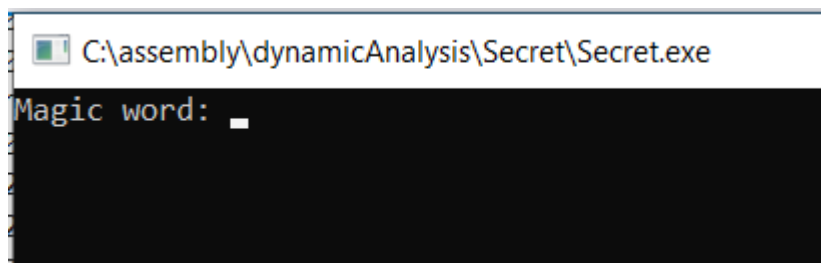


WriteUp "secret" exercise – Yishai Lutvak

נרץ את התרגיל secret.exe בwindbg



ראינו שהוא מדפיס "Magic word:"

לכן נחפש את המחזורות הזאת ונמצא מתי ניגשים לזיכרון שבו היא נמצאת כדי לאתר את החומר.

Disassembly window showing assembly code for 'Secret.exe'. The code includes instructions like 'mov', 'je', 'call', 'lea', 'sub', and 'mov' with their respective addresses and operands. The 'Locals' window shows the current state of local variables, including 'Secret' and 'Magic word'.

Breakpoints window showing the following breakpoints:

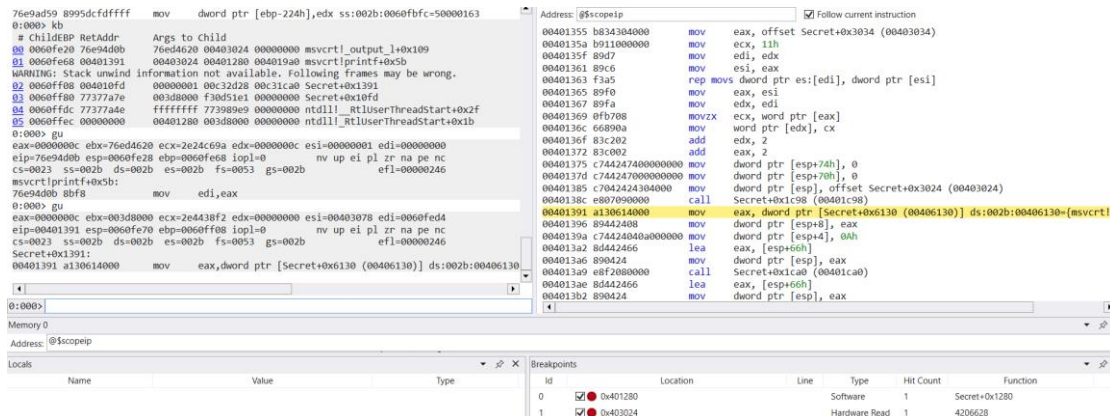
Id	Location	Line	Type	Hit Count	Function
0	0x401280		Software	1	Secret+0x1280

Disassembly window showing assembly code for 'Secret.exe'. The code includes instructions like 'mov', 'je', 'call', 'lea', 'sub', and 'mov' with their respective addresses and operands. The 'Locals' window shows the current state of local variables, including 'Secret' and 'Magic word'.

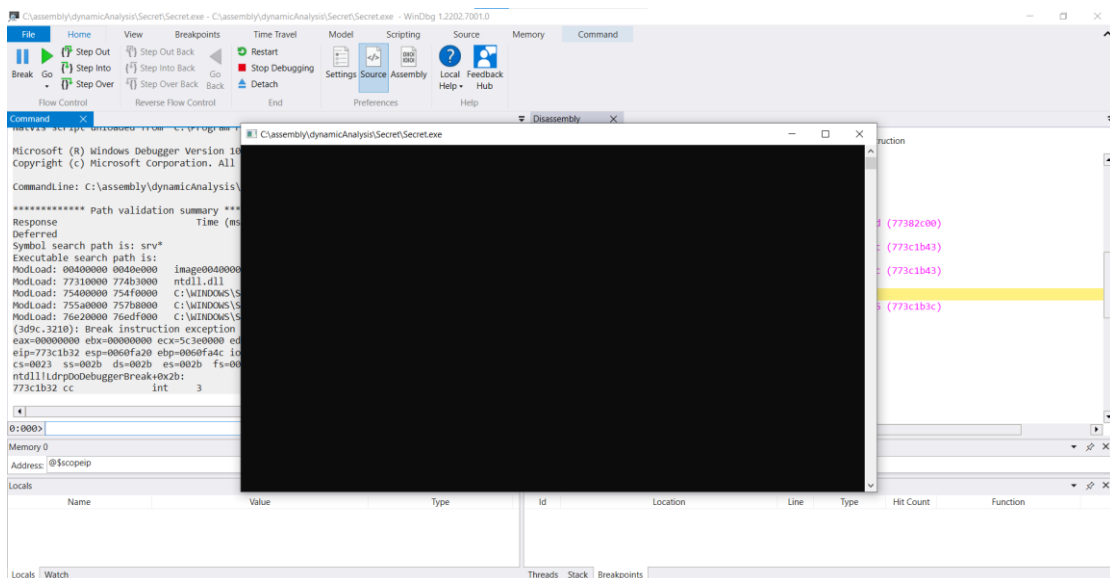
Breakpoints window showing the following breakpoints:

Id	Location	Line	Type	Hit Count	Function
0	0x401280		Software	1	Secret+0x1280
1	0x403024		Hardware Read	1	4206628

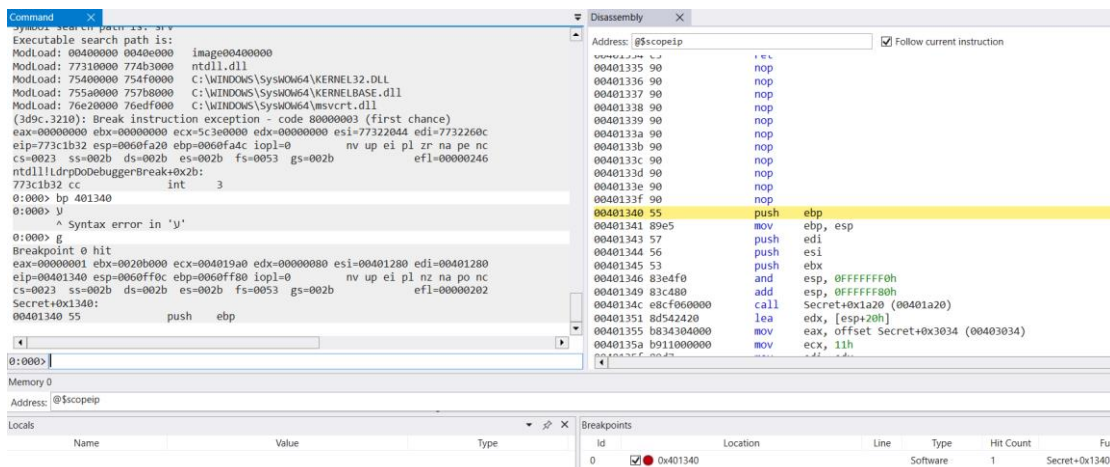
נבקש לראות את הקריאות למחשנית באמצעות הפקודה kb, אחר כך נזהה שכתובת המחזורת המודפסת "Magic word:" (403024) מועברת כארגומנט (בפעם הראשונה) שתי רמות למעלה יותר במחשנית, כאשר כתובת החזרה היא 401391. לכן נבצע פעמיים את הפקודה gu ונגיע לmain



נעשה restart



נשים קט בתחילת הmain (401340) ונריץ שוב.



נגלה כי הפקודה call בכתובת 40138C מדפיסה למסך את ההודעה "Magic word:".

The screenshot shows a debugger window with assembly code on the left and a disassembly view on the right. The assembly code includes instructions like `mov ecx, edi`, `movzx ecx, word ptr [eax]`, `mov word ptr [edx], cx`, `add edx, 2`, `mov dword ptr [esp+74h], 0`, `mov dword ptr [esp+70h], 0`, `mov dword ptr [esp], offset Secret+0x3024 (00403024)`, `call Secret+0x1c98 (00401c98)`, and `mov eax, dword ptr [Secret+0x6130 (00406130)]`. The disassembly view shows the same instructions with comments. A small window titled "C:\assembly\dynamicAnalysis\Secret.exe" is open, displaying the text "Magic word:".

אחר כך נגלה כי הפקודה call בכתובת 4013A9 מקבלת קלט מהמשתמש

נקליד סתם משהו – "aaaaaaaa" (10 פעמים a)

The screenshot shows a debugger window with assembly code on the left and a disassembly view on the right. The assembly code includes instructions like `mov ecx, edi`, `movzx ecx, word ptr [eax]`, `mov word ptr [edx], cx`, `add edx, 2`, `mov dword ptr [esp+74h], 0`, `mov dword ptr [esp+70h], 0`, `mov dword ptr [esp], offset Secret+0x3024 (00403024)`, `call Secret+0x1c98 (00401c98)`, and `mov eax, dword ptr [Secret+0x6130 (00406130)]`. The disassembly view shows the same instructions with comments. A small window titled "C:\assembly\dynamicAnalysis\Secret.exe" is open, displaying the text "Magic word: aaaaaaaaaa".

ניתן לראות את הקלט שהכנסנו "aaaaaaaa" בכתובת (esp+66h) כלומר בכתובת 60FED6

(משום מה נכנסו רק 9 ולא 10 כמו שהכנסנו!)

The screenshot shows a debugger window with assembly code on the left and a disassembly view on the right. The assembly code includes instructions like `mov ecx, edi`, `movzx ecx, word ptr [eax]`, `mov word ptr [edx], cx`, `add edx, 2`, `mov dword ptr [esp+74h], 0`, `mov dword ptr [esp+70h], 0`, `mov dword ptr [esp], offset Secret+0x3024 (00403024)`, `call Secret+0x1c98 (00401c98)`, and `mov eax, dword ptr [Secret+0x6130 (00406130)]`. The disassembly view shows the same instructions with comments. A small window titled "C:\assembly\dynamicAnalysis\Secret.exe" is open, displaying the text "Magic word: aaaaaaaaaa".

נזהה שתי קפיצות שמעיפות אותנו לסוף התוכנית.

אחת בכתובת 4013BD ואחת בכתובת 4013FF:

```
Disassembly X
Address: @$scopeip [X] Follow current instruction
00401390 03442400 mov dword ptr [esp+0], eax
0040139a c744240a000000 mov dword ptr [esp+4], 0Ah
004013a2 8d442466 lea eax, [esp+66h]
004013a6 890424 mov dword ptr [esp], eax
004013a9 e8f2080000 call Secret+0x1ca0 (00401ca0)
004013ae 8d442466 lea eax, [esp+66h]
004013b2 890424 mov dword ptr [esp], eax
004013b5 e8ee080000 call Secret+0x1ca8 (00401ca8)
004013ba 83f803 cmp eax, 3
004013bd 0f8698000000 jbe Secret+0x145b (0040145b) [br=0]
004013c3 c744247c01000000 mov dword ptr [esp+7Ch], 1
004013cb eb19 jmp Secret+0x13e6 (004013e6)
004013cd 8d542466 lea edx, [esp+66h]
004013d1 8b44247c mov eax, dword ptr [esp+7Ch]
004013d5 01d0 add eax, edx
004013d7 0fb600 movzx eax, byte ptr [eax]
004013da 0fbec0 movsx eax, al
```

```
Disassembly X
Address: @$scopeip [X] Follow current instruction
004013da 0fbec0 movsx eax, al
004013dd 01442474 add dword ptr [esp+74h], eax
004013e1 8344247c01 add dword ptr [esp+7Ch], 1
004013e6 837c247c09 cmp dword ptr [esp+7Ch], 9
004013eb 7ee0 jle Secret+0x13cd (004013cd)
004013ed 8b442474 mov eax, dword ptr [esp+74h]
004013f1 2d4e030000 sub eax, 34Eh
004013f6 89442470 mov dword ptr [esp+70h], eax
004013fa 837c247014 cmp dword ptr [esp+70h], 14h
004013ff 7553 jne Secret+0x1454 (00401454) [br=1]
00401401 c744247800000000 mov dword ptr [esp+78h], 0
00401409 eb24 jmp Secret+0x142f (0040142f)
0040140b 8d542420 lea edx, [esp+20h]
0040140f 8b442478 mov eax, dword ptr [esp+78h]
00401413 01d0 add eax, edx
00401415 0fb610 movzx edx, byte ptr [eax]
00401418 8b442470 mov eax, dword ptr [esp+70h]
0040141c 31d0 xor eax, edx
0040141e 8d4c2420 lea ecx, [esp+20h]
00401422 8b542478 mov edx, dword ptr [esp+78h]
00401426 01ca add edx, ecx
00401428 8802 mov byte ptr [edx], al
0040142a 8344247801 add dword ptr [esp+78h], 1
```

אחר כך זה נראה שאין מה שיזרוק אותנו החוצה לסוף התכנית.

לכן ננסה לראות אם הוא מכניס אותנו לקטע שמעבר לכתובת 4013FF ואם לא אז פשוט נשנה ערך של הרגיסטר .eip.

```

Disassembly X
Address: @$scopeip [X] Follow current instruction
00401420 8002 mov     byte ptr [esp], al
0040142a 8344247801 add     dword ptr [esp+78h], 1
0040142f 8b5c2478 mov     ebx, dword ptr [esp+78h]
00401433 8d442420 lea     eax, [esp+20h]
00401437 890424 mov     dword ptr [esp], eax
0040143a e869080000 call    Secret+0x1ca8 (00401ca8)
0040143f 39c3 cmp     ebx, eax
00401441 72c8 jb     Secret+0x140b (0040140b)
00401443 8d442420 lea     eax, [esp+20h]
00401447 890424 mov     dword ptr [esp], eax
0040144a e861080000 call    Secret+0x1cb0 (00401cb0)
0040144f e80c080000 call    Secret+0x1c60 (00401c60)
00401454 b800000000 mov     eax, 0
00401459 eb05 jmp     Secret+0x1460 (00401460)
0040145b b800000000 mov     eax, 0
00401460 8d65f4 lea     esp, [ebp-0Ch]
00401463 5b pop     ebx
00401464 5e pop     esi
00401465 5f pop     edi
00401466 5d pop     ebp
00401467 c3 ret
00401468 6690 nop
0040146a 6690 nop

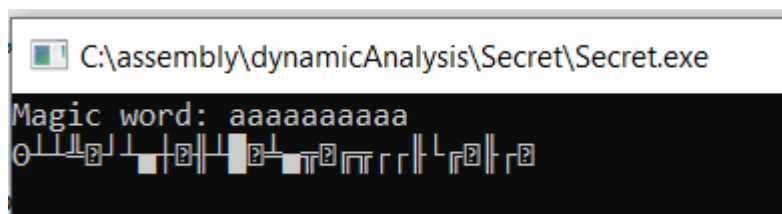
```

הוא העיף אותנו לסוף התכנית. בסה. לא נורא. נשנה את ערכו של הרגיסטר eip ל401401.

Command X
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b efl=00000293
Secret+0x13fa:
004013fa 837c247014 cmp dword ptr [esp+70h], 14h ss:002b:0060fee0=fffffbfa
0:000> p
eax=fffffbfa ebx=002cf000 ecx=0060fed6 edx=0060fed6 esi=00403078 edi=0060fed4
eip=004013ff esp=0060fe70 ebp=0060ff08 iopl=0 nv up ei ng nz na pe nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b efl=00000286
Secret+0x13ff:
004013ff 7553 jne Secret+0x1454 (00401454) [br=1]
0:000> p
eax=fffffbfa ebx=002cf000 ecx=0060fed6 edx=0060fed6 esi=00403078 edi=0060fed4
eip=00401409 esp=0060fe70 ebp=0060ff08 iopl=0 nv up ei ng nz na pe nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b efl=00000286
Secret+0x1409:
00401409 eb24 jmp Secret+0x142f (0040142f)
0:000> r eip=401401

Disassembly X
Address: @\$scopeip [X] Follow current instruction
004013da 0fbec0 movsx eax, al
004013dd 01442474 add dword ptr [esp+74h], eax
004013e1 834424c01 add dword ptr [esp+7Ch], 1
004013e6 837c247c09 cmp dword ptr [esp+7Ch], 9
004013eb 7ee0 jle Secret+0x13cd (004013cd)
004013ed 8b442474 mov eax, dword ptr [esp+74h]
004013f1 2d4e030000 sub eax, 34Eh
004013f6 89442470 mov dword ptr [esp+70h], eax
004013fa 837c247014 cmp dword ptr [esp+70h], 14h
004013ff 7553 jne Secret+0x1454 (00401454)
00401401 c744247800000000 mov dword ptr [esp+78h], 0
00401409 eb24 jmp Secret+0x142f (0040142f)
0040140b 8d542420 lea edx, [esp+20h]
0040140f 8b442478 mov eax, dword ptr [esp+78h]
00401413 01d0 add eax, edx
00401415 0fb610 movzx edx, byte ptr [eax]
00401418 8b442470 mov eax, dword ptr [esp+70h]
0040141c 31d0 xor eax, edx
0040141e 8d4c2420 lea ecx, [esp+20h]
00401422 8b542478 mov edx, dword ptr [esp+78h]
00401426 01ca add edx, ecx
00401428 8802 mov byte ptr [edx], al

נריך עד הסוף ונראה מה קורה...



הודפס משו לא ברור. כנראה יש פה איזה מחרוזת מוצפנת ושיבשנו את הקוד שמפענח אותה איפה שהוא בדרך.

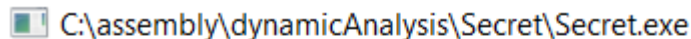
נעשה restart, נריך עוד פעם לאט לאט, ובכל פעם שצריך נשנה את eip כדי לבצע את הפקודה שהייתה צריכה להתבצע אם היינו מקלידים את מילת הקסם.

למשל, לפני השורה המסומנת (בתמונה הבאה) אני אכניס ל־eax את הערך 14h, כדי שההשוואה בשורה הבאה תצא נכונה...:

[illegible]

```
0:000> r eax=14h
0:000> r
eax=00000014 ebx=00366000 ecx=0060fed6 edx=0060fed6 esi=00403078 edi=0060fed4
eip=004013f6 esp=0060fe70 ebp=0060ff08 iopl=0         nv up ei ng nz ac po cy
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000293
Secret+0x13f6:
004013f6 89442470          mov     dword ptr [esp+70h],eax ss:002b:0060fee0=00000000
0:000> g
```

והנה הצלחנו לפענח את המחרוזת המוצפנת:

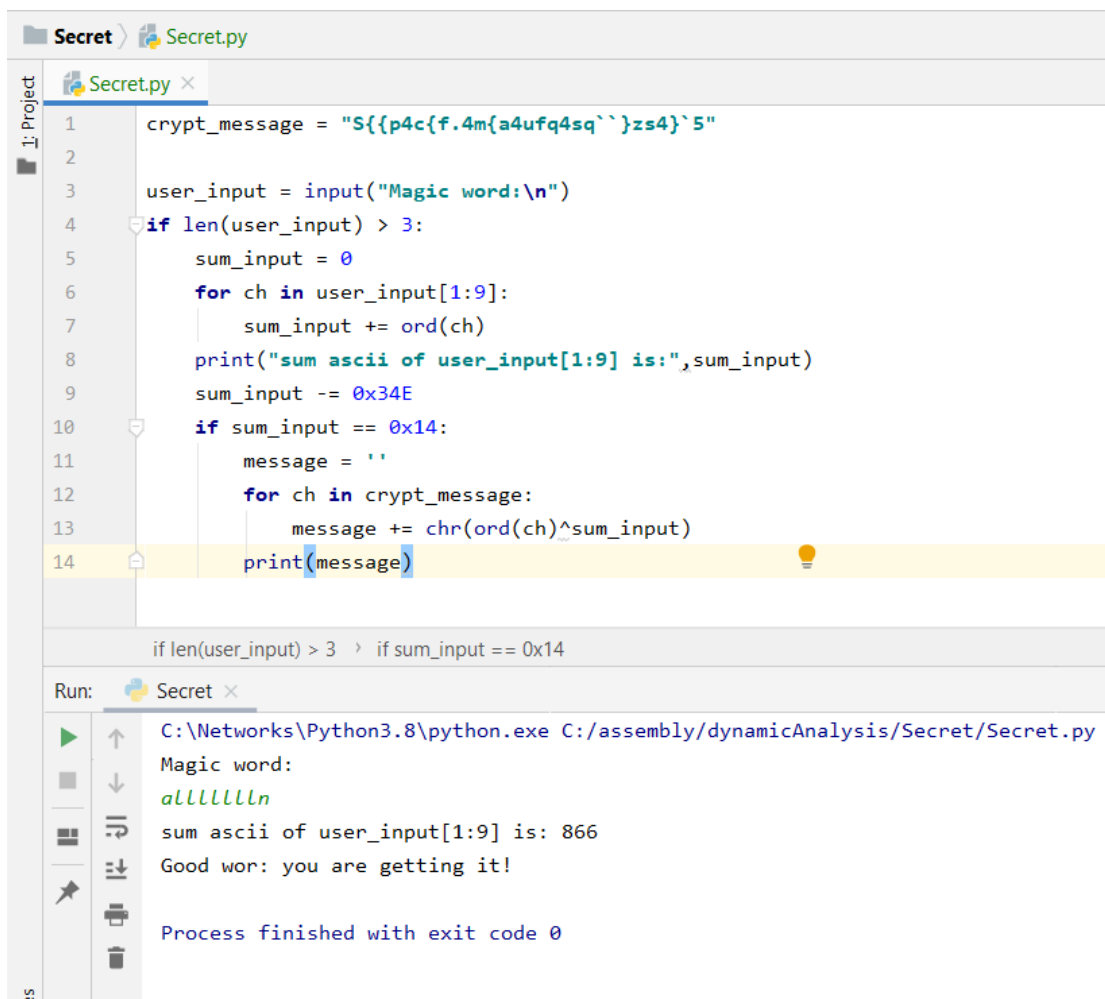


```
Magic word: aaaaaaaaaa
Good work you are getting it!
```


לאחר מחקר נוסף והבנה של מהלך הקוד הבנתי שהמספר הסודי הוא 866 והמפתח הוא 20

```
secret_main.asm
1 // main
2 //block1-----
3 00401340 push ebp
4 00401341 mov ebp, esp
5 00401343 push edi
6 00401344 push esi
7 00401345 push ebx
8
9 00401346 and esp, 0FFFFFFF0h
10 00401349 add esp, 0FFFFFF80h
11 0040134c call Secret+0x1a20 (00401a20)
12
13 00401351 lea edx, [esp+20h] // esp=0060fe70
14 00401355 mov eax, offset Secret+0x3034 (00403034) // edx=des_mes, # (esp+20h)~0060fe90~des_mes
15 0040135a mov ecx, 17 // eax=src_mes, # 00403034~src_mes~"S{p4c{f.4m{a4ufq4sq``}zs4}`5"
16 0040135f mov edi, edx // ecx=17
17 00401361 mov esi, eax // edi=0060fe90
18 00401363 rep movs dword ptr es:[edi], dword ptr [esi] // esi=00403078 edi=0060fed4, des_mes="S{p4c{f.4m{a4ufq4sq``}zs4}`5"
19 00401365 mov eax, esi // eax=00403078
20 00401367 mov ecx, edi // ecx=0060fed4
21 00401369 movzx ecx, word ptr [eax] // ecx=[eax]=49h
22 0040136c mov word ptr [edx], cx // [edx]=49h
23 0040136f add edx, 2 // edx=0060fed6
24 00401372 add eax, 2 // eax=0040307a
25 00401375 mov dword ptr [esp+74h], 0 // [esp+74h]~[0060fee4]=0
26 0040137d mov dword ptr [esp+70h], 0 // [esp+70h]~[0060fee0]=0
27 00401385 mov dword ptr [esp], offset Secret+0x3024 (00403024) // [esp] = 00403024, [00403024] = "Magic word:"
28 00401388 call Secret+0x1c98 (00401c98) // call msvcrt!printf, print "Magic word:"
29
30 00401391 mov eax, dword ptr [Secret+0x6130 (00406130)] // eax=[00406130]=76744ch
31 00401396 mov dword ptr [esp+8], eax // [esp+8]~[0060fe78]=76744ch
32 0040139a mov dword ptr [esp+4], 0Ah // [esp+4]~[0060fe74]=0Ah
33 004013a2 lea eax, [esp+66h] // eax=input, # [esp+66h]~0060fedc~input
34 004013a6 mov dword ptr [esp], eax // [esp]=input
35 004013a9 call Secret+0x1ca0 (00401ca0) // call msvcrt!fgets
36
37 004013ae lea eax, [esp+66h] // eax=input
38 004013b2 mov dword ptr [esp], eax
39 004013b5 call Secret+0x1ca8 (00401ca8) // call msvcrt!strlen, eax=len(input)
40
41 004013ba cmp eax, 3 // len(input) compare 3
42 004013bd jbe Secret+0x145b (0040145b) // if eax<=3 goto block10
43
44 //block2-----
45 004013c3 mov dword ptr [esp+7Ch], 1 // counter1=1, # [esp+7Ch]~[0060fee8]~counter1
46 004013cb jmp Secret+0x13e6 (004013e6) // goto block4
47
48 //block3-----
49 004013cd lea edx, [esp+66h] // edx=input
50 004013d1 mov eax, dword ptr [esp+7Ch] // eax=counter1
51 004013d5 add eax, edx // eax=counter1+input
52 004013d7 movzx eax, byte ptr [eax] // eax=000000@input[counter1],# (1..9)
53 004013da movsx eax, al // eax=0000/1111@a1
54 004013dd add dword ptr [esp+74h], eax // sum=sum+eax, # [0060fee4]~sum
55 004013e1 add dword ptr [esp+7Ch], 1 // inc counter1, # 2=>3=>4=>5=>6=>7=>8=>9=>10
56
57 //block4-----
58 004013e6 cmp dword ptr [esp+7Ch], 9 // counter1 compare to 9
59 004013eb jle Secret+0x13cd (004013cd) // if counter1<=9 goto block3
60
61 //block5-----
62 004013ed mov eax, dword ptr [esp+74h] // eax=sum, # sum need to be 866~362h
63 004013f1 sub eax, 34Eh, # 20~14h~00010100b~key_value // eax=sum-34Eh, # 20~14h~00010100b~key_value
64 004013f6 mov dword ptr [esp+70h], eax // key=eax, # [esp+70h]~[0060fee0]~key
65 004013fa cmp dword ptr [esp+70h], 14h // key compare 14h
66 004013ff jne Secret+0x1454 (00401454) // if eax!=14h goto block9
67
68 //block6-----
69 00401401 mov dword ptr [esp+78h], 0 // [0060fee8]=0, # [esp+78h]~[0060fee8]~counter2
70 00401409 jmp Secret+0x142f (0040142f) // goto block8
71
72 //block7-----
73 0040140b lea edx, [esp+20h] // edx=des_mes
74 0040140f mov eax, dword ptr [esp+78h] // eax=counter2
75 00401413 add eax, edx // eax=des_mes+counter2
76 00401415 movzx edx, byte ptr [eax] // edx=000000@des_mes[counter2]
77 00401418 mov eax, dword ptr [esp+70h] // eax=key_value
78 0040141c xor eax, edx // eax = (key_value xor 000000@des_mes[counter2])
79 0040141e lea ecx, [esp+20h] // ecx=des_mes
80 00401422 mov edx, dword ptr [esp+78h] // edx=counter2
81 00401426 add ecx, ecx // edx=counter2+des_mes
82 00401428 mov byte ptr [edx], al // des_mes[counter2]=al
83 0040142a add dword ptr [esp+78h], 1 // inc counter2
84
85 //block8-----
86 0040142f mov ebx, dword ptr [esp+78h] // ebx=counter2
87 00401433 lea eax, [esp+20h] // eax=des_mes
88 00401437 mov dword ptr [esp], eax // [esp]=des_mes
89 0040143a call Secret+0x1ca8 (00401ca8) // call msvcrt!strlen, eax=len(des_mes)
90
91 0040143f cmp ebx, eax // counter2 compare len(des_mes)
92 00401441 jb Secret+0x140b (0040140b) // if ebx<eax goto block7
93
94 //block9-----
95 00401443 lea eax, [esp+20h] // eax=des_mes
96 00401447 mov dword ptr [esp], eax // [esp]=des_mes
97 0040144a call Secret+0x1cb0 (00401cb0) // call msvcrt!puts
98 0040144f call Secret+0x1c60 (00401c60) // call msvcrt!_getch
99
100 //block10-----
101 00401454 mov eax, 0 // eax=0
102 00401459 jmp Secret+0x1460 (00401460) // goto block12
103
104 //block11-----
105 0040145b mov eax, 0 // eax=0
106
107 //block12-----
108 00401460 lea esp, [ebp-0Ch]
109 00401463 pop ebx
110 00401464 pop esi
111 00401465 pop edi
112 00401466 pop ebp
113 00401467 ret
114
115 //end main
```

להלן קוד בפייטון המדמה את הפעולה התוכנית:



The screenshot shows a Python IDE with a file named `Secret.py` open. The code in the file is as follows:

```
1 crypt_message = "S{p4c{f.4m{a4ufq4sq``}zs4}`5"
2
3 user_input = input("Magic word:\n")
4 if len(user_input) > 3:
5     sum_input = 0
6     for ch in user_input[1:9]:
7         sum_input += ord(ch)
8     print("sum ascii of user_input[1:9] is:", sum_input)
9     sum_input -= 0x34E
10    if sum_input == 0x14:
11        message = ''
12        for ch in crypt_message:
13            message += chr(ord(ch)^sum_input)
14    print(message)
```

Below the code editor, the output of the script is displayed:

```
Run: C:\Networks\Python3.8\python.exe C:/assembly/dynamicAnalysis/Secret/Secret.py
Magic word:
aLLLLLLn
sum ascii of user_input[1:9] is: 866
Good wor: you are getting it!
Process finished with exit code 0
```