

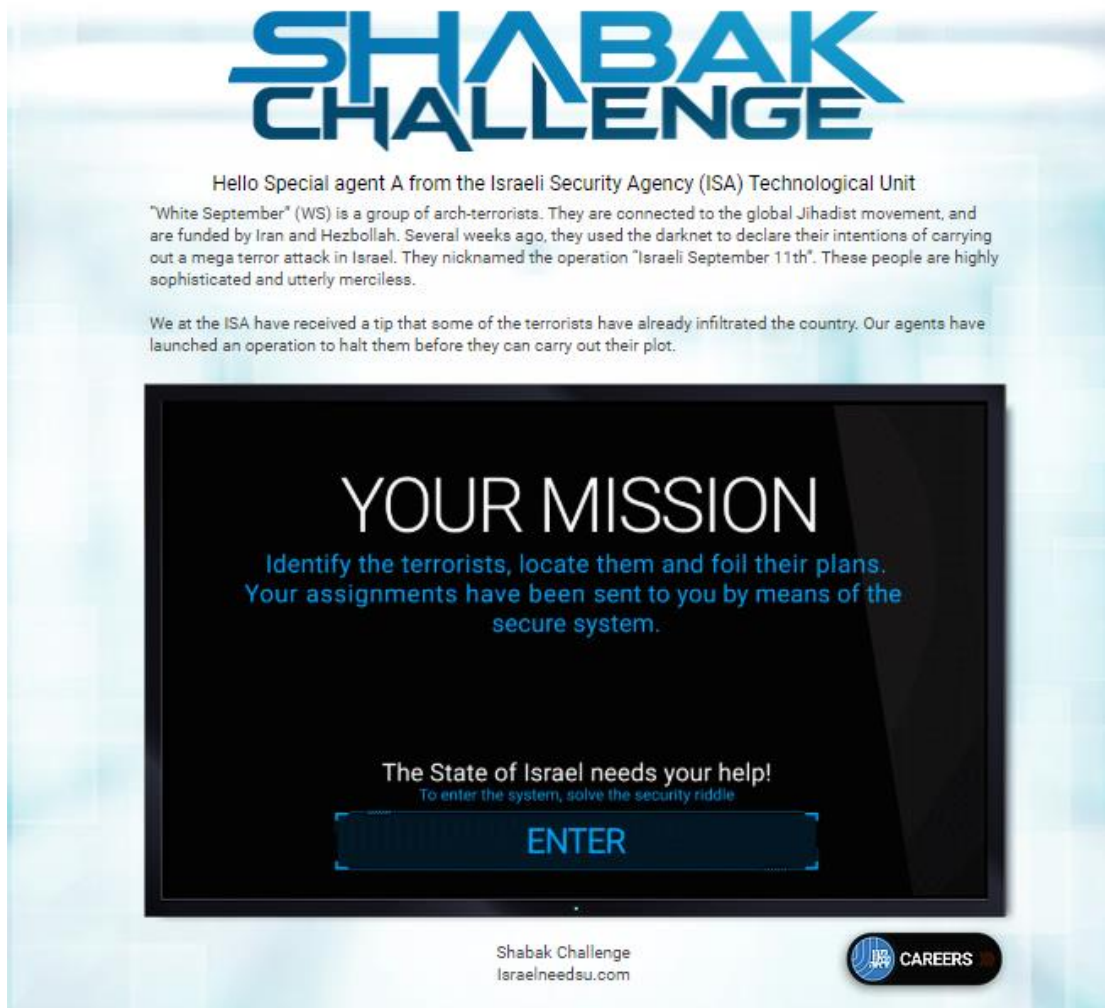
אתגר השב"כ – Shabak challenge

הקדמה

אתגר השב"כ הוא אתגר שפרסם השב"כ במטרה לעודד גיוס לשורות הארגון.

התחלה:

יאללה Israelneedsu.com



SHABAK CHALLENGE

Hello Special agent A from the Israeli Security Agency (ISA) Technological Unit

"White September" (WS) is a group of arch-terrorists. They are connected to the global Jihadist movement, and are funded by Iran and Hezbollah. Several weeks ago, they used the darknet to declare their intentions of carrying out a mega terror attack in Israel. They nicknamed the operation "Israeli September 11th". These people are highly sophisticated and utterly merciless.

We at the ISA have received a tip that some of the terrorists have already infiltrated the country. Our agents have launched an operation to halt them before they can carry out their plot.


YOUR MISSION

Identify the terrorists, locate them and foil their plans.
Your assignments have been sent to you by means of the secure system.

The State of Israel needs your help!
To enter the system, solve the security riddle

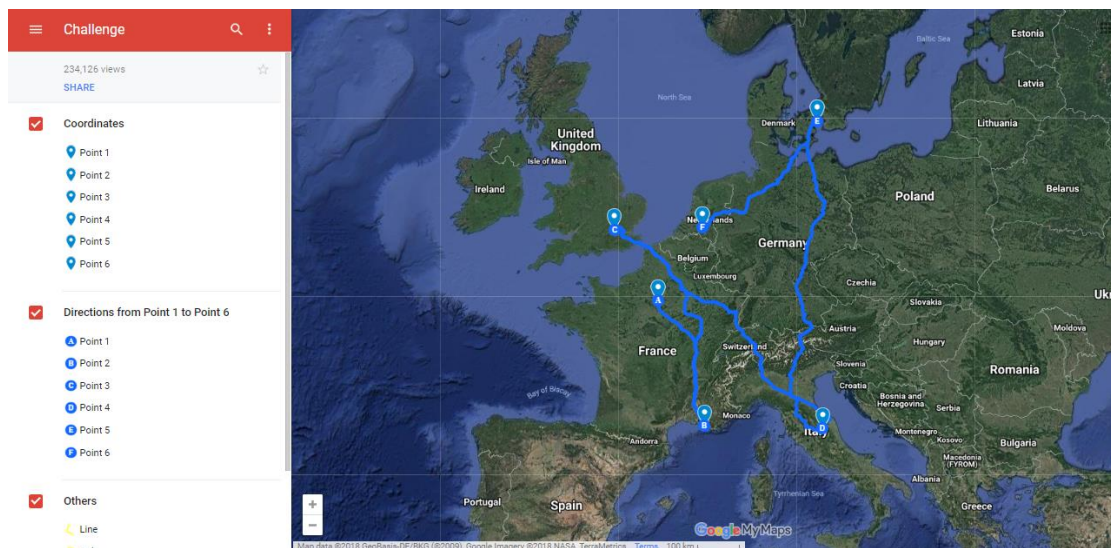
ENTER

Shabak Challenge
Israelneedsu.com

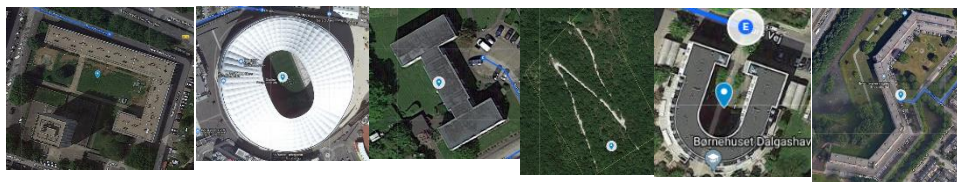




אם נלחץ על הסמל נופנה למסלול ב google maps



נתבונן מקרוב על נקודות הציון של המסלול.



וקיבלנו JOINUS

חלק שני:

בחלק הזה ניתן לבחור מסלול מארבעה מסלולים (או את ארבעתם)

WELCOME AGENT A!

Below are your technological assignments.
Completing an assignment will lead you to the next one, while providing us with critical information.
To begin, choose your field of expertise:

EMBEDDED SOFTWARE

SIGNAL PROCESSING

HARDWARE

SOFTWARE & DATA SCIENCE

israelneedsu.com

בחרתי להתמקד ב embedded software.

STAGE 1 2

CAREERS

Cat and Mouse

A routine counter-surveillance check of a senior Minister of Defense's vehicle revealed an electronic device in the undercarriage. We suspected it was a tracking device, and sent it to the Technological Department for an in-depth analysis.

After reverse-engineering the product, the following information was uncovered:

- A partial scheme of the electrical circuit and its components (see the attached electronic_scheme.pdf file).
- A disassembly of the code programmed to the micro controller (see the attached program.c file).
- The memory dump of the external memory component (see attached external_mem_dump.bin file).

Earlier that week, we had intercepted a suspicious SMS sent by a suspected member of White September. The suspect's message read "package received".

The message was received on 30/10/18, at 01:21 AM UTC. Our analysis team analyzed the data and determined that the message had most likely been sent by the same WS member who had installed the device in the senior official's vehicle. The analysts suspect he retrieved it from a WS dead drop.

Our engineers believe that when the message was sent, the device was online, and that therefore the location of the dead drop could potentially be extracted from it.

Your mission:
Find the exact coordinates of the dead drop.

Download files

Hash (SHA-256):

FC92343875186AE4CFCE3668ADB9428700AD1F9210DE051B63F905EB9670C715

Download

Answer

Latitude (dd.ddddd)

Longitude (dd.ddddd)

אנחנו מקבלים קובץ dump של זיכרון מהמכשיר, וכן את הקוד שיצר את הdump, וסכמה של מבנה הרכיב. אנחנו צריכים להבין איפה המכשיר היה בשעה הנתונה.

הפונקציה הראשונה שקופצת לעין save_to_flush

```
static void save_to_flash(uint8_t *buffer, uint32_t length){ /* Save buffer to flash at next empty address */ }
```

יופי, הפונקציה הזו היא זו ששומרת לזיכרון.

הפונקציה השנייה format_save

```
static uint32_t format_save(uint8_t *in_buffer, uint8_t a, uint32_t length, uint8_t *out_buffer)
{
    *out_buffer = a;
    *(uint32_t *)&out_buffer[1] = length;
    memcpy(&out_buffer[5], in_buffer, length);
    return length + 5;
}
```

כלומר מידע שנשמר לזיכרון נשמר מהצורה בייט אחד של סוג, וארבעה בייטים של אורך ולאחר מכן המידע.

בא נראה מתי הן נקראות

פעם אחת בפונקציה main

```
while (1)
{
    formatted_length = 0;
    size = uart_read(recv_buffer);

    if (size > 0)
    {
        res = parse(recv_buffer, size, parsed_buffers);
        if (reset == TRUE)
        {
            if(res != PARSER_TYPE_2) continue;

            formatted_length = format1(parsed_buffers, temp_buffer);

            reset = FALSE;
            a = TRUE;
        }
        else if (should_save == TRUE)
        {
            if(res != PARSER_TYPE_1) continue;

            formatted_length = format2(&parsed_buffers[1], temp_buffer);

            should_save = FALSE;

            a = FALSE;
        }

        if(formatted_length > 0)
        {
            save_length = format_save(temp_buffer, (a == TRUE) ? 0 : 1, formatted_length, save_buffer);
            save_to_flash(save_buffer, save_length);
        }
    }
}
```

כלומר המידע שנקרא מה uart נשמר בשני סוגים סוג 0 וסוג 1, מגניב¹.

פעם השניה בISR על INT2_vect

```
ISR(INT2_vect)
{
    static uint8_t interrupt_buffer[MAX_SAVE_BUFFER_SIZE];
    uint8_t *temp_buffer;
    uint8_t a;
    uint32_t length;
    if (PIND & 0x01)
    {
        is_triggered = TRUE;
        a = 2;
    }
    else
    {
        is_triggered = FALSE;
        a = 3;
    }

    length = format_save(temp_buffer, a, 0, interrupt_buffer);
    save_to_flash(interrupt_buffer, length);
}
```

ISR² – דרך להגדיר callback כשמתרחשת פסיקה.

אם כן אנחנו רואים עוד שני סוגים של מידע שנשמר שני פסיקות מסוג 2,3 שניהם ללא מידע.

¹ נשים לב כי savebuffer הוקצה בגודל 20 ולא נבדק כי format_lenght אכן קטן מגודל המערך, אומנם בקוד הנוכחי זו אינה בעיה, אבל ככה מתחילות חולשות בקוד (בפונקציה format_save מועתק מידע למערך לפי האורך המועבר).

² <https://techterms.com/definition/isr>

כעת להבין מהם בדיוק מידע מהסוג 0,1 ראינו כי המידע מתפנח בפונקציות

```
static uint32_t format1(uint8_t **in_buffer, uint8_t *out_buffer)
{
    *(int *)out_buffer = atoi(in_buffer[0]);
    out_buffer += sizeof(float);
    *(int *)out_buffer = atoi(in_buffer[8]);
    return 2 * sizeof(float);
}

static uint32_t format2(uint8_t **in_buffers, uint8_t *out_buffer)
{
    char val1_str[10] = { 0 };
    for (uint8_t i = 0; i < 3; i += 2)
    {
        memset(val1_str, 0, sizeof(val1_str));
        uint8_t *pos = strchr(in_buffers[i], '.');
        memcpy(val1_str, in_buffers[i], pos - in_buffers[i] - 2);
        uint16_t val1 = atoi(val1_str);
        float val2 = atof(pos - 2);
        float res = val1 + (val2 / 60);
        if (*in_buffers[i + 1] == 'W' || *in_buffers[i + 1] == 'S')
            res *= -1;
        *(float *)out_buffer = res;
        out_buffer += sizeof(float);
    }
    return 2 * sizeof(float);
}
```

סוג 0 (format1) הוא מסוג של שני שלמים.

סוג 1 (format2) מכיל מידע של שני float .

במקום לנסות להבין מה בדיוק הפונקציות האלו מפענחות, אפשר לראות מה נשמר בזיכרון.

```
(125000, 281018)
(32.005531311035156, 34.88541030883789)
```

אם כן סוג 0 נראה כמו זמן – תאריך ושעה, ואכן אם נסתכל בפונקצית main נראה שמשתנה משתנה בשם reset כלומר סוג 0 זה זמן בו היה reset למכשיר.

וסוג 1 נראה כמו נ"צ gps .

כרגע יש לנו זמן התחלה, נשאר להבין כל כמה זמן נשמרת נ"צ, ניתן לראות כי נ"צ נשמר או"א should_save הוא TRUE. הוא משתנה לtrue רק ב

```
ISR(TIMER1_COMPA_vect)
{
    if (++counter == counter_max_val)
    {
        should_save = TRUE;
        counter = 0;
        counter_max_val = (is_triggered == TRUE) ? 15 : 150;
    }
}
```

נבין מתי הפסיקה מתרחשת, ניתן להסיק כי מדובר בפסיקה שמתרחשת כל זמן קבוע.

נאסוף את הנתונים

```

configure_sysclk(); /* Assume system clock is 16 MHz */
static void configure1(void)
{
    cli();
    should_save = FALSE;
    counter = 0;
    TCCR1A = 0;
    TCCR1B = 0;
    TCNT1 = 0;
    OCR1A = 62499;
    TCCR1B |= (1 << WGM12);
    TCCR1B |= (1 << CS12) | (1 << CS10); // prescaler == 1024
    TIMSK1 |= (1 << OCIE1A);
    sei();
}

```

קיבלנו כי הפסיקה מתרחשת כל 4 שניות³. נשים לב שבכל פעם אנחנו מקדמים את counter עד counter_max_val שהוא משתנה כתלות האם אנחנו אחרי פסיקה מסוג 2 או מסוג 3, כאשר max מאותחל ל75.

לסיכום, כל 4 שניות counter יגדל עד לערך ה max שהוא 15 או 150 כתלות במצב הפסיקה, ואז תישמר מדידה אחת.

אם כן יש זמן התחלה (אתחול המכשיר), זמן התקדמות וזמן סיום, נריץ וסיימו.

location: (32.218929290771484, 35.232261657714844)

בשלב הזה נתקעתי במשך הרבה זמן, מתברר שהמיקום שציפו לקבל הוא בעיגול 5 ספרות לאחר הנקודה ולכן התשובה היא (32.21893, 35.23226). מבאס, הייתי מצפה שתתקבל גם התשובה ללא עיגול.

המיקום⁴:



³ ע"פ <http://www.avrbeginners.net/architecture/timers/timers.html>

⁴ קישור למסלול המלא goo.gl/H3zCx5 וכסו"כ שב"כ הייתי יותר מודאג שאיש WS התחיל בנתב"ג (אולי הוא בדיוק חזר מהטיול באירופה של השאלה הקודמת) עבר בכנסת ונכנס לקריה.

SHABAK CHALLENGE



You have completed challenge **1**

Next


STAGE 1 2

Code Red

We have uncovered some suspicious online traffic and identified communication between two IP addresses. Both these addresses come from computers we believe belong to senior members of the White September organization (see attached file chat.pcap).

The Technological Department believes that the WS members are using an application designed for concealing messages. Based previously gathered intel, our analysts team assesses that WS is planning a terror attack against Israel in the near future. They believe the exact date was disclosed in the aforementioned communication.

Your Mission:
Reveal the exact date and time of the planned attack.



Download files

Hash (SHA-256):

BD5A2E96CBA3DA83534BF2E9E13C19F3D9146F939C2A6F06740C53D3ACA1BFD1

Download

Answer

DD/MM/YYYY

HH:MM

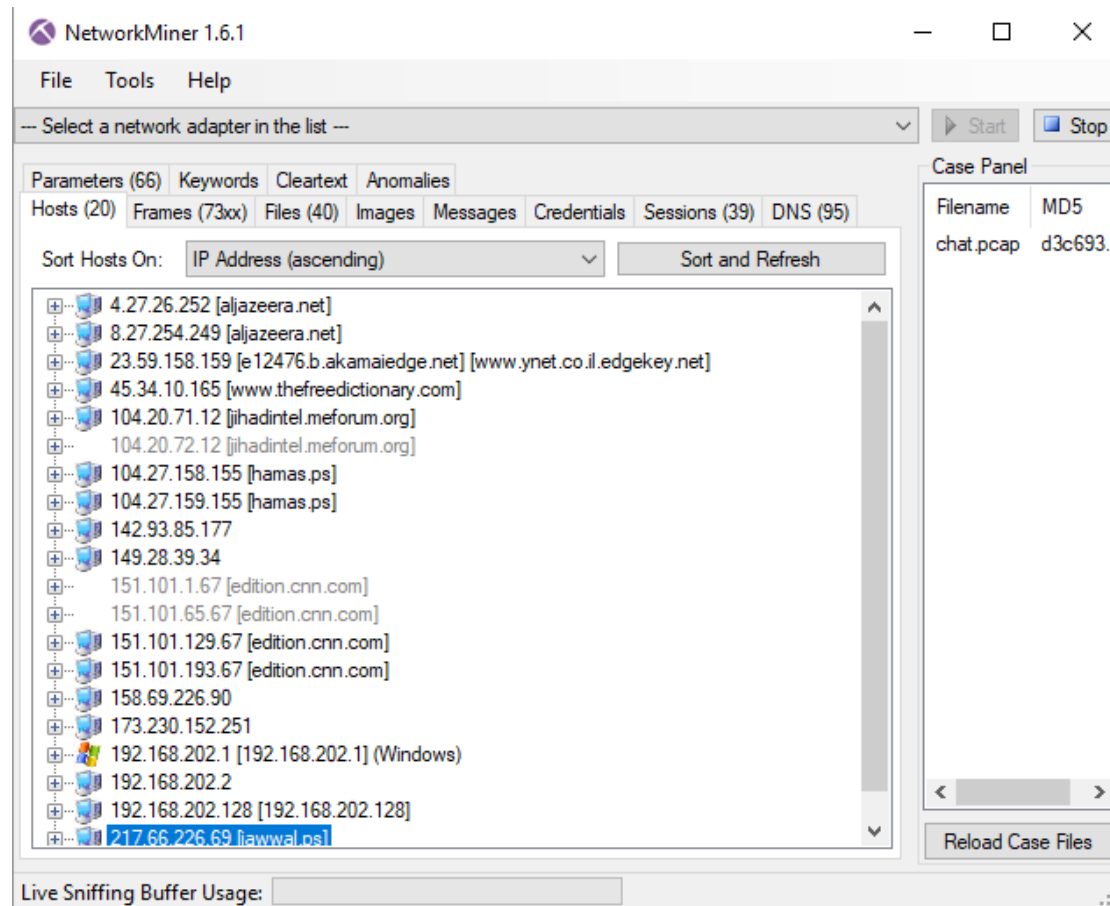
☐ I'm not a robot

reCAPTCHA
Privacy - Terms

Submit

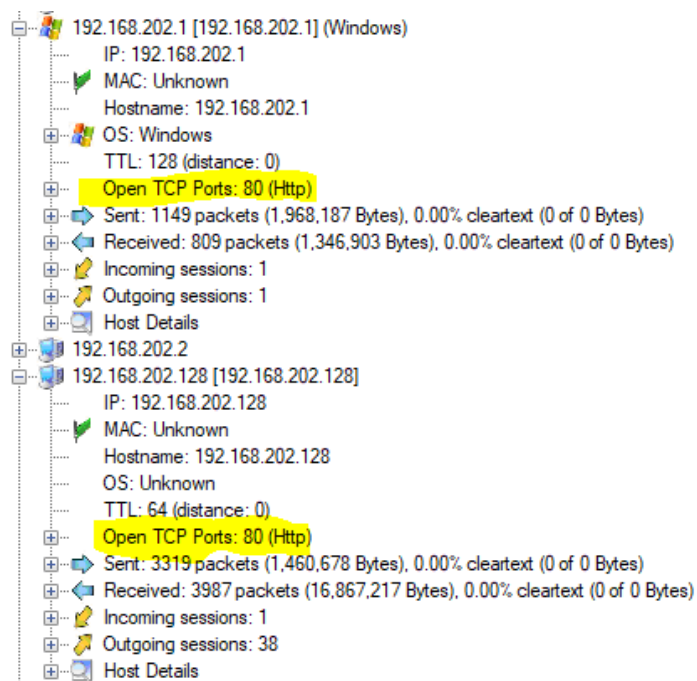
הקשר של התרגיל ל embedded לא ברור לי אבל גם השלב הזה היה כיף.

למעשה בשלב הזה אנחנו מקבלים קובץ pcap שמכיל את התקשורת נתחיל למפות את המחשבים שברשת לצורך כך נשתמש בתוכנה ⁵ networkMiner.



אנחנו רואים הרבה מאוד כתובות IP, כנראה אתרים שגולשים אליהם, כאשר שלוש כתובות צדות את העין 192.168.202.1/2/128 הנחתי כי זו הרשת של הws. הניחוש הראשוני כי המחשבים של x.x.x.1 ו x.x.x.2 הם שני המחשבים של החברים בארגון, אבל לצערי אין ולו הודעה אחת שעוברת בין שני המחשבים האלו.

מתבוננות עמוקה יותר, ניתן לראות שגם ב x.1 וגם ב x.128 פתוח פורט 80 בפרוטוקול http



בעזרת ⁶wireshark נחלץ את הפקטות הרלוונטיות.

Expression	Info	Length	Protocol	Destination	Source	Time	No.
((p.src eq 192.168.202.128 and p.dst eq 192.168.202.1) or (p.src eq 192.168.202.1 and p.dst eq 192.168.202.128)) and http	(POST /sessions HTTP/1.1 (application/json 62	HTTP	192.168.202.128	192.168.202.1	37.093813	1443	
	HTTP/1.1 200 OK 148	HTTP	192.168.202.1	192.168.202.128	37.208269	4149	
	(POST /messages HTTP/1.1 (application/json 3337	HTTP	192.168.202.128	192.168.202.1	48.086333	3389	
	HTTP/1.1 200 OK 148	HTTP	192.168.202.1	192.168.202.128	48.171655	3391	
	(POST /messages HTTP/1.1 (application/json 2492	HTTP	192.168.202.128	192.168.202.1	70.899641	2290	
	HTTP/1.1 200 OK 148	HTTP	192.168.202.1	192.168.202.128	71.889111	2375	
	(POST /messages HTTP/1.1 (application/json 246	HTTP	192.168.202.128	192.168.202.1	87.787596	2765	
	HTTP/1.1 200 OK 148	HTTP	192.168.202.1	192.168.202.128	88.783369	2831	
	(POST /messages HTTP/1.1 (application/json 439	HTTP	192.168.202.128	192.168.202.1	104.123943	3606	
	HTTP/1.1 200 OK 148	HTTP	192.168.202.1	192.168.202.128	104.265463	3614	
	(POST /messages HTTP/1.1 (application/json 1045	HTTP	192.168.202.128	192.168.202.1	130.465066	4180	
	HTTP/1.1 200 OK 148	HTTP	192.168.202.1	192.168.202.128	131.205115	4201	
	(POST /messages HTTP/1.1 (application/json 62	HTTP	192.168.202.128	192.168.202.1	143.098244	5036	
	HTTP/1.1 200 OK 148	HTTP	192.168.202.1	192.168.202.128	143.176715	5038	
	(POST /messages HTTP/1.1 (application/json 3130	HTTP	192.168.202.128	192.168.202.1	160.907818	5500	
	HTTP/1.1 200 OK 148	HTTP	192.168.202.1	192.168.202.128	161.371526	5502	
	(POST /messages HTTP/1.1 (application/json 1728	HTTP	192.168.202.128	192.168.202.1	191.230656	6411	
	HTTP/1.1 200 OK 148	HTTP	192.168.202.1	192.168.202.128	192.047481	6437	
	(POST /messages HTTP/1.1 (application/json 3146	HTTP	192.168.202.128	192.168.202.1	211.578369	7320	
	HTTP/1.1 200 OK 148	HTTP	192.168.202.1	192.168.202.128	211.739788	7322	

אם כן אנחנו רואים תקשורת בין שני מחשבים שמעבירים json אחד לשני.

נתבונן בהודעה הראשונה

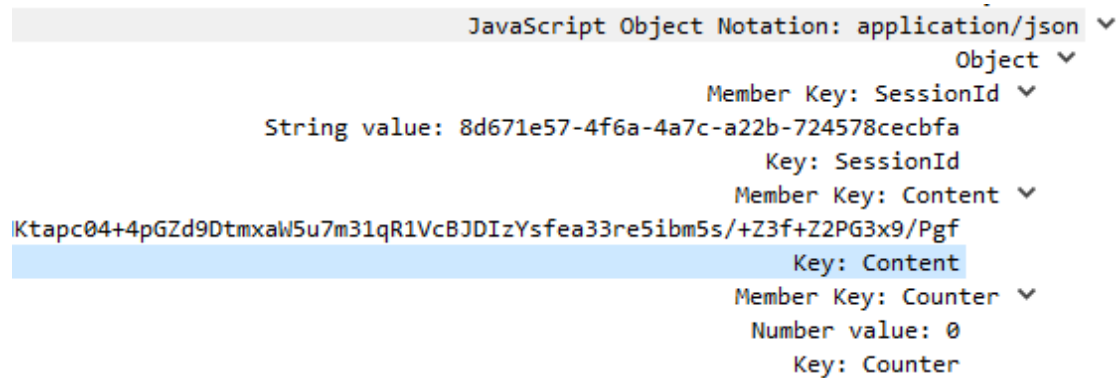
JavaScript Object Notation: application/json	Object
Member Key: Id	String value: 8d671e57-4f6a-4a7c-a22b-724578cecbfa
Key: Id	Member Key: Urls
Array	String value: https://static.wixstatic.com/media/57cf4c_afea1f0bb82348d9bdc24653ea3208f9~mv2.png
String value: https://static.wixstatic.com/media/57cf4c_9fa5cba479a24e73a24fb52163d9209b~mv2.png	String value: https://static.wixstatic.com/media/57cf4c_4080f95bf84349e5887042c3a06f7114~mv2.png
String value: https://static.wixstatic.com/media/57cf4c_0bf3bbad5f74409bad0a3a10b1dbd537~mv2.png	String value: https://static.wixstatic.com/media/57cf4c_0aa6e7ffcc024f7ba2b6611f72f2432d~mv2.png
String value: https://static.wixstatic.com/media/57cf4c_d9f88c5ddc93488d91ac03c56cc901ae~mv2.png	String value: https://static.wixstatic.com/media/57cf4c_56a9ed0fd9c84c98935307aebb4783f7~mv2.png
Key: Urls	

יש ID כנראה משמש כID session .

ומערך של כתובות לתמונות (אחלה תמונות רקע למחשב)



נתבונן בהודעה הבאה (וכן נראות גם שאר ההודעות)



אכן, המחזרות הראשונה שקיבלנו היא SessionID
לאחר מכן מתקבל התוכן מקודד בbase64, counter שמכיל את מספר ההודעה.
כעת נחלץ את התוכן, ומתקבלות תמונות הנראות זהות להודעות שנשלחו.

ההשערה הראשונה שמדובר באיזושהי בסטנוגרפיה⁷ בשלב הראשון נתחיל פשוט כora של המידע
בין שתי ההודעות (בשלב הזה ניסיתי לחלץ את המידע בעצמי, טעות!! תמיד יש ספריה בפיטון
שעושה כל מה שתרצה לעשות).

ואכן מתברר כי שתי התמונות זהות עד כדי

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00155770	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00155780	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00155790	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
001557A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
001557B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
001557C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
001557D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
001557E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
001557F0	0D	0A	41	68	6C	61	6E	2C	20	68	6F	77	20	61	72	65	..Ahlan, how are
00155800	20	79	6F	75	3F	0D	0A	0D	0A								you?....

מגניב!!! הצלחנו להוציא מחזרת ראשונה, נעדכן את סקריפט כך שנקבל רק את התווים השונים.

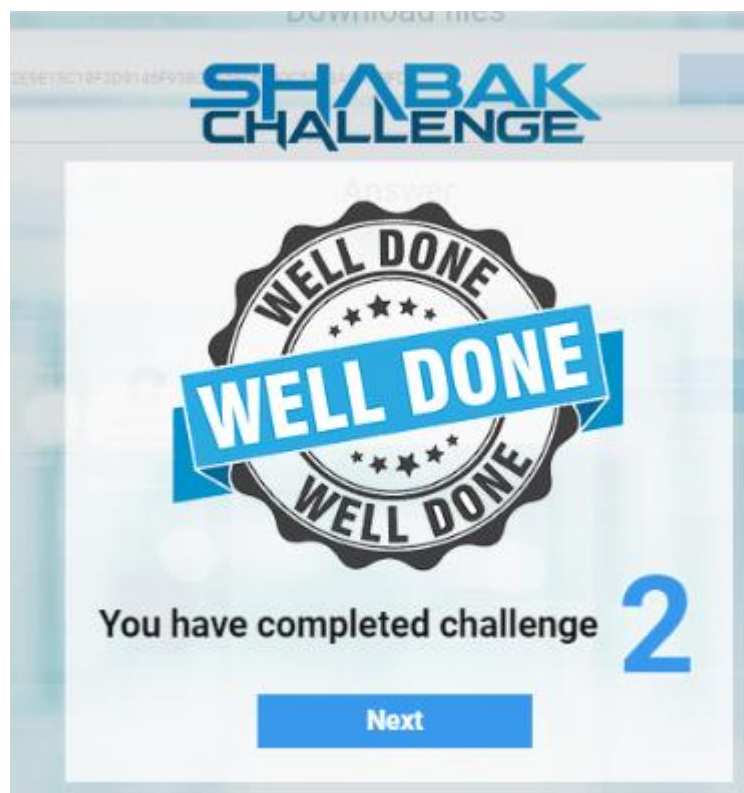
נריץ על כל התמונות

```
Ahlan, how are you?
@im`o!i`chch-!H&l!ghod/
Jmu"cpq">{mw=
Boobkvngvojoobk"#Klt#bqf#boo#wkf#aqlwkfqp<
Pla}$eva$ehh$a|gmpa`$bkv$jas$}aev#w$ara*
R`%dw`%dii`s`w|%uwjpa%jc%|jp+
Qc&qghr&rnc&vgtr@&ri&urgtr&gr&46<76&c~gerj@
Nitofkkfo+'ohwbarkk~'f'khs'ha'wbhwkb'pnkk'dhj(b
Af{`iddi`$(Iddi`}(Icjiz&
```

אנחנו כ"כ קרובים. מה פספסנו? נשים לב שלמעשה לא השתמשנו בcounter ננסה לקסור כל הודעה עם מספר ההודעה. ואכן נקבל

```
Ahlan, how are you?  
Ahlan habibi, I'm fine.  
How are you?  
Allahumdulillah! How are all the brothers?  
They are all excited for new year's eve.  
We are all very proud of you.  
We want the party to start at 20:10 exactly.  
Inshallah, hopefully a lot of people will come.  
Inshallah, Allahu Akbar.
```

אם כן הפצצה עתידה להתפוצץ ב 31/12/2018 בשעה 20:10
סך הכל סקריפט של 32 שורות כולל רווחים (ואפשר בהרבה פחות).
סיימנו



סיום:



בשלב הזה אנחנו מקבלים סרטון הממליץ לנו להגיש קו"ח באתר בצורה הרגילה. האמת, סיום טיפה מאכזב, בד"כ באתגרים מהסוג הזה מקבליים מייל מיוחד או מעין token שמאפשר להגיש את הקו"ח באופן שמסמל שסיימת את האתגר.

סיכום:

נהנתי מאוד לפתור את האתגר, למדתי הרבה לאורך כל הדרך. יאללה לאתגר הבא,