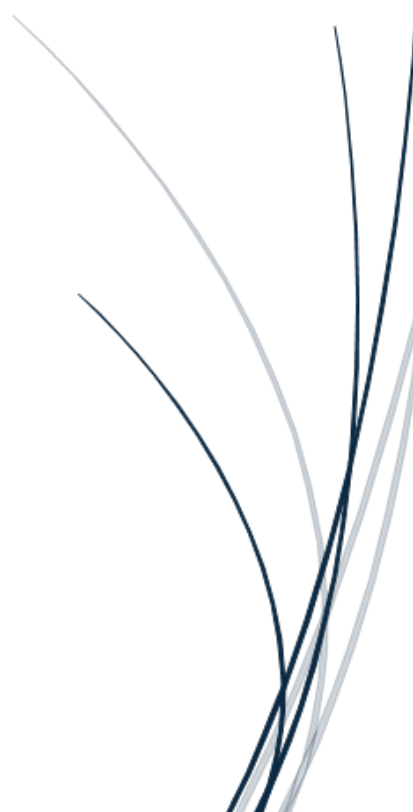


25.8.2025

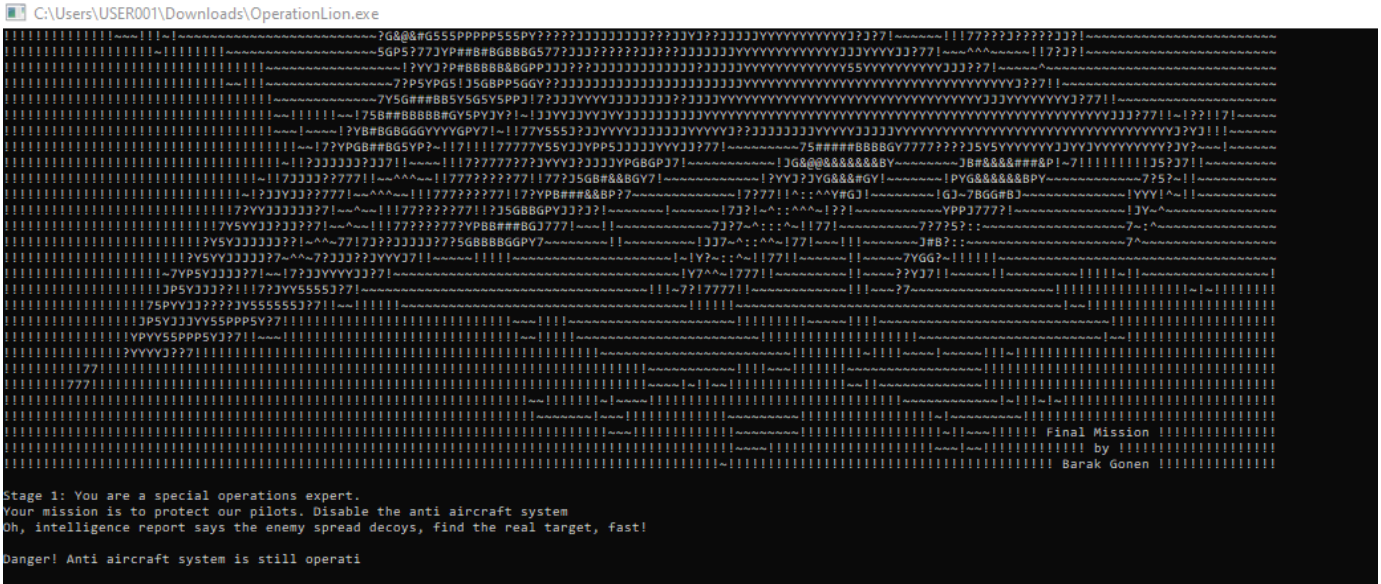
עבודה ברברסינג

שם: נעמי גולקין



שלב 1:

לתגור הרברסינג יש 3 שלבים לעשות על מנת לפתור אותו.
נתחיל בקובץ OperationLoin.exe שקיבלתי לתגור הרברסינג.
והנה התוצאה של מה שהרצתי בF9:



כעת אציג קוד של main:

```

argv= dword ptr 0Ch
envp= dword ptr 10h

push    ebp
mov     ebp, esp
sub     esp, 0Ch
mov     eax, __security_cookie
xor     eax, ebp
mov     [ebp+var_4], eax
mov     eax, off_409018 ; "....."
push    eax ; char
push    offset Format ; "%s\n"
call    sub_401040
add     esp, 8
push    700h ; dwMilliseconds
call    ds:Sleep
push    offset aStage1YouAreAS ; "Stage 1: You are a special operations e"
call    sub_401290
add     esp, 4
mov     ecx, ds:dword_405474
mov     dword ptr [ebp+var_C], ecx
mov     dx, ds:word_405478
mov     word ptr [ebp+var_C+4], dx
lea     eax, [ebp+var_C]
push    eax
call    len_word
add     esp, 4
call    sub_401330
xor     eax, eax
mov     ecx, [ebp+var_4]
xor     ecx, ebp ; StackCookie
call    @_security_check_cookie@4 ; __security_check_cookie(x)
mov     esp, ebp
pop     ebp
retn

_main endp

```

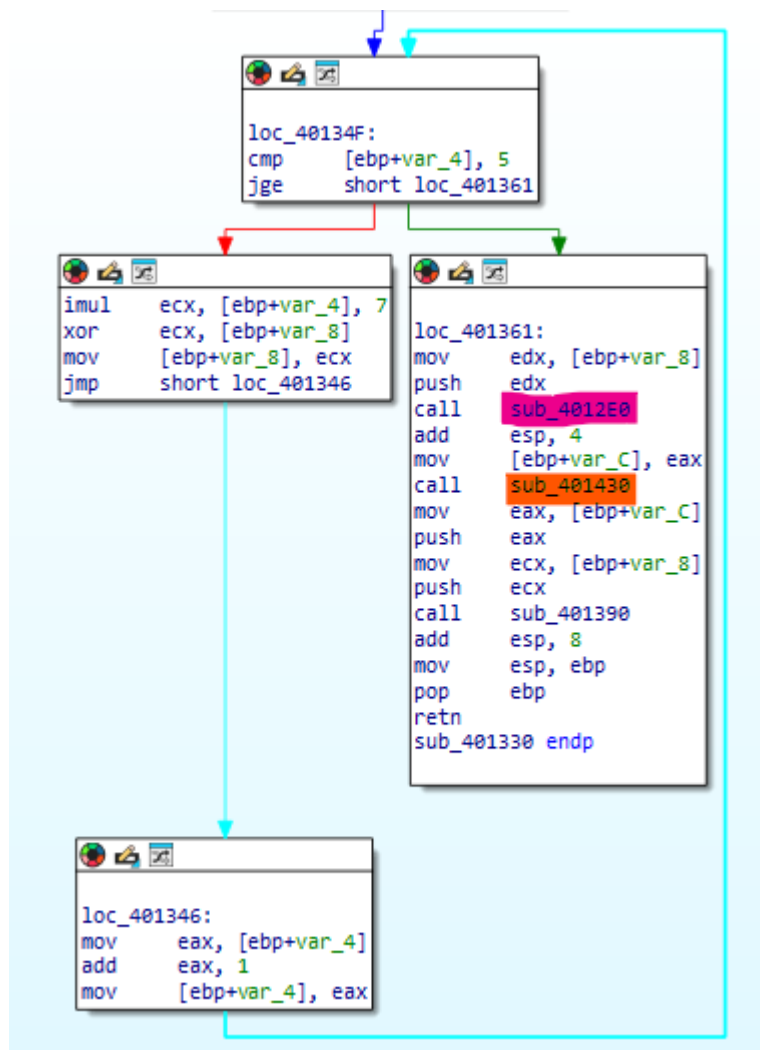
ניתן לראות פה בהתחלה של main את ההדפסה שראיתי מיד שהרצתי את הקובץ F9b.

ניקח את מה שיש פה בקוד בצבע **צהוב** וזה מתורגם לנו למילה **Radar** - תירגמתי ל ASCII מ Little Endian מ - rdata.

כעת נכניס את המילה הזאתי לתוך הפונק' **len_word** שמה יש XOR פעמיים על כל אות ואות מהמילה Radar - וכאשר, עושים XOR פעמיים זה משאיר אותי אותו דבר... $x \oplus a \oplus a = x$ וכעת, מתי הלופ של xor על xor נגמר? מתי שאנחנו נגיע שנגיע לסוף המילה Radar ואז, יחזור לנו על האורך של Radar דהיינו 5.

כעת ניכנס לתוך **sub_401330** בתחילת הפונק' יש שמה את המשתנה עם הערך 123 ועושים עליו איזה לולאת for שהתוצאה הסופית שלה זה המס' 123.

כעת מה שהולך אחרי ה for עדיין בתוך הפונק' **sub_401330** אציג מ - IDA:



הנה, בגמר ה for נלך לחלק שקורה אח"כ בפונק', ניכר שיש שמה קריאה לכמה פונקציות.
sub_4012E0 הפונקציה הראשונה

```
int __cdecl sub_4012E0(int a1)
{
    int v2; // [esp+0h] [ebp-4h]

    v2 = a1 % 3;
    if ( ! (a1 % 3) )
        return 2 * a1;
    if ( v2 == 1 )
        return a1 + 100;
    if ( v2 == 2 )
        return a1 - 50;
    return 0;
}
```

היא תיקח את המס' 123 שהיא קיבלה ב a1 ותפנה אותו לפי המקרה הספציפי שמתאים לו.
ופה זה מתאים למקרה שהמס' 123 מתחלק ב 3 ללא שארית ולכן, נחזיר את המס' עצמו כפול 2.
דהיינו המס' הוא 123 ולכן: $246 = 2 * 123$

וכעת, נפתח את הפונקציה השניה sub_401430
ישר על ההתחלה אני רואה נתיב C:\\ReversingCTF ושם של קובץ DroneAttack.txt ואז, יש אזור לקראת סוף הפונק' של הדפסה:

Stage 2: You are a jet fighter pilot

וכמובן ארצה להגיע לשמה כי ההדפסה הזאת מעידה על סיום שלב 1 ותחילת שלב 2.
אז קודם כל קצת ניסוי וטעייה

אנסה לשים את הנתיב הזה במחשב שלי + קובץ txt הזה.
והרצתי את הקובץ exe ואכן נכנס לי תוכן לתוך הקובץ DroneAttack.txt
וכמו כן, גם נוצר לי dll בשם AttackIRGC.dll
וכמובן, זה מה שהלך בהמשך הקוד.
והכי חשוב כרגע נכנסתי לשלב 2 - כי, הודפס לי
...Stage 2: You are a jet fighter pilot...

שלב 2:

בשלב זה יש לי dll וקובץ txt ופתחתי את הdll ב IDA וכמו כן, ב- CFF וזה הרגיש לא תקין, ולמה?

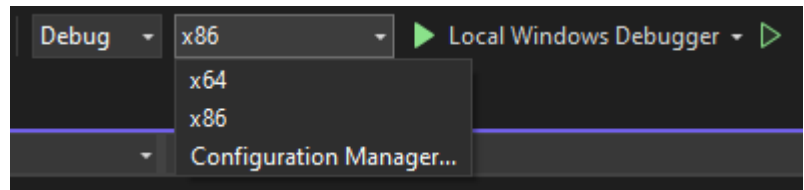
כי, קודם כל אין לנו את ה MZ המוכר הידוע שמסמל שזה קובץ dll תקין ודבר נוסף, בלט כאן מאוד המילה BOMB שהופיעה לאורך כל הdll. לאחר די זמן חשבתי שאולי זה קשור להודעה שהופיעה בתחילת שלב 2

Your mission: release bombs on IRGC headquarters

ניסיתי לחשוב אולי אני יסיר את כל המילה BOMB מהdll. אז, בניתי לזה קוד קטן בפיתון שיסיר לי את כל ה BOMB מה dll ומה שמתברר שהdll החדש שניסיתי ליצור לא הניב לי משהו יעיל מדי... אבל, מה שכן זה היה נראה די בטוח יש כאן משהו לעשות עם ה BOMB על הקובץ הdll הזה, כל השאלה מה לעשות... חשבתי על כיוון של הצפנות שאולי הקובץ txt משלב 1 מצפין את הdll יחד עם BOMB. לכן, ניסיתי קצת כל מיני דברים עם הצפנות ולהבין מה הולך ומה אפשר לעשות. ובסוף הכיוון הזה עם קובץ txt ירד. לאחר מכן, חשבתי שוב על העניין של הצפנות הרי יש לנו את הrsa שזה ההצפנה די בסיסית ננסה לעשות את זה עם BOMB במחזוריות על ה dll. אצרף את קוד בפיתון שהוא הוביל לי לdll תקין שמתחיל ב IMZ! וכמובן, שפתחתי את הdll ב IDA היה ניכר בהחלט שיש כאן דברים שנראים המשך ישיר לחלק 2.

```
1 def xor_file_with_key(input_path, output_path, key=b'BOMB'):  
2     with open(input_path, 'rb') as f:  
3         data = f.read()  
4  
5     decrypted = bytearray()  
6     key_len = len(key)  
7  
8     for i in range(len(data)):  
9         decrypted.append(data[i] ^ key[i % key_len])  
10  
11     with open(output_path, 'wb') as f:  
12         f.write(decrypted)  
13  
14  
15 xor_file_with_key(input_path: 'C:/ReversingCTF/AttackIRGC.dll', output_path: 'C:/ReversingCTF/AttackIRGCNew.dll')
```

עכשיו שיש לי dll חדש ותקין אז, אני רוצה להריץ אותו. כמובן, dll הוא לא קובץ exe ולכן, אריץ את הdll דרך קוד. עכשיו, המחשב שלי x64 והdll ב x32/x86 אז, נלך ל ++C ושמה בכל הרצה והרצה ניתן לבחור איך להריץ.



ואכן, ברגע שבחרתי x86 כל הבעיות של חוסר ההתאמה וההתחברות לdll נעלמו.
והנה הקוד בC++:

```
1  #include <windows.h>
2  #include <iostream>
3
4  typedef int(__cdecl* HackSecurityFunc)(int);
5
6  int main()
7  {
8      HMODULE hDll = LoadLibrary(L"C:\\ReversingCTF\\AttackIRGCNew.dll");
9      if (!hDll) {
10         std::cerr << "Failed to load DLL\n";
11         return 1;
12     }
13
14     HackSecurityFunc hack_security = (HackSecurityFunc)GetProcAddress(hDll, "hack_security");
15     if (!hack_security) {
16         std::cerr << "Failed to get function address\n";
17         FreeLibrary(hDll);
18         return 1;
19     }
20
21     int result = hack_security(8200);
22
23     std::cout << "Result from hack_security: " << result << std::endl;
24
25     FreeLibrary(hDll);
26     return 0;
27 }
28
```

אז קודם כל הנתיב לתיקייה עם הdll החדש והתקין עם הMZ.
שמה אני מנסה להתחבר אליו.

וכעת למה פניתי דווקא ל *hack_security*?
כי, קודם כל שנכנסתי לחיפוש מחרוזות בdll הזה
ישר ראיתי שמה את המחרוזת של **Stage 3**... אז ידעתי שכדי להדפיס את המחרוזת
ההצלחה הזאת של **Stage 3**
יהיה עלי לעשות ולעבור בהצלחה את הפונק' *hack_security*
כי, המחרוזת הזאת נמצאת בה.

(כמו כן, הפונק' *hack_security* נמצאת גם בקטגוריה של Exports בCFF, אז, כל
הסימנים מובילים ל *hack_security*)

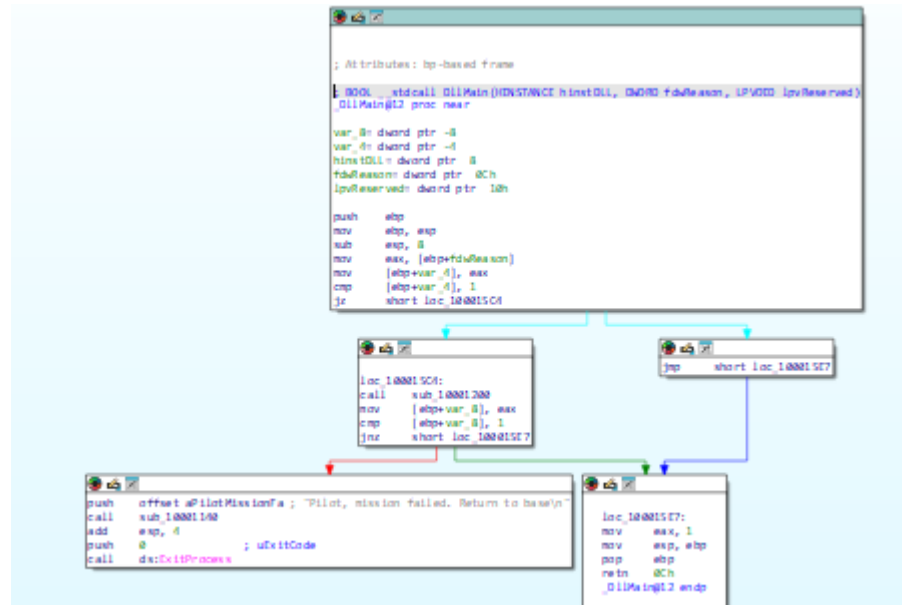
וכמובן בהתחלה כדי לדעת מה הולך עם hack_security נשלוף מIDA:

```
int __cdecl hack_security(int a1)
```

כלומר, מקבלת int מחזירה int, ו-__cdecl נרצה לעשות מצביע אליה מהקוד ב-C++.

ועכשיו קצת חקירה סטטית מה הולך ב hack_security, צריך להכניס את הערך 8200 כדי לחצות את התנאי שמה. כדי שנוכל להגיע לאזור של הדפסת **Stage 3**

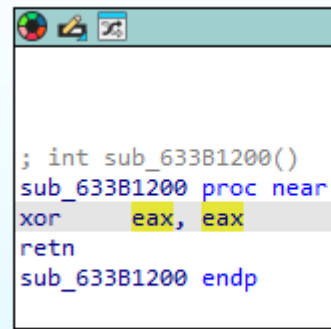
עכשיו, האם ניתן להריץ את הdll? התשובה לא. בגלל, שיש מנגנון של אנטי דיבאג, ולמה ישר ידעתי את זה? כי, שפותחים את הdll הוא ישר נפתח ככה:



נביא את זה בפסאודו קוד, ונראה שתמיד ב - sub_10001200 אם נחזיר בה תמיד 0, אז זה נדרוס ככה את המנגנון של אנטי דיבאג.

```
BOOL __stdcall DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
{
    if ( fdwReason == 1 && sub_10001200() == 1 )
    {
        sub_10001140("Pilot, mission failed. Return to base\n");
        ExitProcess(0);
    }
    return 1;
}
```

אז, הנה אציג את דריסת המנגנון ותמיד ארצה להחזיר 0 כדלהלן:



```

; int sub_633B1200()
sub_633B1200 proc near
xor    eax, eax
retn
sub_633B1200 endp

```

ורק אחרי שהסרתי את המגבלה של המנגנון של אנטי דיבאג
אז, הודפס שלב 3 בהצלחה.

```

Great job pilot, bombs hit IRGC.

Stage 3: Welcome cyber specialist.
Your mission : Penetrate the security system of the supreme leader.
The location of the enriched Uranium is stored there.
Your country depends on your skills. We COUNT on you. Good luck.
Enter code

```

הערה: שאלתי בקבוצה בטלגרם והמרצה אמר שאם זיהיתי בוודאות מנגנון אנטי דיבאג
אז מותר לפצפץ.

סיכום שלב 2: היה עלי ליצור dll חדש מהישן. אח"כ היה עלי להריץ את הdll החדש על
ידי בניית קוד בC++ והורדת מנגנון אנטי דיבאג.

שלב 3:

אצרף פה צילום מסך מהפסאודו קוד של הפונק' `hack_security` (אחרי שכתבתי שמות משמעותיים לפונק')

```
int __cdecl hack_security(int a1)
{
    FILE *file; // [esp+0h] [ebp-34h]
    int i; // [esp+4h] [ebp-30h]
    int j; // [esp+8h] [ebp-2Ch]
    char is_valid; // [esp+fh] [ebp-25h]
    char raw_buffer[4]; // [esp+10h] [ebp-24h] BYREF

    if ( a1 != 8200 )
    {
        print_message("Pilot, you missed the target. Use your SIGINT to find it\n");
        ExitProcess(0);
    }
    print_message("Great job pilot, bombs hit IRGC.\n\n");
    print_message(
        "Stage 3: Welcome cyber specialist.\n"
        "Your mission : Penetrate the security system of the supreme leader.\n"
        "The location of the enriched Uranium is stored there.\n");
    print_message("Your country depends on your skills. We COUNT on you. Good luck.\n");
    file = fopen("C:\\\\ReversingCTF\\DroneAttack.txt", "r");
    for ( i = 0; i < 8; ++i )
        read_hex(file, "%x", (char*)&raw_buffer[4 * i]);
    fclose(file);
    for ( j = 0; j < 8; ++j )
        global_counts[j] = count_bits_set(*(_DWORD *)&raw_buffer[4 * j]);
    print("Enter code\n");
    for ( is_valid = validate_code(); !is_valid; is_valid = validate_code() )
        print("Wrong code\n");
    print_message("Great work hero, you hacked the system. Prepare for a message from your instructor\n\n");
    return final_message();
}
```

פה בהתחלה אני רואה את ההצלחה משלב 2 ותחילת עבודה על שלב 3.
נסביר פה בנקודות מה הולך:

1. יש לנו לולאת `for` שבה אני שולפת את הערכים מ `DroneAttack.txt`
כל 4 תווים - HEX מעתיקה אותם ל `raw_buffer`

2. אחר כך, אני עוברת ללולאה ונכנסת לפונק' `count_bits_set` ששמה סופרת כמה ביטים 1 יש בכל מס' מ `raw_buffer`.

בסופו של דבר זה נכנס לי לתוך `global_counts`:
[9,10,9,8,8,8,11,8]

והנה גם הקוד בזמן אמת (אנליזה דינאמית) מראה לי מה יש בתוך `global_counts`:
[9,10,9,8,8,8,11,8]

וזה כמובן אחרי ששמתי breakpoint על המילים **Enter code**

כדי שאוכל מיד לראות את מה שהולך בתוך **global_counts** (הערה: לפעמים במהלך ה writeUp הכתובות של הטבלאות/משתנים עשויות להשתנות בין ריצות של קובץ exe בגלל הזזות, בכל מקרה הכוונה לאותה מיקום)

```
.data:72404378 global_counts dd 9 ; DATA XREF:
hack_security+DB↑w
.data:7240437C db 0Ah
.data:7240437D db 0
.data:7240437E db 0
.data:7240437F db 0
.data:72404380 db 9
.data:72404381 db 0
.data:72404382 db 0
.data:72404383 db 0
.data:72404384 db 8
.data:72404385 db 0
.data:72404386 db 0
.data:72404387 db 0
.data:72404388 db 8
.data:72404389 db 0
.wrdata:7240438A db 0
.data:7240438B db 0
.data:7240438C db 8
.data:7240438D db 0
.data:7240438E db 0
.data:7240438F db 0
.data:72404390 db 0Bh
.data:72404391 db 0
.data:72404392 db 0
.data:72404393 db 0
.data:72404394 db 8
.data:72404395 db 0
.data:72404396 db 0
.data:72404397 db 0
```

קצת הסברים אז, חיברתי את dll לקוד שלי ב ++C בזמן אמת כדי שאוכל לדבאג את dll ולהריץ אותו (מה שלא היה נצרך עדיין לשלב 2 - לדבאג בזמן אמת)

צריך להריץ את dll וגם את ה exe של הקוד שלי ב ++C בהרשאות מנהל. וכמו כן, צריך להריץ את ה exe - כלומר ישיר דרך ה console

אכנס ל IDA ושמה אלך ל-IDA Debugger-> Attach to process
ואחפש מתוך רשימה את exe הספציפי שלי שבו יש את הקוד ב++C.

עד שאמצא ואתחבר ל exe מתוך הרשימה ב IDA אז, הוספתי לקוד ב++C
בשורה הראשונה את השורה הבאה: `std::cin.get();`
ולמה? כדי, שימתין לי עד שאני מחברת את ה IDA ל exe
באותה מידה יכולתי לשים שמה `sleep`
(ואם לא הייתי עושה את זה מה היה יכול לקרות? יש מצב גדול שפשוט exe היה פשוט
רץ לי מהר ואז לא הייתי יכולה לדבאג את הקוד שנמצא די בהתחלה.)

3. לאחר מכן, יש לנו את ההדפסה **Enter code**
צריכה להכניס קוד כמו שצריך כי אחרת יודפס **Wrong code**

4. עכשיו ניכנס לתוך **valid_code** להבין יותר. ואסביר מה עשיתי כדי לדעת מה הסיסמא:

```
int current_value; // [esp+38h] [ebp-A0h]
_BYTE *v8; // [esp+3Ch] [ebp-9Ch]
int i; // [esp+44h] [ebp-94h]
char is_valid; // [esp+48h] [ebp-8Dh]
_DWORD digit_seen[9]; // [esp+4Ch] [ebp-8Ch] BYREF
char temp_char; // [esp+70h] [ebp-68h] BYREF
_BYTE v13[99]; // [esp+71h] [ebp-67h] BYREF

is_valid = 1;
if ( sub_72401100("%s", (char)&temp_char) == 1
    && (v5 = v13, v8 = &v13[strlen(&temp_char)], v4 = v8 - v13, v8 - v13 == 8) )
{
    memset(digit_seen, 0, sizeof(digit_seen));
    for ( i = 0; i < 8; ++i )
    {
        if ( !isdigit(v13[i - 1]) || v13[i - 1] < 49 || v13[i - 1] > 56 )
        {
            is_valid = 0;
            return 0;
        }
        current_digit = v13[i - 1] - 48;
        if ( digit_seen[current_digit] )
        {
            is_valid = 0;
            return 0;
        }
        digit_seen[current_digit] = 1;
        order_user[i + 1] = current_digit;
    }
}
```

} A

} B

```

for ( current_value = 1; current_value < 8; ++current_value )
{
    current_val = dword_72404374[order_user[current_value]];
    next_val = dword_72404374[order_user[current_value + 1]];
    if ( next_val < current_val )
    {
        is_valid = 0;
        return 0;
    }
}
return is_valid;
}
else
{
    is_valid = 0;
    return 0;
}
}

```

} <

חלק A:

בהתחלה מוודאים שאני מכניסה בדיוק 8 תווים. אם לא, זה פסול.

חלק B:

אסור להכניס אותו מס' פעמיים ויותר.

יש לנו מערך שבודק האם יש לנו כפילויות ב-8 מספרים שנכניס,

כל תא במערך digit_seen מייצג האם המס' כבר הופיעה (0 = לא הופיע, 1 = כן).

אח"כ יש לולאה על כל 8 הספרות, קודם כל בודקת שאני מס' שהטווח שלו בין 1 ל-8

(בקוד ASCII בין 1 ל-8 זה בין 49 ל-56) אחרת, זה פסול.

אח"כ, ממירים כל ספרה ממחרזת למס' int

- אם המס' הופיע כבר במאגר אז זה פסול.

- אחרת, אם המס' הזה הופיע פעם ראשונה במאגר אזי, אכניס אותו למאגר ואציין

לידו 1 שזה יציין שזה מס' שכבר השתמשו פעם אחת.

- ובנוסף, שומרים את הספרה ברשימה order_user ששומרת מה user הכניס

לפי סדר הקלדתו. (תכף נראה מה זה יעזור לנו)

מה שאני מבינה שגם אם אין לי כח לכאורה למצוא את הפתרון, אני יכולה להפעיל קוד

בפייתון שמנסה לי את כל האופציות מ 1 עד 88888888

וזה נקרא Brute Force - אבל, זה לא בשבילנו:

חלק C:

הרי המשתמש הכניס קוד באורך 8 ספרות שונות, מ-1 עד 8.

זה נשמר ברשימה שנקראת order_user.

לכל ספרה מתוך order_user הולכים לטבלה dword_72404374 ומוציאים את הערך

המתאים.

לדוג': אם המשתמש הכניס את הספרה 3 אזי, ניגשים ל-

`dword_72404374[3]`.

עכשיו, נקודה חשובה:

בתחילת שלב 3 יש את המס' הבאים `[9,10,9,8,8,8,11,8]` בתוך `dword_72404378`. בגלל שהמשתמש תמיד יכול להכניס רק מספרים 1–8 (ולא מאפס), זה אומר שכל הגישה לטבלה `dword_72404374` בפונקציה פה מכוונת למעשה להשוואה מול הערכים האלו - `[9,10,9,8,8,8,11,8]`

אז, נחזור ונסכם הבדיקה שעושים:

עוברים על כל זוג ספרות רצופות מ-`order_user`,

נניח הספרות הן a ו-b.

שולפים:

`current_val = dword_72404374[a]`

`next_val = dword_72404374[b]`

ואז בודקים: האם `next_val >= current_val` ?

כלומר, חייב להיות סדר עולה (גדול שווה) בין הערכים, אחרת הקוד נכשל.

והנה הקוד לזה גם בפייתון לכל התהליך:

```

# The values extracted from the TXT file
table = [9, 10, 9, 8, 8, 8, 11, 8]

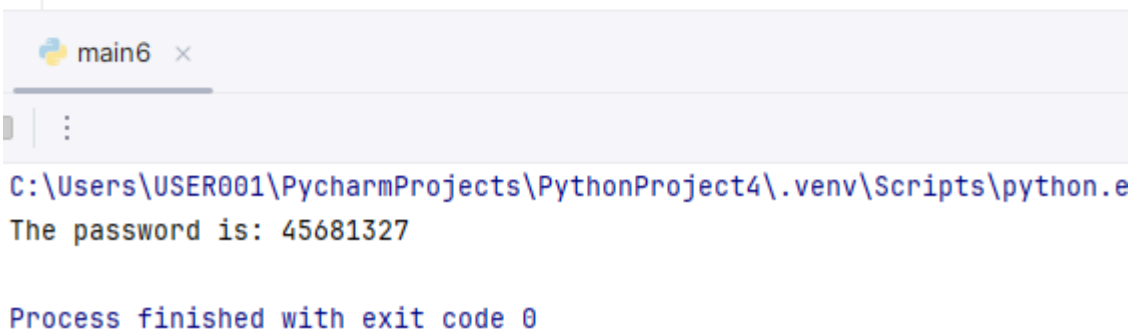
# Build pairs of (value, index)
pairs = [(value, index + 1) for index, value in enumerate(table)]

# Sort pairs by value (ascending order)
pairs.sort(key=lambda x: x[0])

# Collect indices in the sorted order
password = ''.join(str(index) for value, index in pairs)

print("The password is:", password)

```



```

main6 x
C:\Users\USER001\PycharmProjects\PythonProject4\.venv\Scripts\python.e
The password is: 45681327

Process finished with exit code 0

```

1. כל מס' מהרשימה [9,10,9,8,8,8,11,8]
נראה מה האינדקס של המיקום שלו ברשימה:
 $8 \rightarrow 8$, $11 \rightarrow 7$, $8 \rightarrow 6$, $8 \rightarrow 5$, $8 \rightarrow 4$, $9 \rightarrow 3$, $10 \rightarrow 2$, $9 \rightarrow 1$
 2. צריך לסדר אותם מקטן לגדול: (8 המס' הכי קטן ו-11 המס' הכי גדול)
 $8 = (4,5,6,8)$
 $9 = (1,3)$
 $10 = (2)$
 $11 = (7)$
 3. נאסוף בהתאם את כל המספרים שהם האינדקס
ולכן המחרוזת שנכניס תהיה: **45681327**
- ועכשיו כאן יש את ההדפסת הסיום האתגר:

45681327

Great work hero, you hacked the system. Prepare for a message from your instructor

Dear student, You reached the end. I am proud of you. Not many can do that.

This was only a game, but parts of the real operation were based on the knowledge that you learned.

I believe that you are part of the technological edge that keeps us here

I wish that you do great things in security, economy, technology and e

סיכום שלב 3: היה פה אנליזה דינאמית לדעת מה הסיסמא.

- סוף האתגר! -