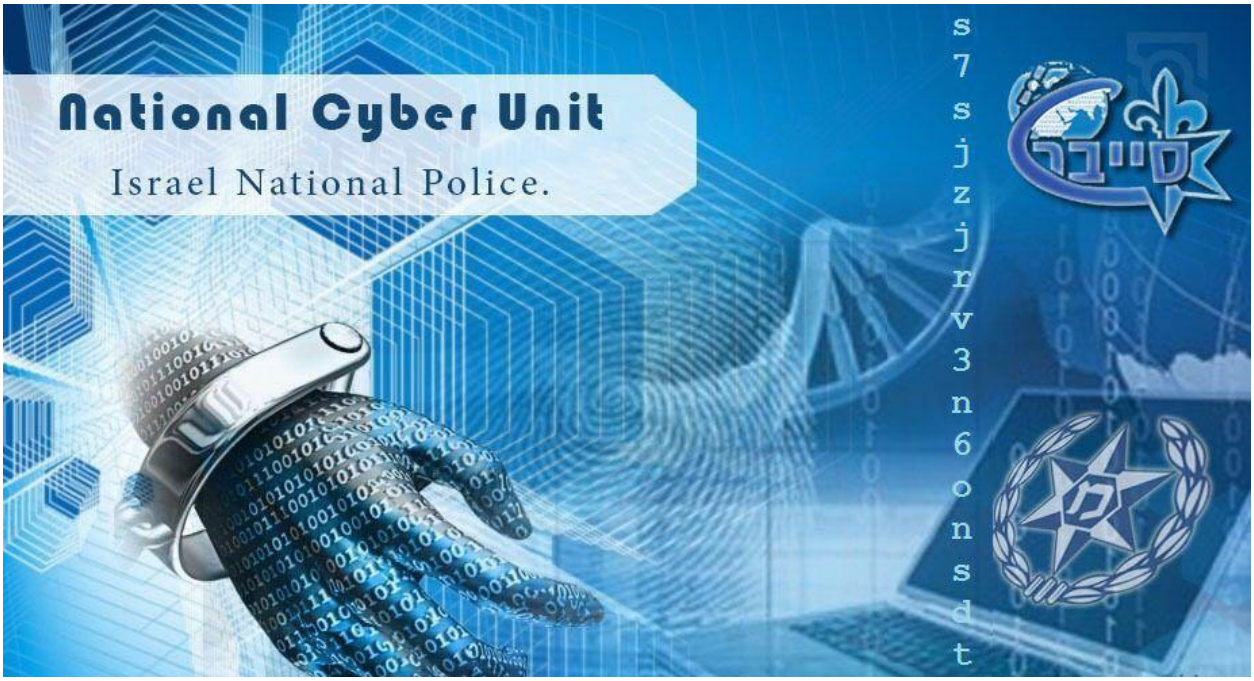
אתגר להב 433 - CYBERTECH 2018

או מה עשיתי במקום ללמוד למבחן במבוא למדמ"ח בשישי בערב.
יואב שטרנברג YOAVST.COM

יום רביעי, 31/01/18, יומו האחרון של כנס ה-CyberTech 2018. מלבד עטים רבים מספור שיספיקו לסמסטר שלם, כדור שיאבד תוך יומיים, וכובע מצחיה עם הכיתוב CYBER עליו לימים החמים כשאתה לא לובש קפוצ'ון, יצא כתבכם הנאמן מהכנס עם פיסת נייר שעליה נמצאת התמונה הבאה:

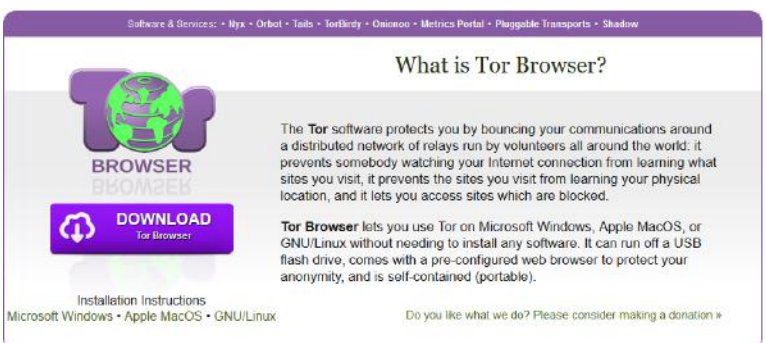


שלב ראשון: תמונה עם המון סייבר

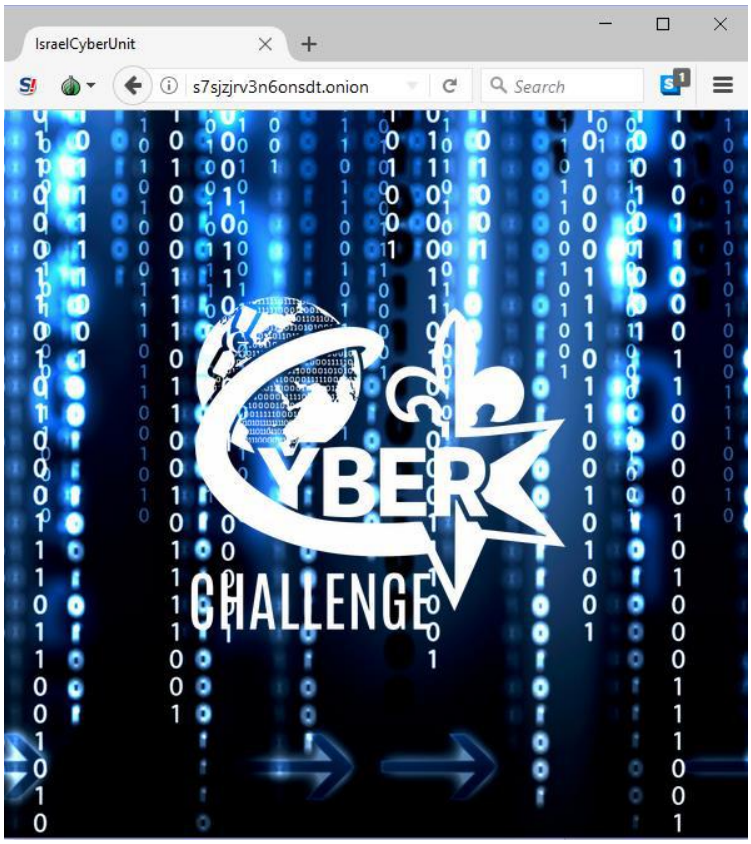
טביעת אצבע, DNA, אחדות ואפסים, שעון חכם, כל זאת ועוד נמצאת בתמונה הזאת. **סייבר.**
אבל אין שום קשר. באסה.
טוב, לא BASE64, לא ROT13, לא נמצא ב-Rainbow tables באינטרנט. אז מה כן? אז מסתבר שזוהי כתובת onion.

כתובת עם סיומת onion. הינה כתובת הניתנת לגישה רק דרך רשת TOR.
כתובות אלו מורכבות מ-16 תווים של אותיות ומספרים שמיוצרים אוטומטית על ידי הרשת בזמן ההגדרה של האתר כ-Hidden service ברשת תור.

נוריד את תור מהאתר הרשמי:



נפתח את תור ונזין את הכתובת. הידד, הגענו ליעד!



שלב שני: אתר עם המון סייבר

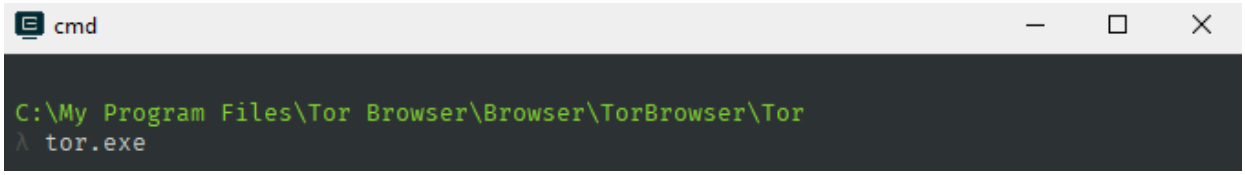
נפתח את קוד המקור של הדף וננסה לראות אם יש משהו מעניין:

```
1 <html>
2 <head>
3     <link rel="icon" href="https://s2.aconvert.com/convert/p3r68-cdx67/9yq47-3mnos-
  001.ico">
4     <style>
5         html {
6             background: url("http://www.indiafoundation.in/wp-content/uploads/2017/09
  /ssw.jpg") no-repeat center center fixed;
7             background-size: cover;
8         }
9     </style>
10    <title>IsraelCyberUnit</title>
11 </head>
12
13 <body>
14    <p style="text-align:center; margin-top: 100px;"><img src="https://i.imgur.com
  /PSfhwFv.png" width="350" height="350"></p>
15
16    <port 4444="" to="" download="" file=""></port>
17 </body>
18 </html>
```

אם נשים לב טוב, נראה שבשורה 16 רומזים לנו שעלינו להתחבר לפורט 4444 כדי להוריד את הקובץ. כשננסה להתחבר באמצעות TOR לאתר הוא יחזיר connection reset. זמן לכתוב קוד! (כלומר בניגוד לכותרת, באמת למדתי למבחן)

הואיל וזה אתר ברשת תור, לא נוכל להתחבר אליו באמצעות בקשה רגילה, לכן נאלץ לחבר את קוד הפייטון שלנו לפרוקסי של תור.

כדי להפעיל את הפרוקסי של תור יש מספר דרכים, אולם הדרך הכי אפיקטיבית שאני מצאתי, היא פשוט להריץ את הקובץ tor.exe שמוחבא בעומק הספריות של ההתקנה של תור



בנוסף, התקנתי את המודל socks עבור פייטון, שמאפשר ליצור בקלות socket שמחוברת לפרוקסי. ננסה להוריד את כל מה שהאתר שולח בפורט הזה:

```python
import socks

def recvall(sock):
    BUFF_SIZE = 4096  # 4 KiB
    data = b""    while True:
        part = sock.recv(BUFF_SIZE)
        data += part
        if not len(part):
            # either 0 or end of data
            break
    return data

socks.setdefaultproxy(socks.PROXY_TYPE_SOCKS5, "127.0.0.1", 9050, True)
s = socks.socksocket()
s.connect(('s7sjzjrv3n6onsdt.onion', 4444))
with open('downloaded_file.bin', 'wb') as f:
    f.write(recvall(s))

```

התוכנה רצה, וירד קובץ בגודל 4270 KB, הידד!

שלב שלישי: קובץ EXE, או השלב שבו אתה מתעלם מהאזהרה של האנטי וירוס

הואיל ואיננו יודעים מהו הקובץ שירד (כן רשמתי בכותרת, אני יודע, תזרמו איתי עד הפסקה הבאה), נפתח Hex Editor כדי לראות את תוכן הקובץ.

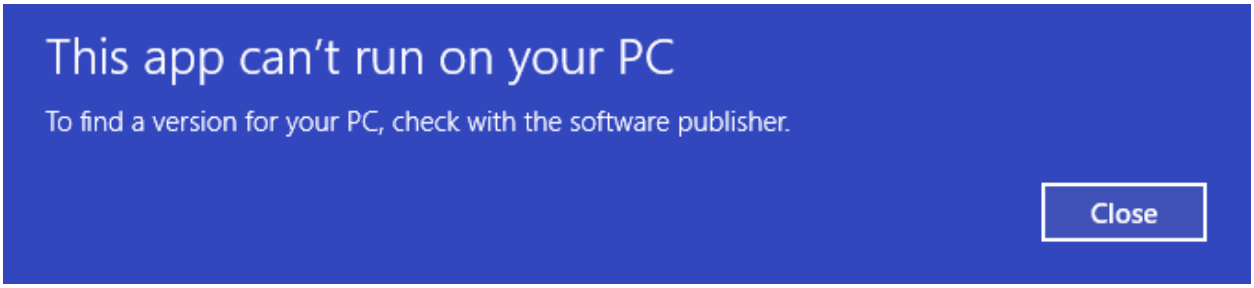
```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000   4C 41 00 00 03 00 00 00 04 00 00 00 FF FF 00 00  LA..........ÿÿ..
00000010   B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ¸.......@.......
00000020   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000030   00 00 00 00 00 00 00 00 00 00 00 00 10 01 00 00  ................
00000040   0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..°..´.Í!¸.LÍ!Th
00000050   69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is program canno
00000060   74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS
00000070   6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
```

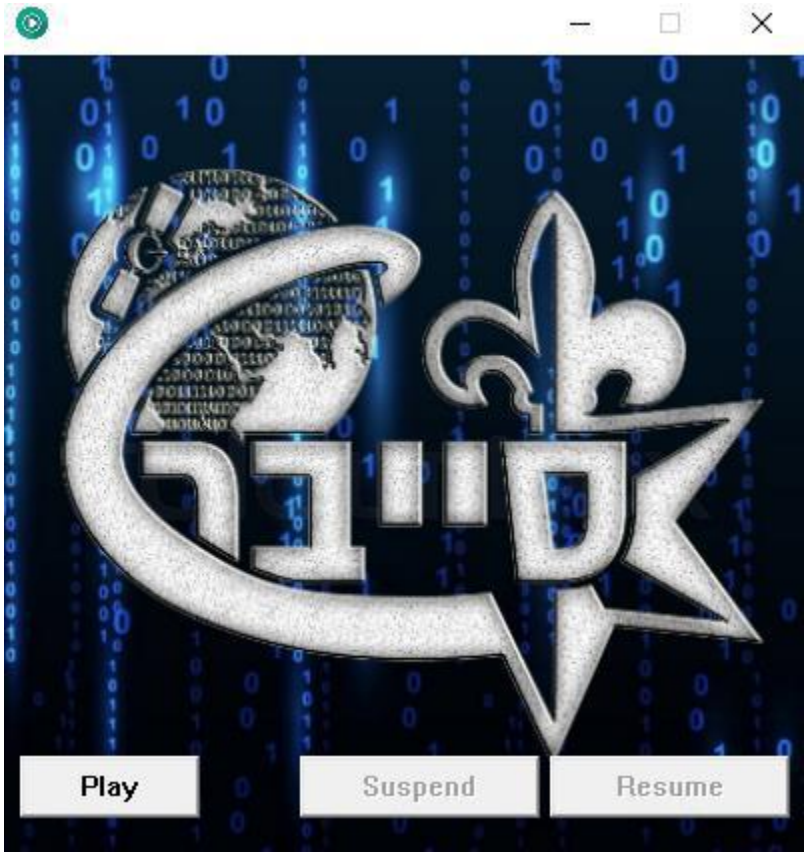
אם עשיתם CTF בעבר, או שלמדתם X86 Assembly 16bit בבית הספר וניסיתם להריץ אותו בגרסת ווינדוס מה-10 שנים האחרונות, נתקלתם בהודעה השגיאה המפורסמת:

This program cannot be run in DOS mode

לכן זה בוודאי EXE. אולם, כשאנו מנסים להריץ את קובץ, אנו נתקלים בשגיאה:

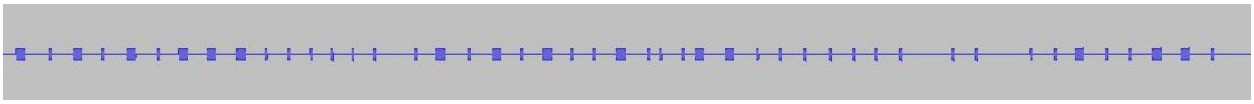


חדי העין בוודאי הבחינו (או מי שלא הבחין מוזמן להסתכל עכשיו ולהנהן) כי 2 הבתים הראשונים של הקובץ הם LA, אולם קבצי EXE מתחילים עם ה-Magic Bytes הידועים MZ. נשנה אותם ונריץ שוב. עובד! יש! נהדר! הידד!



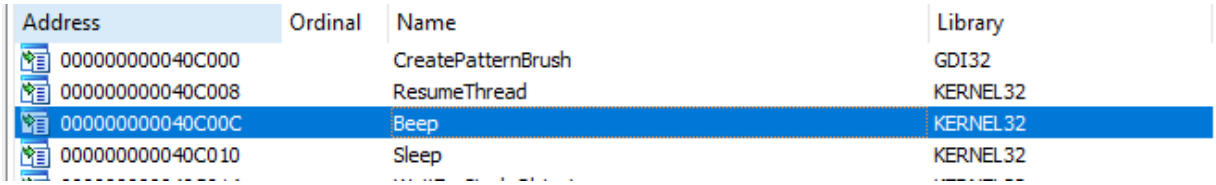
זה השלב שאני אמור להזהיר שללחוץ על Play בתוכנה שהורדתם ממקור לא ידוע במיוחד אם הוא מהרשת האפלה זה לא מומלץ, אבל כתוב פה סייבר בגדול. מה כבר יכול להשתבש?

טוב, לאחר כל הכסת"ח המשפטי נלחץ על כפתור ה-Play. לא קורה כלום. לאחר מספר דקות הבנתי שהרמקול שלי כבוי. הדלקתי אותו. עכשיו שומעים ציפצופים מהירים.



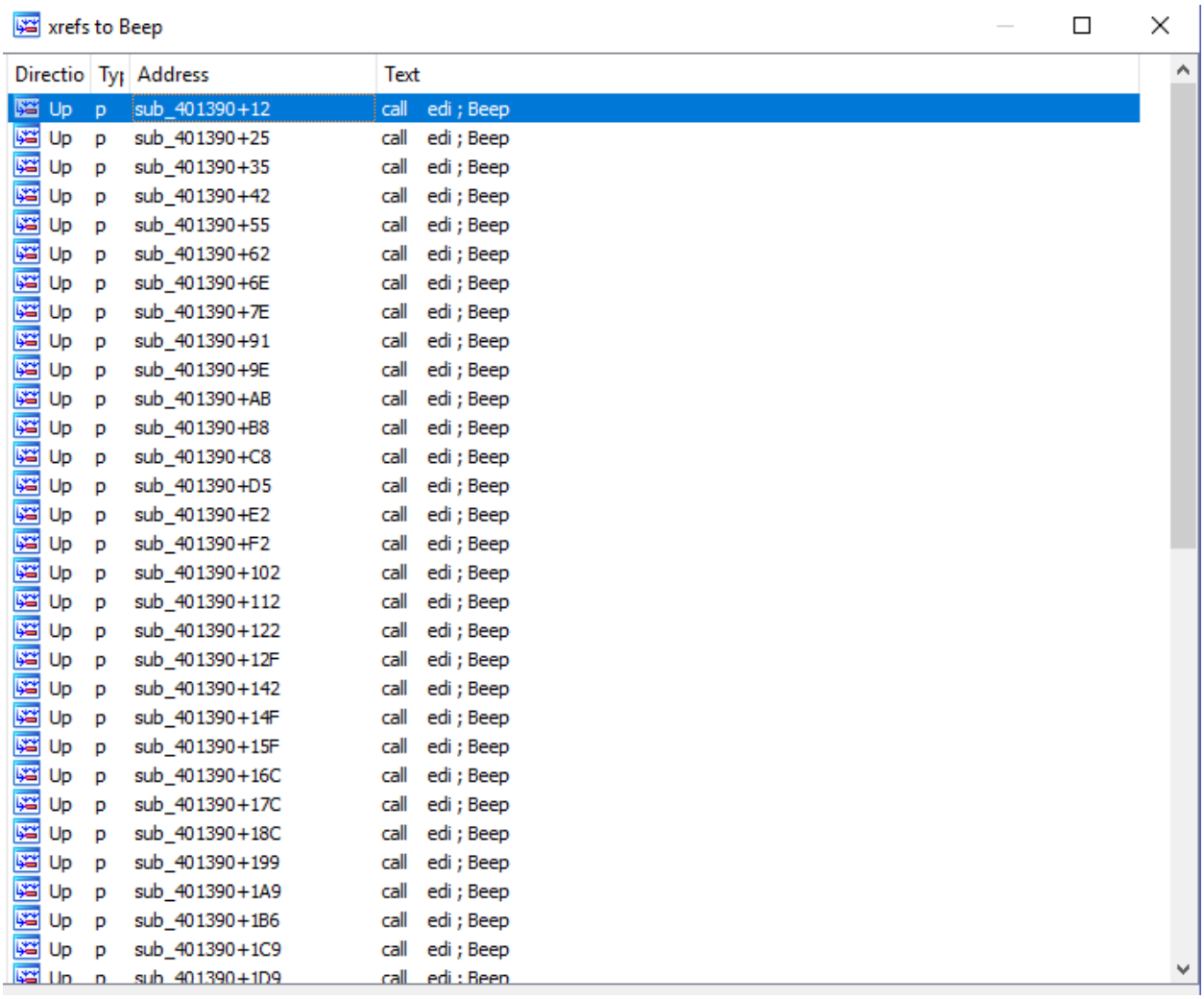
בתור אדם שראה סרט או שניים בחייו, זה חייב להיות קוד מורס. חייב! הבעיה שזה מהיר מידי עבורי, ולכן נאלץ להסתכל על הקוד של ה-EXE כדי לחלץ את קוד המורס.

נשים לב שהתוכנה משתמש בפונקציה Beep:



Address	Ordinal	Name	Library
000000000040C000		CreatePatternBrush	GDI32
000000000040C008		ResumeThread	KERNEL32
000000000040C00C		Beep	KERNEL32
000000000040C010		Sleep	KERNEL32

תוך שימוש במספר קיצורי מקלדת מועילים שמצאתי באינטרנט, נגלה כי כל הקריאות ל-Beep הן מאותן הפונקציה:



xrefs to Beep

Directio	Typ	Address	Text
Up	p	sub_401390+12	call edi ; Beep
Up	p	sub_401390+25	call edi ; Beep
Up	p	sub_401390+35	call edi ; Beep
Up	p	sub_401390+42	call edi ; Beep
Up	p	sub_401390+55	call edi ; Beep
Up	p	sub_401390+62	call edi ; Beep
Up	p	sub_401390+6E	call edi ; Beep
Up	p	sub_401390+7E	call edi ; Beep
Up	p	sub_401390+91	call edi ; Beep
Up	p	sub_401390+9E	call edi ; Beep
Up	p	sub_401390+AB	call edi ; Beep
Up	p	sub_401390+B8	call edi ; Beep
Up	p	sub_401390+C8	call edi ; Beep
Up	p	sub_401390+D5	call edi ; Beep
Up	p	sub_401390+E2	call edi ; Beep
Up	p	sub_401390+F2	call edi ; Beep
Up	p	sub_401390+102	call edi ; Beep
Up	p	sub_401390+112	call edi ; Beep
Up	p	sub_401390+122	call edi ; Beep
Up	p	sub_401390+12F	call edi ; Beep
Up	p	sub_401390+142	call edi ; Beep
Up	p	sub_401390+14F	call edi ; Beep
Up	p	sub_401390+15F	call edi ; Beep
Up	p	sub_401390+16C	call edi ; Beep
Up	p	sub_401390+17C	call edi ; Beep
Up	p	sub_401390+18C	call edi ; Beep
Up	p	sub_401390+199	call edi ; Beep
Up	p	sub_401390+1A9	call edi ; Beep
Up	p	sub_401390+1B6	call edi ; Beep
Up	p	sub_401390+1C9	call edi ; Beep
Up	p	sub_401390+1D9	call edi ; Beep

כשנכנס להגדרה של הפונקציה עצמה, נשים לב שהיא מורכבת מהמון קריאות ל-Beep אשר לאחריו בא Sleep:

```
sub_401390 proc near
push    esi
push    edi
mov     edi, ds:Beep
push    15Eh            ; dwDuration
push    320h            ; dwFreq
call    edi ; Beep
mov     esi, ds:Sleep
push    64h             ; dwMilliseconds
call    esi ; Sleep
push    5Ah             ; dwDuration
push    320h            ; dwFreq
call    edi ; Beep
push    64h             ; dwMilliseconds
call    esi ; Sleep
```

נשים לב שיש 2 סוגים של קריאות ל-Beep, קריאות עם אורך של 5Ah, וקריאות עם אורך של 15Eh. אלה יהיו הנקודות והקווים שלנו. בנוסף נשים לב שיש סליפים שהם ארוכים יותר, אלה יהיו הסימן לסוף האות. לאחר מעבר ארוך ומייגע (אך מלא התמדה ונחישות) על כל הקריאות, נקבל את קוד המורס הבא:

-.-. -.-- -... ...-- .-. -.-. .-. .. -- . ....- ...-- ...-- .. -. .--.

שאם נתרגם אותו ממורס לאנגלית, נקבל "cyb3rcrime433inp". אוקי, זה מגניב והכל, אבל מה זה עוזר לנו?

כמו סופר פואנטה טוב, הסתרתי מכם משהו. כזה שמשנה את העלילה ממש. נו אתם יודעים למה אני מתכוון... בקיצור, חלק מה-Header של קובץ PE הוא רשימת ה-Sectionים שלו. והוא כולל Section נורא מוזר:

```
00000200  00 00 00 00 00 00 00 00 2E 74 65 78 74 00 00 00  .........text...
00000210  00 B0 00 00 00 10 00 00 00 AE 00 00 00 04 00 00  .°.......®......
00000220  00 00 00 00 00 00 00 00 00 00 00 00 20 00 00 60  ............ ..`
00000230  2E 72 64 61 74 61 00 00 00 60 00 00 00 C0 00 00  .rdata...`...À..
00000240  00 5A 00 00 00 B2 00 00 00 00 00 00 00 00 00 00  .Z...².........
00000250  00 00 00 00 40 00 00 40 2E 64 61 74 61 00 00 00  ....@..@.data...
00000260  00 20 00 00 00 20 01 00 00 08 00 00 00 0C 01 00  . ... ..........
00000270  00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 C0  ............@..À
00000280  2E 67 66 69 64 73 00 00 00 10 00 00 00 40 01 00  .gfids.......@..
00000290  00 02 00 00 00 14 01 00 00 00 00 00 00 00 00 00  ................
000002A0  00 00 00 00 40 00 00 40 2E 72 73 72 63 00 00 00  ....@..@.rsrc...
000002B0  00 40 3F 00 00 50 01 00 00 32 3F 00 00 16 01 00  .@?..P...2?.....
000002C0  00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 40  ............@..@
000002D0  2E 72 65 6C 6F 63 00 00 40 0E 00 00 00 90 40 00  .reloc..@.....@.
000002E0  00 10 00 00 00 48 40 00 00 00 00 00 00 00 00 00  .....H@.........
000002F0  00 00 00 00 40 00 00 42 49 56 3D 5C 78 30 30 00  ....@..BIV=\x00.
00000300  D0 5D 02 00 00 A0 40 00 00 5E 02 00 00 58 40 00  Ð]... @..^...X@.
```

IV=\x00? זה לא שם מוכר ל-Section. אבל לא חשדתי, מה אני חשדניסט? עד עכשיו. למי שהצפנה היא נר לרגליו, בוודאי קפצה המשמעות הבאה של IV: וקטור איתחול, או initialization vector. בהצפנה, וקטור אתחול משמש להגברת הרנדומליות

של צופן בלוקים כמו AES. הואיל ויש לנו סטרינג לא ברור באורך המושלם 16, ננסה לפענח את הסגמנט הנ"ל באמצעותו ובאמצעות וקטור האיתחול הנתון, שהוא הסטרינג null.

אבל רגע! אם נגלול לסוף הסגמנט נגלה את הבתים הבאים:

```
00025DD0  69 67 6E 6F 72 65 20 6C 61 73 74 20 7A 65 72 6F  ignore last zero
00025DE0  73 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  s...............
00025DF0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

טוב, אז נמחוק אותם ונמשיך בתהליך הפענוח. כרגיל, נבלה המון זמן באינטרנט למצוא ספרייה שעושה את זה, ועובדת בפייטון 3, אבל אני אחסוך את התהליך:

```python
from Crypto.Cipher import AES

with open('aes-section.bin', 'rb') as f:
    content = f.read()
    obj = AES.new(b'cyb3rcrime433inp', AES.MODE_CBC, b'\00'*16)
    result = obj.decrypt(content)
    with open('content.bin', 'wb') as f1:
        f1.write(result)

```

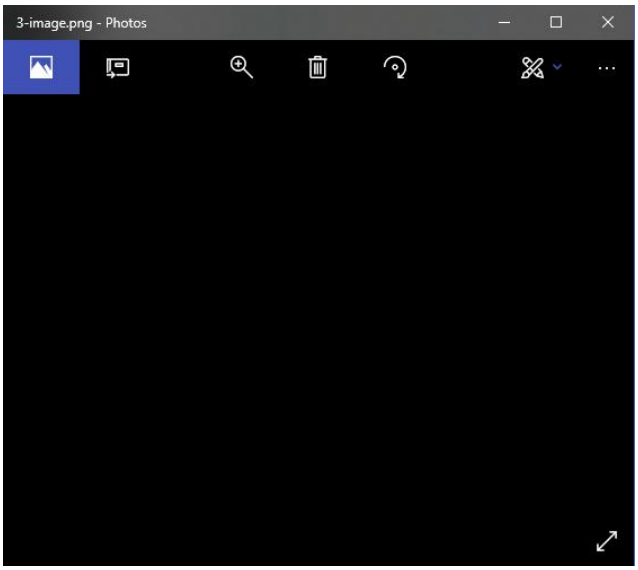
נריץ, ונקבל קובץ חדש. הפעם, למען הגיוון, נריץ את הפקודה file על הקובץ כדי לראות מאיזה סוג הוא

```
λ file content.bin
content.bin: PNG image data, 2053 x 137, 8-bit/color RGBA, non-interlaced
```

זוהי תמונה!

שלב רביעי: תמונה, ואולי קצת יותר

נשנה את הסיומת ונפתח אותה:

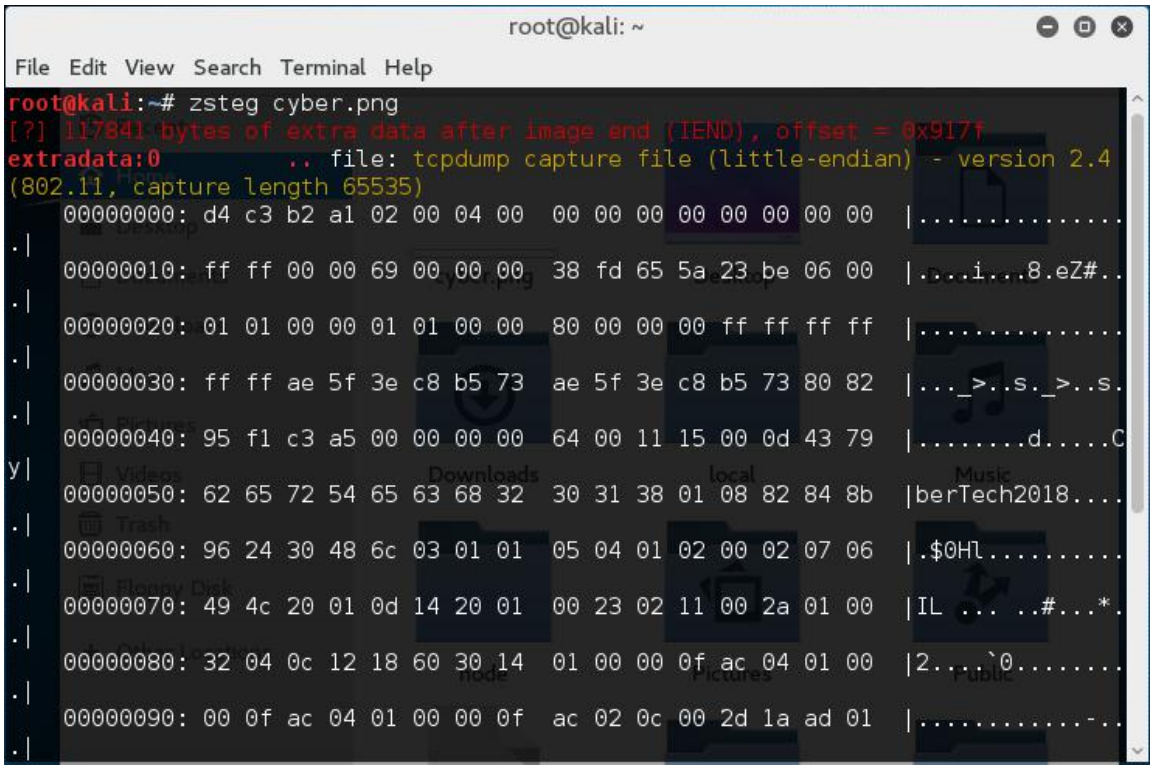


אוקי... אל תסתכל בקנקן אלא במה שבתוכו? כבר התחלתי להתעצבן על האתגר, כשלפתע עשיתי טעות שהתבררה כמכרה זהב – פתחתי את התמונה בצייר. טוב, החדשות הטובות, התמונה לא ריקה, זה פשוט טקסט שחור על לגבי רקע שקוף.
החדשות הרעות:

TGlmZUlzNGJvdXQwJ3MmMSdz

ננחש שהקו הישר מסמל L קטנה, ונריץ Base64, קיבלנו "LifeIs4bout0's&1's". האם זה רפרנס לפרק "eps1.1_ones-and-zer0es.mpeg" של Mr Robot? אני מניח שלעולם לא נדע. בכל מקרה, אני חושב שמגיע לי ה-MEME הבא:

טוב יש לנו סלוגן קליט. הגיע הזמן לראות מה עוד התמונה מחביאה. לשם כך נשתמש ב-Zsteg הידוע:



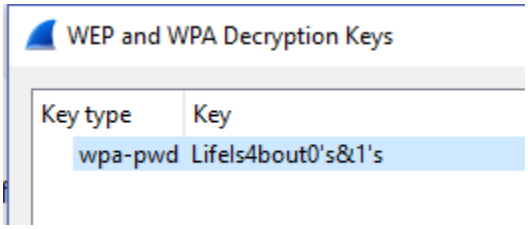
שלב חמישי: ה-WIFI של השכן תמיד מהיר יותר

הפתעה! יש קובץ הסנפה של תעבורת רשת! נחלץ אותו מהקובץ באמצעות ה-offset האורך הנתונים, ונפתח ב-Wireshark:

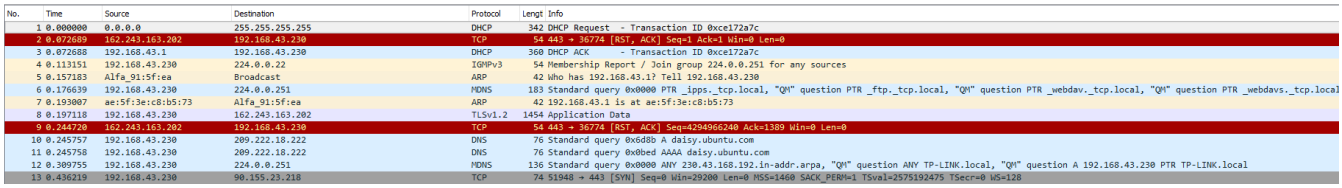
No.	Time	Source	Destination	Protocol	Lengt	Info
1	0.000000	ae:5f:3e:c8:b5:73	Broadcast	802.11	257	Beacon frame, SN=2088, FN=0, Flags=........, BI=100, SSID=CyberTech2018
2	0.870932		IntelCor_7b:da:a4 (b8:8a:60:7b:da:a4) (RA)	802.11	10	Acknowledgement, Flags=........
3	2.175570		IntelCor_7b:da:a4 (b8:8a:60:7b:da:a4) (RA)	802.11	10	Acknowledgement, Flags=........
4	3.297472	ae:5f:3e:c8:b5:73	IntelCor_ab:f6:1c	802.11	251	Probe Response, SN=2121, FN=0, Flags=........, BI=100, SSID=CyberTech2018
5	3.297940		ae:5f:3e:c8:b5:73 (ae:5f:3e:c8:b5:73) (RA)	802.11	10	Acknowledgement, Flags=........
6	3.300032	ae:5f:3e:c8:b5:73	IntelCor_ab:f6:1c	802.11	251	Probe Response, SN=2122, FN=0, Flags=........, BI=100, SSID=CyberTech2018
7	3.300500		ae:5f:3e:c8:b5:73 (ae:5f:3e:c8:b5:73) (RA)	802.11	10	Acknowledgement, Flags=........
8	4.572984	ae:5f:3e:c8:b5:73	Broadcast	802.11	76	Data, SN=2136, FN=0, Flags=.p....F.
9	5.392178	ae:5f:3e:c8:b5:73	Broadcast	802.11	76	Data, SN=2145, FN=0, Flags=.p....F.
10	5.992786		a2:6c:ac:7f:10:e5 (a2:6c:ac:7f:10:e5) (RA)	802.11	10	Acknowledgement, Flags=........

זוהי תעבורת רשת של רשת WIFI לא חשודה בכלל בשם CyberTech2018, אבל היא מוצפנת ולכן אנחנו לא יכולים לפענח את התעבורה. לו רק היינו יודעים את הסיסמה....

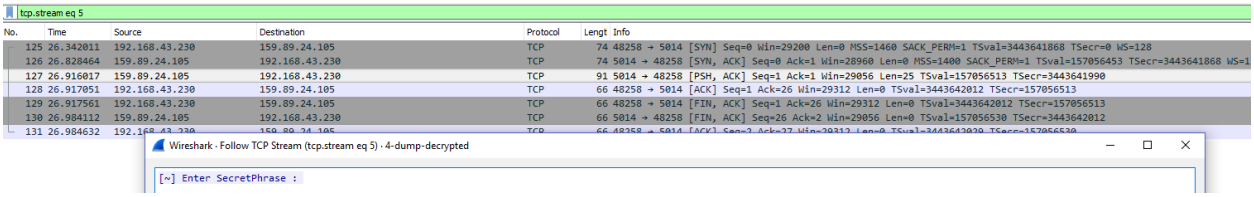
אבל אנחנו יודעים! "Lifels4bout0's&1's" באופן מפתיע היא הסיסמה.



כעת נגלו אלינו הפקטות המפוענחות:



לאחר מעבר על הפקטות, אנו מגלים TCP STREAM חשוד:



ננסה להתחבר אל הכתובת ולראות אם היא זמינה.

```python
import socket
s = socket.socket()
s.connect(('159.89.24.105', 5014))
print(s.recv(1024))
```

אולם כשנריץ את זה, נגלה שהוא מבקש סיסמה. אנחנו במבוי סתום!

```
λ python 4-connect-server.py
b'[~] Enter SecretPhrase : '
```

נחזור לקובץ ההסנפה, נבזבז עוד מספר דקות עד שעות על לנסות לחפש בו משהו חשוד ונשים לב שיש פקטות מוזרות במקצת:

ipv6.hopopts and not icmpv6

No.	Time	Source	Destination	Protocol	Lengt	Info
50	5.956474	192.168.43.230	192.168.10.235	IPv4	39	IPv6 hop-by-hop options[Malformed Packet]
51	6.017914	192.168.43.230	192.168.10.235	IPv4	39	IPv6 hop-by-hop options[Malformed Packet]
54	6.394234	192.168.43.230	192.168.10.235	IPv4	39	IPv6 hop-by-hop options[Malformed Packet]
56	6.481851	192.168.43.230	192.168.10.235	IPv4	39	IPv6 hop-by-hop options[Malformed Packet]
61	6.662586	192.168.43.230	192.168.10.235	IPv4	39	IPv6 hop-by-hop options[Malformed Packet]
62	6.751161	192.168.43.230	192.168.10.235	IPv4	39	IPv6 hop-by-hop options[Malformed Packet]
78	16.158715	192.168.43.230	192.168.10.235	IPv4	39	IPv6 hop-by-hop options[Malformed Packet]
81	16.265208	192.168.43.230	192.168.10.235	IPv4	39	IPv6 hop-by-hop options[Malformed Packet]
82	16.349691	192.168.43.230	192.168.10.235	IPv4	39	IPv6 hop-by-hop options[Malformed Packet]
88	16.534075	192.168.43.230	192.168.10.235	IPv4	39	IPv6 hop-by-hop options[Malformed Packet]
90	16.694845	192.168.43.230	192.168.10.235	IPv4	39	IPv6 hop-by-hop options[Malformed Packet]
97	16.787003	192.168.43.230	192.168.10.235	IPv4	39	IPv6 hop-by-hop options[Malformed Packet]

מלבד העובדה שהן מופיעות אחת אחרי השנייה מבחינת זמנים, הייצוג הבינארי שלהם כל אחת מהן כולל 4 בתים שנראים כמו Base64:

```
0000  ae 5f 3e c8 b5 73 00 c0  ca 91 5f ea 08 00 45 00   ._>..s.. .._...E.
0010  00 19 00 01 00 00 40 00  c2 c2 c0 a8 2b e6 c0 a8   ......@. ....+...
0020  0a eb 61 77 3d 3d 0a                               ..aw==.
```

אז ניקח את כל הסטרינגים הללו, נמיר אותם לתווים, וננסה לשלוח את זה כסיסמה. עובד. כה לחי!
ירד קובץ בגודל 90 KB הנראה בערך ככה:

5-server-content.txt - Notepad

File Edit Format View Help

```
<chunk number:84>           ;$M7          MM                    7$HNMNQ?.              MM -CNMNO.
<chunk number:14>                                                          .CMMO??C$MN>
<chunk number:47>        ?MMMMMMMMMMMMMMMMMMMMMMMMH:                          :MMMMMMMMMMMMMMMMMMMMMMMMM
<chunk number:22>:;              .-!7OHMMMMMMMMMMMMMMMMMMMMMMMMM:
<chunk number:26>                        -C-
<chunk number:52>                        ?M7 7NMC;
<chunk number:63>                                                                        ?MMMMMMMMMMMMM
<chunk number:41>                     .NMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM:             :MMMMM
<chunk number:16>MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMN:
<chunk number:58>HNQC>:.                                                    .$MMM!
<chunk number:33>>>          !MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMO-    .--;;.;;.;.;:>OHMMMMMMMMMMMMMMMM
<chunk number:73>                                        >NM$.       7MMMMMMMM-                  CMMMM
<chunk number:74>MMMMMMMMMMMMMMMN!
<chunk number:29>     :HMQC7OMHOHMH!                                                        :MMMMMMMMMMMM
<chunk number:56>                                  OMMM?                 ;?NMMMMMMMMMMMMMMMMMMQ7.
<chunk number:81>               ;!777!;
<chunk number:44>               :MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMN-               :MMMMMMMMMMMMMM
<chunk number:42>MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM!
<chunk number:10>
<chunk number:37>                                                                    ;$MC;  ;Q$--?MN:
<chunk number:35>                                                        !Q.
<chunk number:83>               MMMMM
<chunk number:28>MMMMMMMMMMMMMMMMMMMMMMMMM?
<chunk number:39>MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
<chunk number:71>MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM7
<chunk number:27>                        ?MMMMMMMMMMMMMMMMMMMMMMMC                        >MMMMMMMMMM
<chunk number:50>    ..              .:;                          OMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
```

אני לא צריך להסביר מה עושים מכאן, נכון? פשוט מסדרים מחדש את ה-chunkים בהתאם למספר שלהם (טוב נו הסברתי)

```python
import re

with open('chunks.txt', 'r') as f:
    text = f.read()

results = sorted((s[0:2], s[3:]) for s in text.split('<chunk number:')[1:])
with open('result.txt', 'w') as f:
    for num, text in results:
        f.write(text)

```

לאחר הסידור מחדש נקבל את קובץ הטקסט הבא.

NICE ASCII ART!

שלב שישי: מצלמת הרשת ישירות לתוככי הסייבר

נכנס לכתובת http://zsjqn6f6c6qyf4wq.onion המופיעה בצד שמאל של התמונה.

נקבל ממשק התחברות למצלמת רשת. בקובץ הטקסט כתוב למעלה Lahav ולמטה Lahav433lnp, אז ננסה להזין את זה. זה עובד. כמה לא צפוי. אפשר להגיד נס אפילו.

סיימנו!

מקווה שנהנתם מה-Recap, ושאתם לא שונאים אותי יותר מידי אחריו. הלכתי ללמוד למבחן אמיתי כמו מבוא לקומבינטוריקה ותורת הגרפים.