

שלב ראשון למצוא את MAIN

bp \$xentry

g

נראה את modules list

lm

```
0:000> lm
start  end  module name
00400000 0040e000 Secret (no symbols)
75290000 7548e000 KERNELBASE (deferred)
768d0000 7698f000 msvcrt (deferred)
76d60000 76e40000 KERNEL32 (deferred)
77890000 77a2a000 ntdll (pdb symbols) C:\ProgramData\Dbg\sym\wntdll.pdb\3D038F
```

נבדוק איזה מחרוזות יש בתוכנית על ידי הפקודה:

s -as 400000 L?40e000

נחפש את המילה Mgick word שזה מודפס בconsole

```
00403000  _Jv_RegisterClasses
00403024  "Magic word: "
00403034  "S{{p4c{f"
0040303d  "4m{a4ufq4sq`` }zs4}`5"
00403056  "qugq84gqzp4m{af4g{xa``}{z:40P^%|U"
00403076  "P$I"
```

רואים שבכתובת 00403024 יש את המחרוזת

נבדוק איך נראה המחסנית, ונסה לגלות מי קורא למחרוזת,

נכניס את הפקודה: ba r1 403024 כדי שיהיה breakpoint שם ואחר כך פקודת g נקבל:

```
0:000> kb
# ChildEBP RetAddr  Args to Child
00 0060fe20 76944fcb 76984620 00403024 00000000 msvcrt!_output_l+0x109
01 0060fe68 00401391 00403024 006a0c9c 0060fe84 msvcrt!printf+0x5b
WARNING: Stack unwind information not available. Following frames may be wrong.
02 0060ff08 004010fd 00000002 006a23c0 006a18e8 Secret+0x1391
03 0060ff80 778f7c24 002ef000 1a1ba488 00000000 Secret+0x10fd
04 0060ffdc 778f7bf4 ffffffff 77918fdb 00000000 ntdll!_RtlUserThreadStart+0x2f
05 0060ffec 00000000 00401280 002ef000 00000000 ntdll!_RtlUserThreadStart+0x1b
```

רואים ש401391 מתבצע קריאה לפונקציה להדפיס את המחרוזת "Magick word"

נבדוק מה יש בכתובת הזו 00401391

Address: 00401391 ☒ Follow current instruction

```

004013b1 89cb      mov     esi, eax
00401363 f3a5      rep movs dword ptr es:[edi], dword ptr [esi]
00401365 89f0      mov     eax, esi
00401367 89fa      mov     edx, edi
00401369 0fb708    movzx   ecx, word ptr [eax]
0040136c 66890a    mov     word ptr [edx], cx
0040136f 83c202    add     edx, 2
00401372 83c002    add     eax, 2
00401375 c744247400000000 mov     dword ptr [esp+74h], 0
0040137d c744247000000000 mov     dword ptr [esp+70h], 0
00401385 c7042424304000 mov     dword ptr [esp], offset Secret+0x3024 (00403024)
0040138c e807090000 call    Secret+0x1c98 (00401c98)
00401391 a130614000 mov     eax, dword ptr [Secret+0x6130 (00406130)]
00401396 89442408    mov     dword ptr [esp+8], eax
0040139a c74424040a000000 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 e800080000 call    Secret+0x1ca8 (00401ca8)

```

רואים שעדיין זה לא main

נכתוב פקודת gu שיעלה במחשנית עוד, ונקבל

```

00 0060fe68 00401391 00403024 006a0c9c 0060fe84 msvcrt!printf+0x5b
WARNING: Stack unwind information not available. Following frames may be wrong.
01 0060ff08 004010fd 00000002 006a23c0 006a18e8 Secret+0x1391
02 0060ff80 778f7c24 002ef000 1a1ba488 00000000 Secret+0x10fd
03 0060ffdc 778f7bf4 ffffffff 77918fdb 00000000 ntdll!_RtlUserThreadStart+0x2f
04 0060ffec 00000000 00401280 002ef000 00000000 ntdll!_RtlUserThreadStart+0x1b
0:000> g

```

DEBUG behuggee is running...

רואים שכבר הגיע להדפסה, נכניס את הכתובת 004010fd ונקבל:

```
Disassembly
Address: 4010fd
[ ] Follow current instruction
004010cc 8910      mov     dword ptr [eax], eax
004010cf e8ec060000 call    Secret+0x17c0 (004017c0)
004010d4 83e4f0    and     esp, 0FFFFFF0h
004010d7 e844090000 call    Secret+0x1a20 (00401a20)
004010dc e89f0b0000 call    Secret+0x1c80 (00401c80)
004010e1 8b00      mov     eax, dword ptr [eax]
004010e3 89442408  mov     dword ptr [esp+8], eax
004010e7 a100504000 mov     eax, dword ptr [Secret+0x5000 (00405000)]
004010ec 89442404  mov     dword ptr [esp+4], eax
004010f0 a104504000 mov     eax, dword ptr [Secret+0x5004 (00405004)]
004010f5 890424    mov     dword ptr [esp], eax
004010f8 e843020000 call    Secret+0x1340 (00401340)
004010fd 89c3      mov     ebx, eax
004010ff e8840b0000 call    Secret+0x1c88 (00401c88)
00401104 8b1c24    mov     ebx, dword ptr [esp+4], ebx
```

מכאן ניתן להבין שהכתובת 0401340 הינו כתובת הmain

שלב שני,

אחרי שאנו יודעים שכתובת הmain הינו 401340

ננסה להבין מה התוכנית עושה, ואיך נוכל לגרום לה להדפיס את המחרוזת הנכונה.

נכניס את הכתובות 401340 ונקבל:

Disassembly

Address: 401340

☒ Follow current instruction

0040133f	90	nop	
00401340	55	main push	ebp
00401341	89e5	mov	ebp, esp
00401343	57	push	edi
00401344	56	push	esi
00401345	53	push	ebx
00401346	83e4f0	and	esp, 0FFFFFF0h
00401349	83c480	add	esp, 0FFFFFF80h
0040134c	e8cf060000	call	Secret+0x1a20 (00401a20)
00401351	8d542420	lea	edx, [esp+20h]
00401355	b834304000	mov	eax, offset Secret+0x3034 (00403034)
0040135a	b911000000	mov	ecx, 11h
0040135f	89d7	mov	edi, edx
00401361	89c6	mov	esi, eax
00401363	f3a5	rep movs	dword ptr es:[edi], dword ptr [esi]
00401365	89f0	mov	eax, esi
00401367	89fa	mov	edx, edi
00401369	0fb708	movzx	ecx, word ptr [eax]
0040136c	66890a	mov	word ptr [edx], cx
0040136f	83c202	add	edx, 2
00401372	83c002	add	eax, 2
00401375	c744247400000000	mov	dword ptr [esp+74h], 0
0040137d	c744247000000000	mov	dword ptr [esp+70h], 0
00401385	c704247000000000	mov	dword ptr [esp], offset Secret+0x3024 (00403024)
0040138c	e80709	printf call	Secret+0x1c98 (00401c98)
00401391	a130614000	mov	eax, dword ptr [Secret+0x6130 (00406130)]
00401396	89442408	mov	dword ptr [esp+8], eax
0040139a	c74424040a000000	mov	dword ptr [esp+4], 0Ah
004013a2	8d442466	lea	eax, [esp+66h]
004013a6	890424	mov	dword ptr [esp], eax
004013a9	e8f209	fgets call	Secret+0x1ca0 (00401ca0)
004013ae	8d442466	lea	eax, [esp+66h]
004013b2	890424	mov	dword ptr [esp], eax
004013b5	e8ee09	strlen call	Secret+0x1ca8 (00401ca8)

main

תחילת הmain

מחרוזת (מוזרה)

Printf("Magick word")

fgets(char *str, int n, FILE *stream)

fgets(add=[esp+66h], n=10, file=input)

Strlen(the input from the user)

0:000> da 00403034

00403034 "S{{p4c{f.4m{a4ufq4sq``}zs4}`5"

לסיכום עד כאן, רואים שמדפיס בהתחלה למסך "magick word"

והקלט שהמשתמש מכניס נקלא על ידי פונקציה fgets, ואותו קלט נשלח לפונקציה שמחזירה את כמות התווים strlen

בודקים האם ס

```

004013ba 83f803      cmp     eax, 3
004013bd 0f8698000000 jbe     Secret+0x145b (0040145b)
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
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)
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
0040142f 8b5c2478     mov     ebx, dword ptr [esp+78h]
00401433 8d442420     lea     eax, [esp+20h]
00401437 890424      mov     dword ptr [esp], eax

0040143a e8690800    strlen   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 printf   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

```

פונקציה: מאתחלת מונה ל1, עוברת על כל התווים החל מהתו השני שהשתמש הכניס, וסוכמת את הערך האסקי שלהם, ומחזירה את הערך של הסכום של כל התווים

```

sumOfCharacters(char* input){
    sum = 0; //the sum of the hex
character
    for(int i = 1; i < 10; i++){
        sum += input[i];
    }
    Return sum
}

```

בודקים אם הערך של כל התווים (מתון השני עד העשירי) פחות מספר (34E) שווה ל14h אז נכנסים לפונקציה שלוקחת את המחרוזת שראינו בהתחלה (המוזרה) ועושה עליו xor עם 14h על כל תוו. ומדפיס למסך את המחרוזת המתוקן.

```

Char* str[] =
    "S{{p4c{f.4m{a4ufq4sq``}zs4}5";
If(sum - 34E == 14h){
    Message = "";
    For(int i = 0; i < str.length; i++){
        Strcat(Message, (str[i] XOR 14h)
    );
    }}
Printf("%s", message);

```

לסיכום ראינו שאם הקלט שהשתמש מכניס, התוכנית בודקת האם הסכום של התווים (מאינדקס 1 עד 10) פחות 34E יצא 14h

אם כן, אז התוכנית לוקחת את המחרוזת S{{p4c{f.4m{a4ufq4sq``}zs4}5 ועושה עליו XOR עם מספר 14h והתוצאה היא מדפיסה על המסך.

ולכן כדי להכניס מחרוזת נכונה צריך לדאוג שסכום האסקי שלהם יהיה 34E + 14

כלומר חוץ מהתוו הראשון ב9 תווים הבאים ערך האסקי שלהם יהיה תואם לה362

	0	1	3	4	5	6	7	8	9	ס"ה
	A	h	a	r	o	n	{	R	}	
hex	41	68	61	72	6f	6e	20			
int	104		97	114	111	110	125	82	123	866

Command Prompt

Magic word: Aharon{R}
Good work you are getting it!