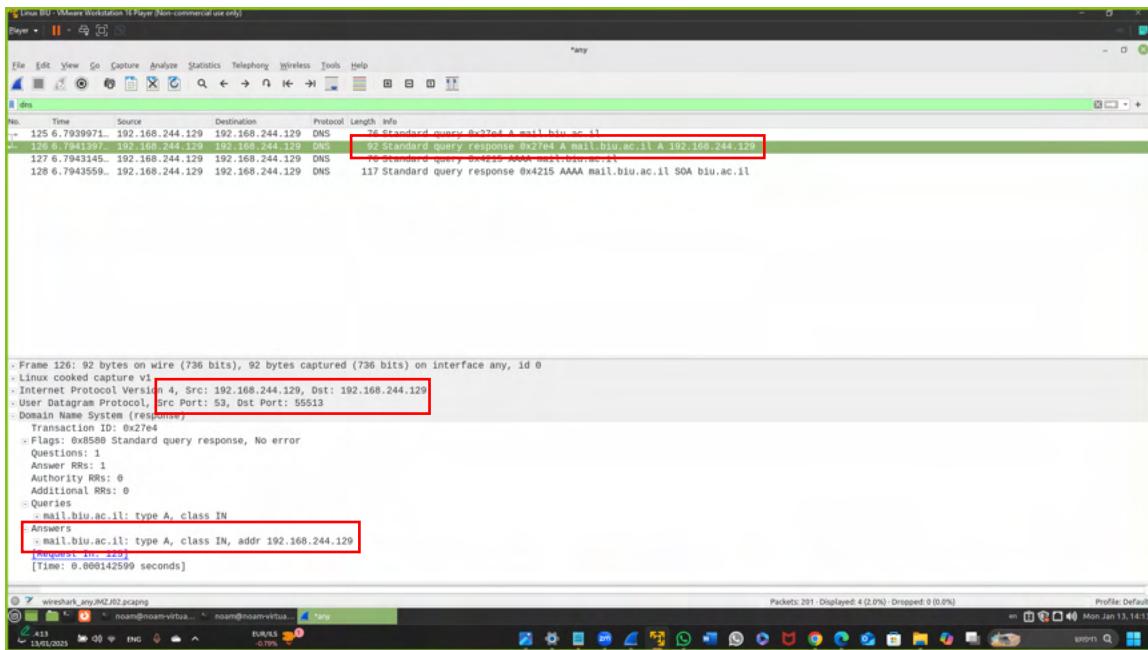
השרת מחזיר לו את IP של הדומיין כפי שהגדכנו. (שוב פניה ל69kv לאחר מכן)



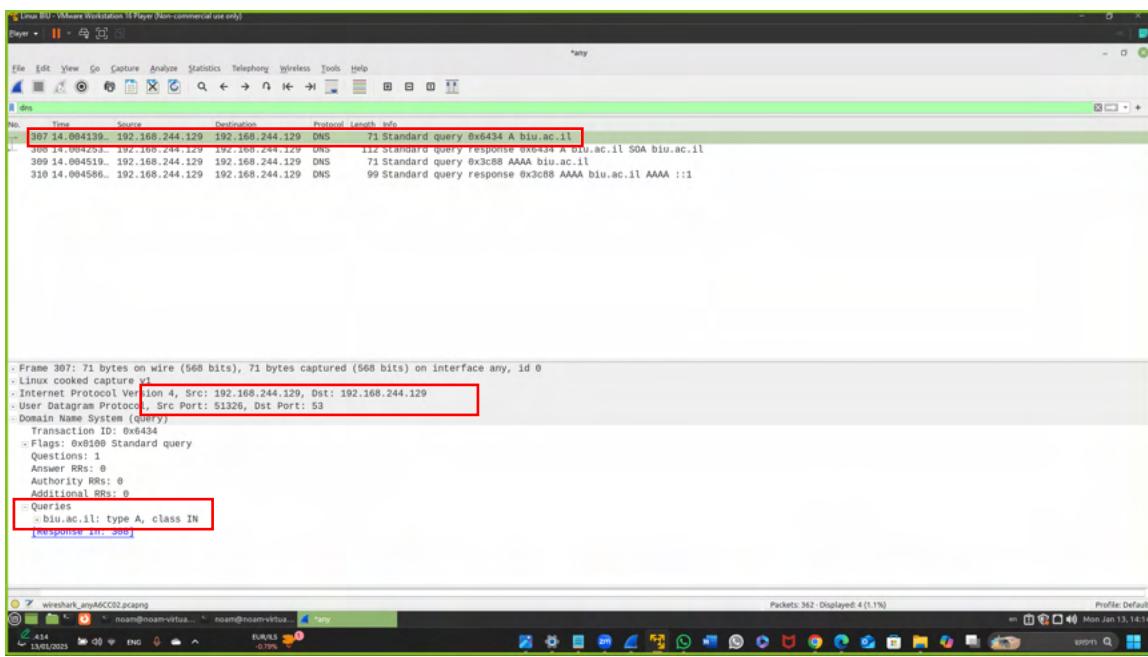
נבע שאלתת DNS אחרת בעזרת mail.biu.ac.il. נבקש את הIP של שרת המייל מהשרת שלנו.



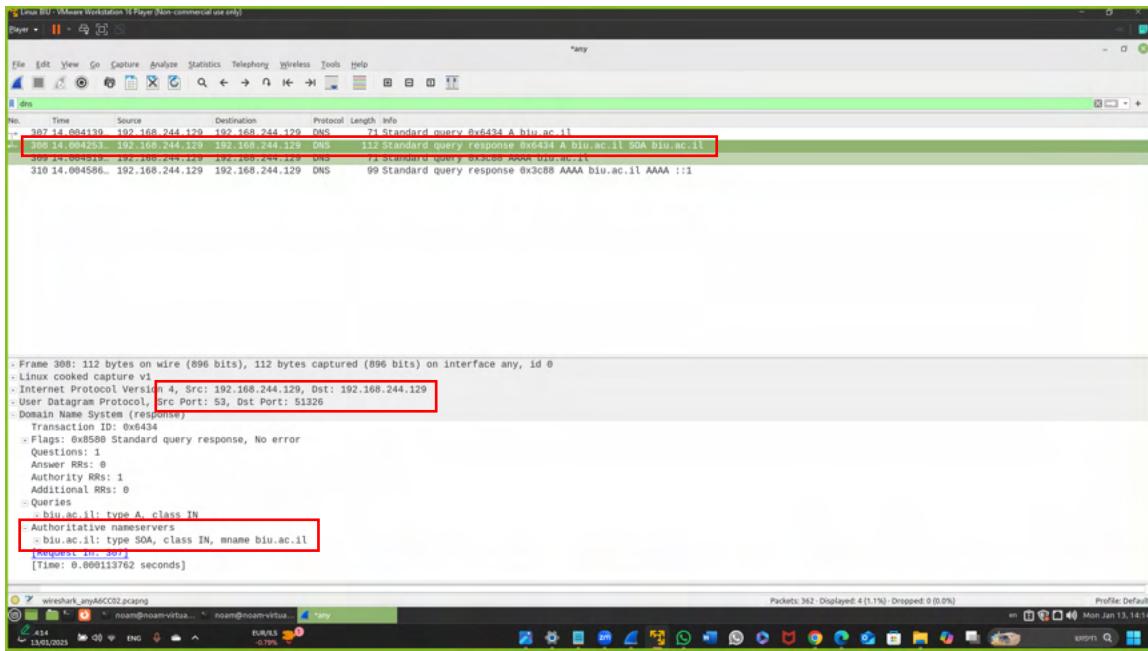
השרת מחזיר לו את IP של הדומיין ל-`mail.biu.ac.il` לאחר מכן (שוב פניה ל-`mail.biu.ac.il` שגדרנו).



נבע שאילתת DNS אחרונה בעזרת `nslookup`. נבקש את IP של הדומיין `biu.ac.il` מהשרת שלנו.



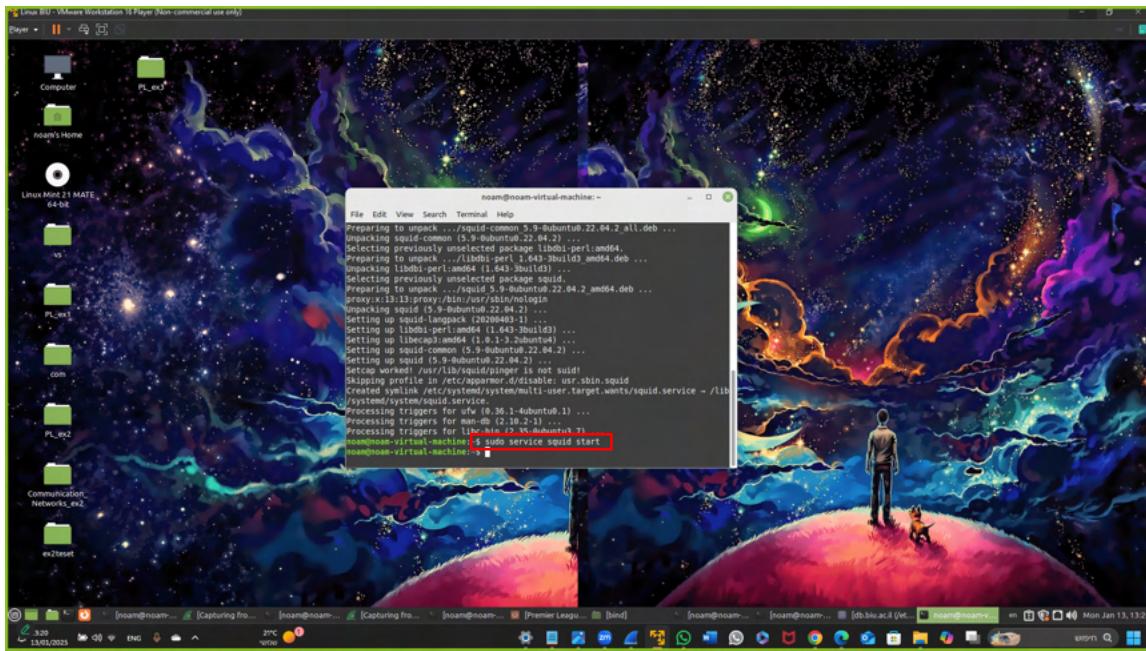
השרת מחזיר לו את שרת השמות של הדומיין `biu.ac.il`. (שוב פניה ל`ipn6` כפ' שהגדרכנו אותו.) (שוב פניה ל`ipn6` לאחר מכן)



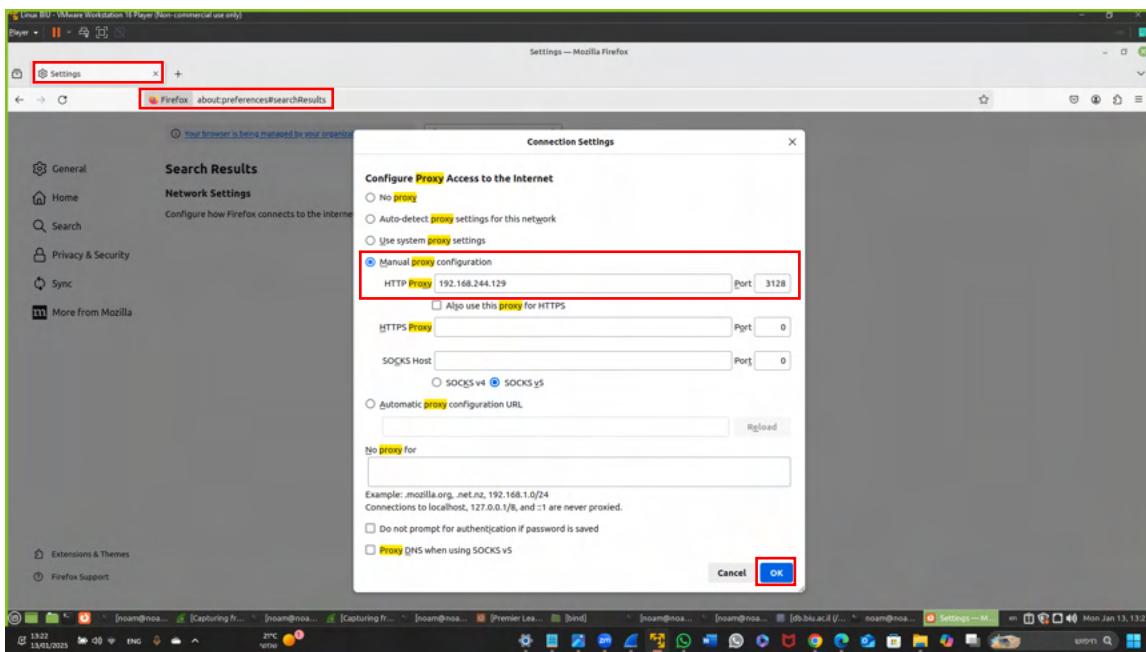
עד כאן חלק שני

## חלק שלישי

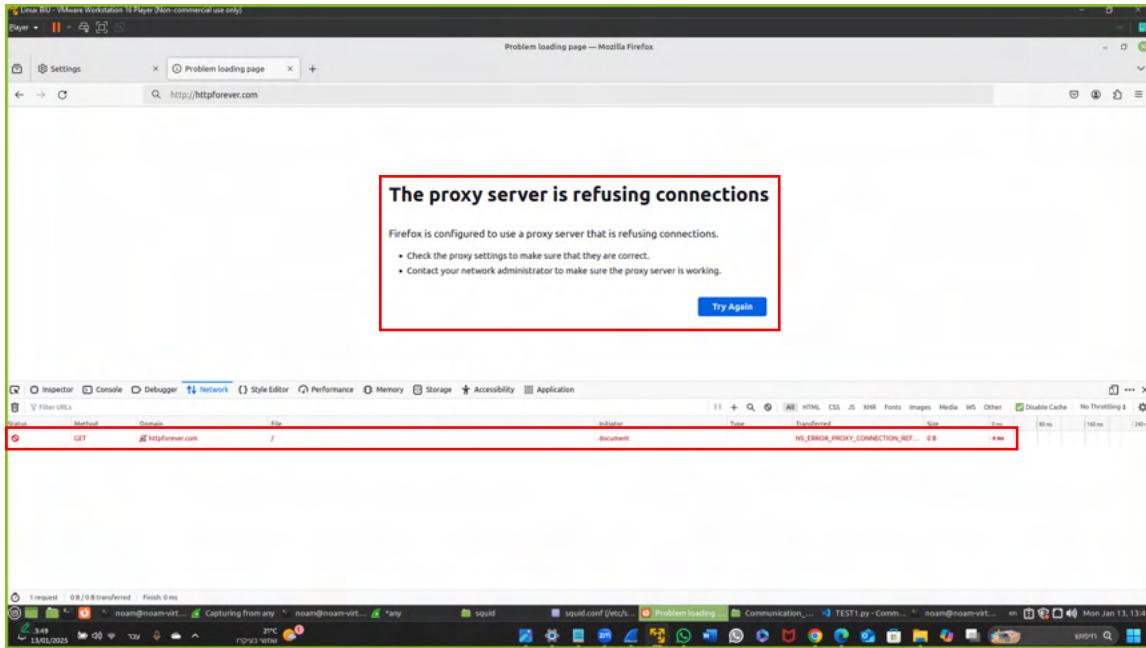
נתקן את שרת הproxy – squid ונ裏ץ אותו.



נפעיל את הדפסן ונגידיר בהגדרות שלו כך ששרת squid ישתנה יהווה שרת הproxy של ע"י הזנת כתובת הקה של השרת את הפורט של הproxy – 3128.



ונוסה כעת לגלוש לאתר האינטרנט <http://httpforever.com>, ונשים לב שגם לא מצלחים  
ושרת הפורקס' שהגדרכנו מסרב לחבר אותנו לאתר.



נשנה את ההגדרה בשרת הפורקס' שגורם לזה. נשנה בקובץ squid.conf את הסטטוס  
להיות allow.

```
# Protect Web applications running on the same server as Squid. They often
# assume that only local users can access them at "localhost" ports.
#http_access deny to_localhost

# Protect cloud servers that provide local users with sensitive info about
# their server via certain well-known link-local (a.k.a. APIPA) addresses.
#http_access deny to_linklocal

# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#include /etc/squid/conf.d/*.conf

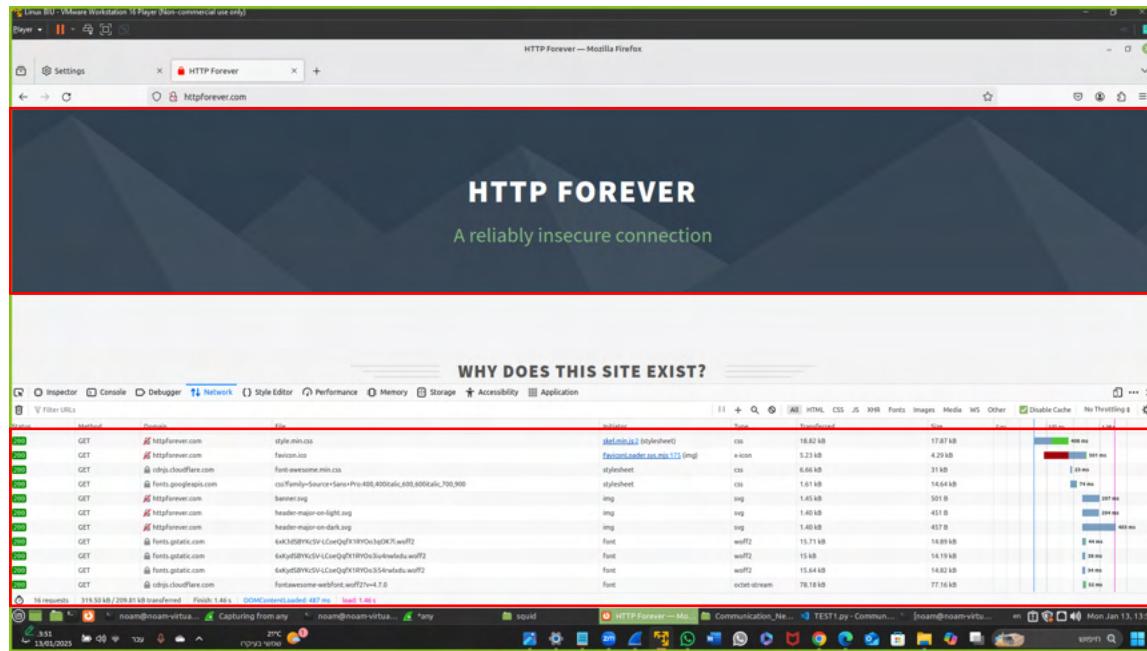
# For example, to allow access from your local networks, you may uncomment the
# following rule (and/or add rules that match your definition of "local"):
# http_access allow localnet

# And finally deny all other access to this proxy
http_access allow all

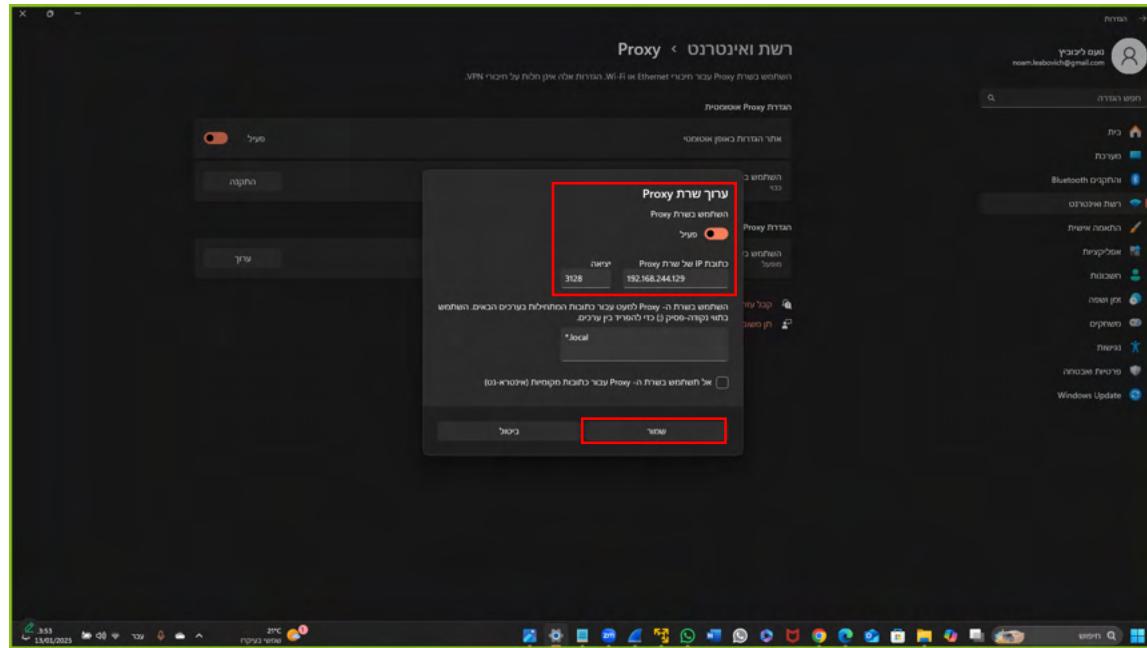
# TAG: adapted http.access
#   Allowing or Denying access based on defined access lists
#
# Essentially identical to http.access, but runs after redirectors
# and ICAP/eCAP adaptation. Allowing access control based on their
# output.
#
# If not set then only http.access is used.
#Default:
# Allow, unless rules exist in squid.conf.

# TAG: http_reply.access
.* A a Search for: http.access deny all
No matches found
```

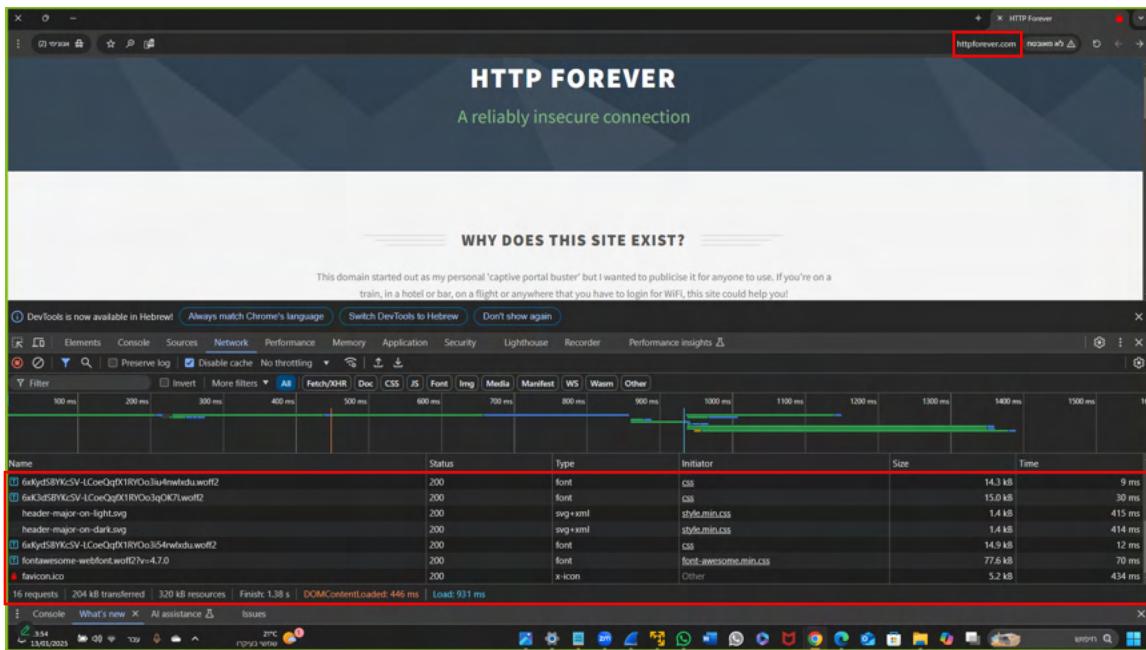
תאתחל את השרת וננסה לגש שוב לאתר <http://httpforever>. נשים לב כי כעת העמוד נתען ומוצג במלואו, וכן פניות האינטרנט עוברות בהצלחה את המisor של שרת הפרוקס, אשר לאחר השינוי שעשינו אמרו להתר הכל.



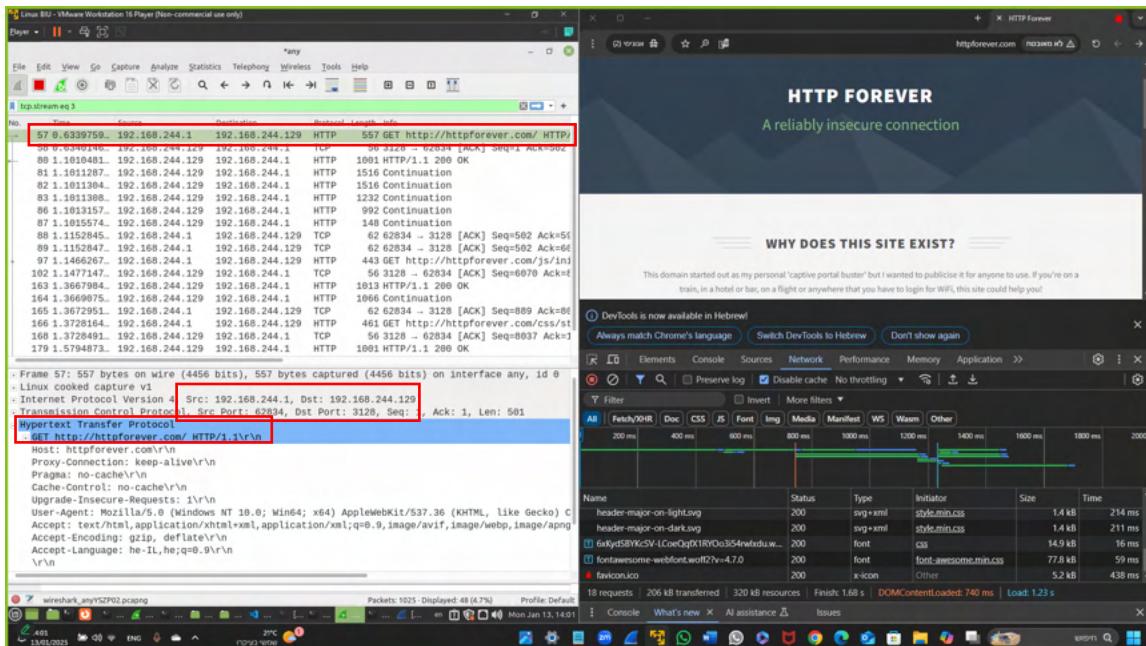
נגידר גם כן את השרת השני שלא מרים את הפרוקס כך שיגloss גם הוא דרך שרת הפרוקס שלנו.



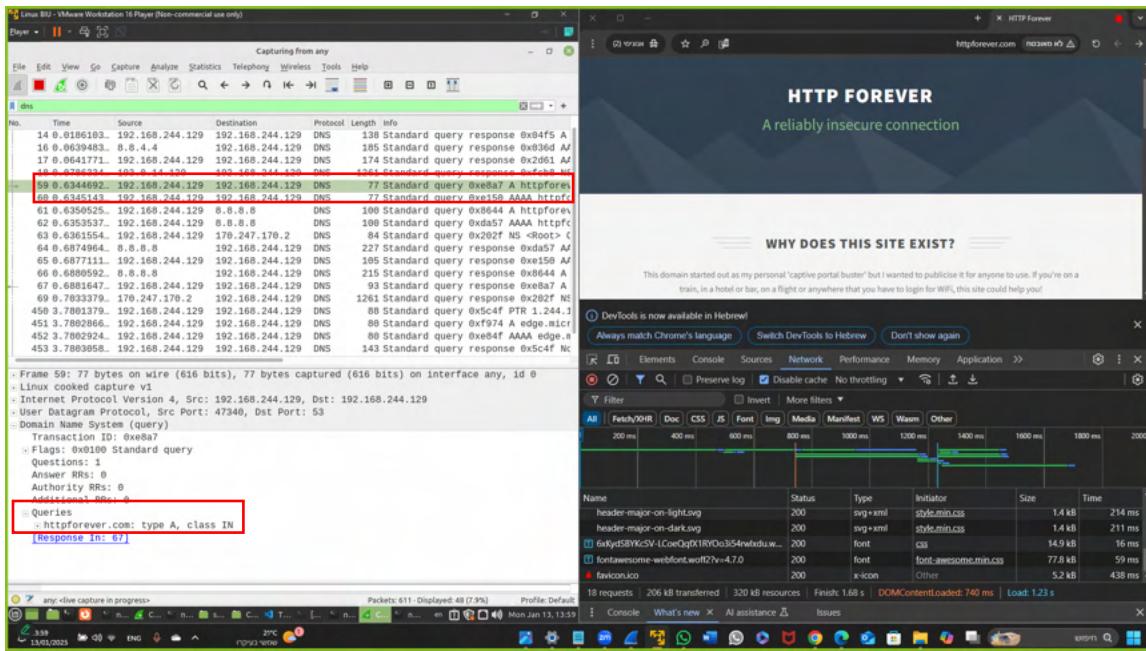
נודא שגם גם דרכו אנו מצלחים לגלוש לאתר. מצלחים.



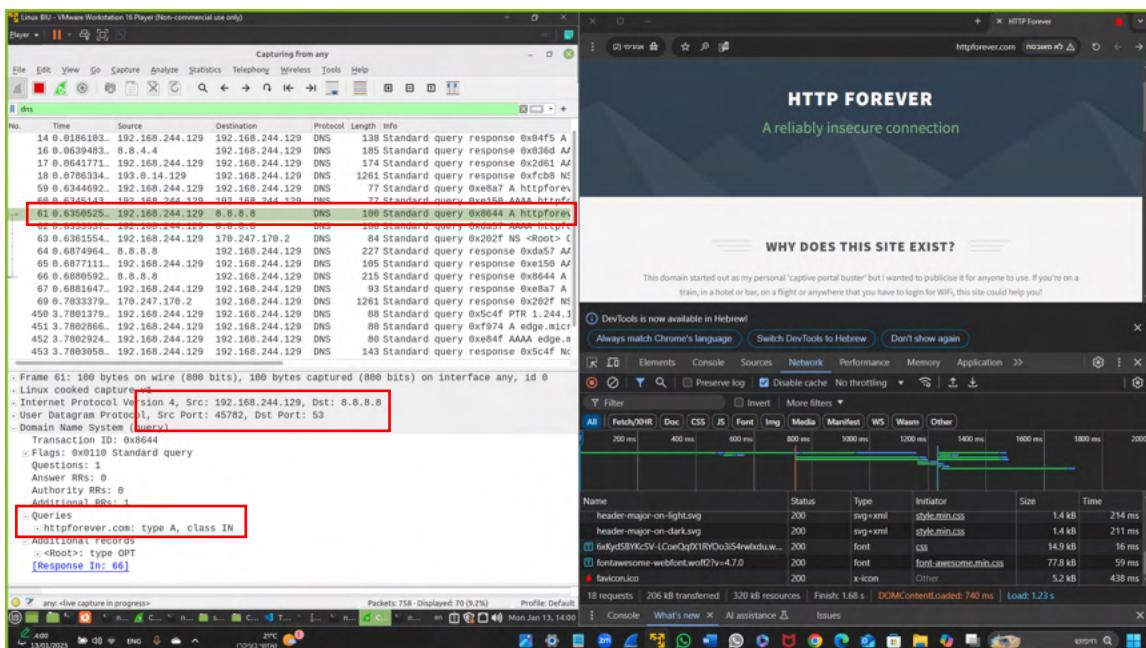
נתבונן בtabウラה שקוראת בקישור זה מארורי הקלעים בעזרת הcriish. ראשית אנו רואים כי הדף שנלנו מבצע פניה get של פרוטוקול [לשרת הפרויקט](http://httpforever.com) שלנו (הרץ בכתובת הkp) – (192.168.244.129 –



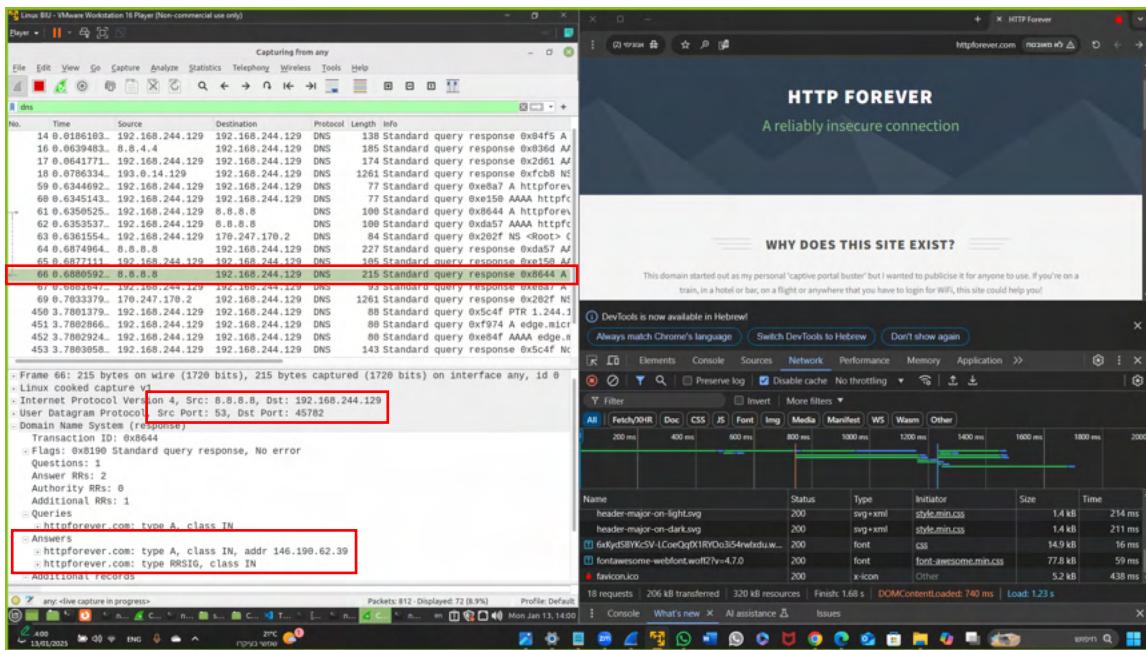
כעת לאחר שקיבל את הבקשה שרת הפרוקס' מרים על מנת למצוא את כתובת הקו של אותו שרת ממנו הוא צריך להחזיר את המשאב ללקוח (שהוא למעשה המחשב השני שלנו).  
 נשים לב בשורת DNS אליו הוא פונה זה השרת אותו הגדרנו בחלק הקודם (ניתן לראות זאת לפי הקו והפורט אליו הוא פונה שם בדיק הקו של אותו מחשב עצמו של שרת הפרוקס'), ושל כך זה מתבצע בחיבור נפרד, מזה שדרכו שרת הפרוקס' קיבל את הבקשה של הלוקו.



מכאן לאחר ששרת **DNS** שלנו שרך באותו מחשב של שרת הפרוקס' מקבל את בקשת DNS, הוא מפנה את השאלה לשרת של גול (בדומה למה שראינו בחלק השני), גם כן בחיבור נפרד.

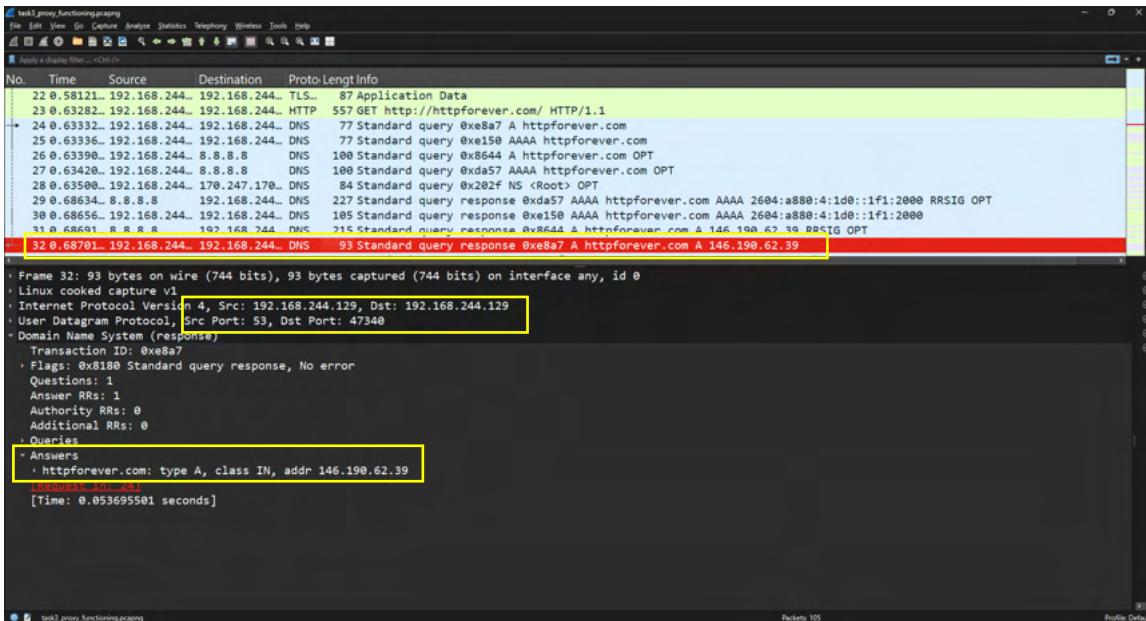


השרת של גול מחזיר כתשובה את כתובת הקו לשרת DNS המקומי שלנו.



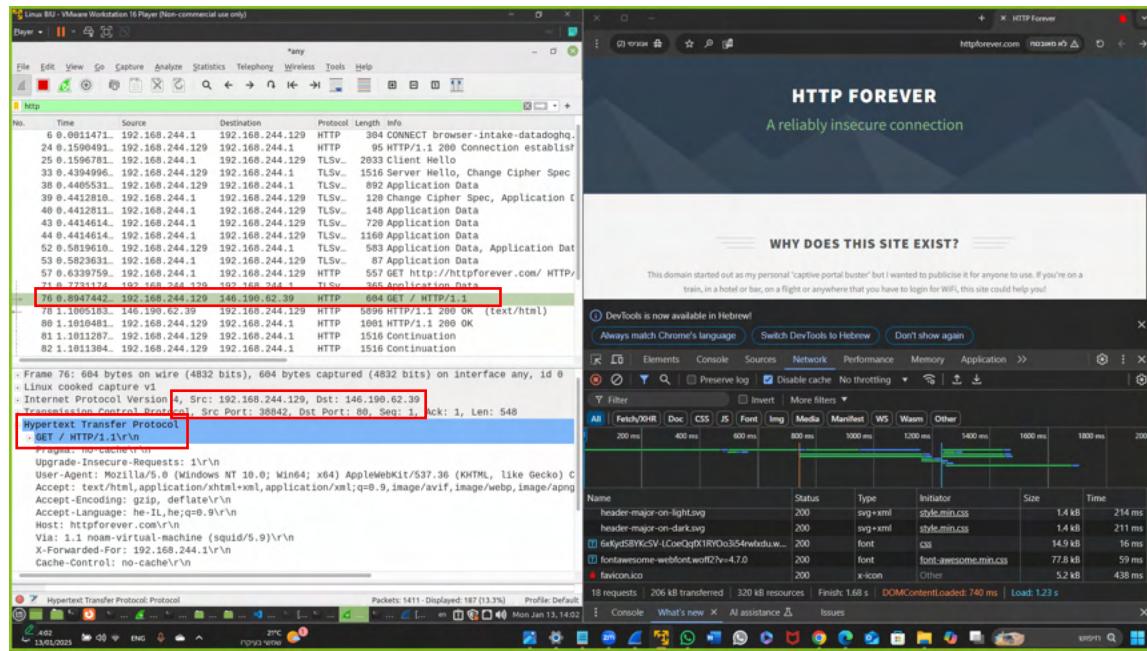
לאחר ששרת **DNS** שלנו מקבל את הכתובת של האתר הוא מסוגל לענות בחרזה את התשובה (146.190.62.39) לשאלת השם של כתובת האתר **httpforever** ששאל אותנו.

### שרת הפרויקט שלנו.

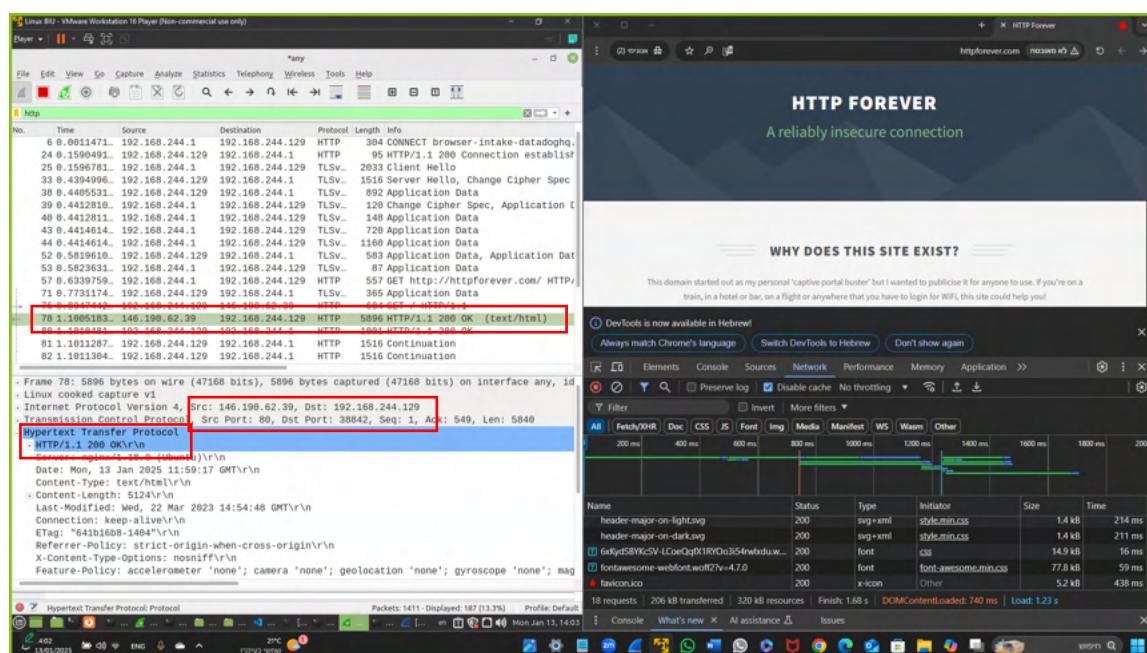


(כמו שאמרנו ב Zimmerman 4 נק' היה גם שאלת 6 נק' שנשלחו ביחד ונענו ביחד..)

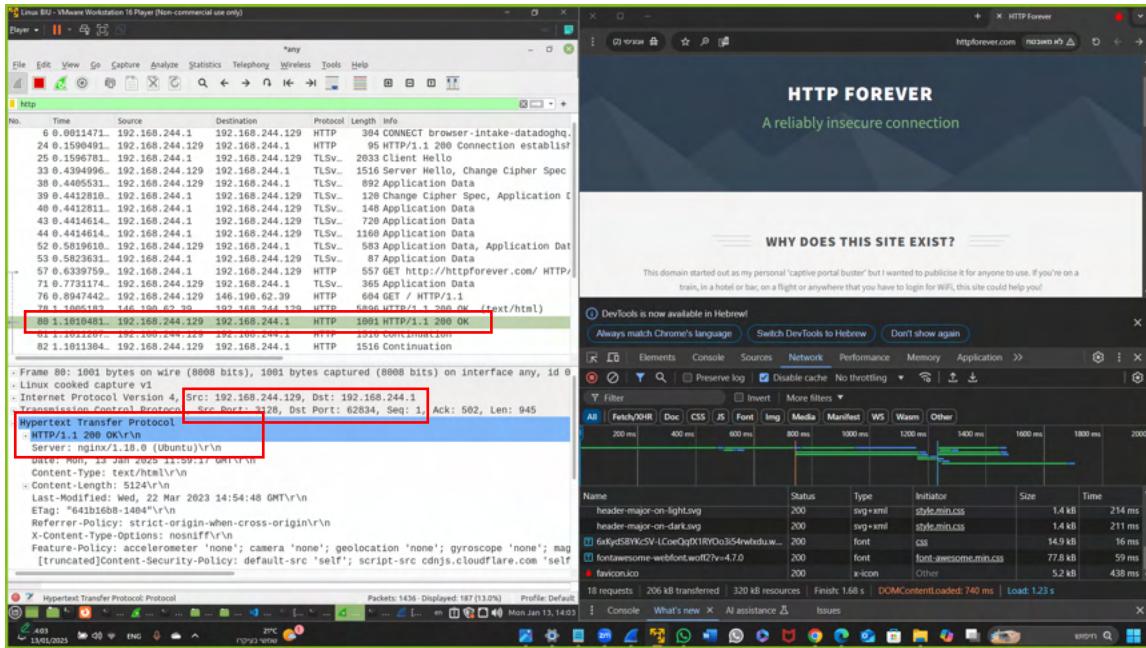
עכשו כאשר שרת הפרויקט שלנו יודע מה כתובת IP של עמוד האינטרנט ממנו הוא ציר להבי משאבChr לפקוד (הדף), הוא מבצע פניה get בפרוטוקול http לכתובת היקزا - 146.190.62.39, כמובן אנו רואים ששרת הפרויקט הוא ממש זה שמבצע את הפניה לשרת האינטרנט ועשה את העבודה עבור הדף.



שרת של **את המשאב לשרת הפרויקט** (146.190.62.39) **httpforever.ok** נotonin חיוי של 200.



ועכשיו סוף סוף שרת הפרויקט אחרי כל התיווך הענק הזה יכול להחזיר את העמוד שהדף ביקש ממנו בתחילת דף הבית ('/'). של אתר האינטרנט ('/'), אשר הדף יוכן בדף יוכן להציג לנו אותו באופן ייזואלי. החל מאותה וילך כאשר הדף יזדקק לעוד Mbpsים (נניח תמונות המוצגות בדף שהוא זה עתה קיבל), התהילר שראינו עכשיו יקרה שוב. ככלומר הדף יפהה שוב לפרויקט, אבל עכשיו הפרויקט יוכל לפנות שירות לשרת של httpforever מזמן הנחה שהוא לא יזדקק שוב לבצע שאלת DNS כיצד שכתובת הקה תהיה בקash (כמובן תלויה ttl), ורק התיווך יהיה יותר קצר מאשר הפעם הראשונה שהוא הטענו.



לסיכום ראיינו כיצד שרת הפרויקט מבצע עבור הדף את פעולות DNS כלו, וכן הוא "מוריד"/מבקש בעצם את התוכן משרת האינטרנט ורק אז מעביר אותו לקליינט – הדף. כמו כן נדבר בתהילר התיווך שראינו הטענו על חיבור נפרד, ולמשל כאשר הפרויקט דיבר בחיבור אחר עם שרת האינטרנט, החיבור שלו עם הדף היה על סוג של "hold" בלבד. תקשורת זורמת ממש, עד השלב שהוא היה מסוגל להחזיר לו דרך החיבור את התוכן שהוא קיבל משרת האינטרנט דרך החיבור הנפרד "המקביל".

עד כאן חלק שני'