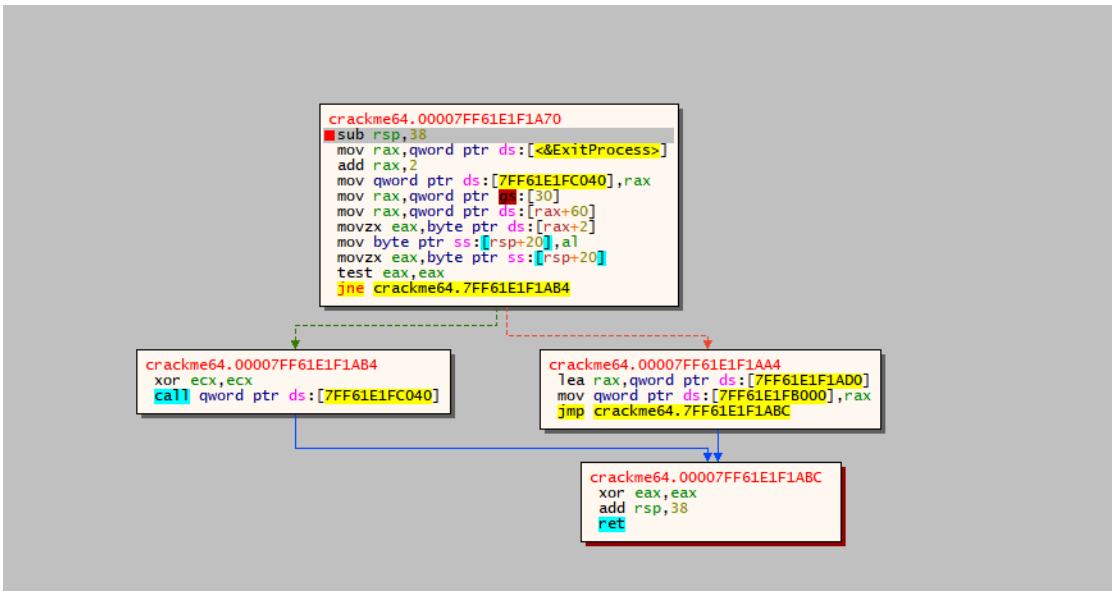
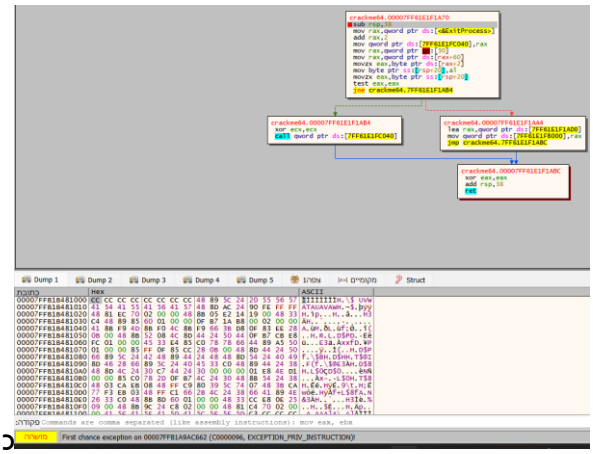
מצאתי מנגנון אחד של אנטי דיבאג:

אחרי שהרצתי את התוכנית עם האנטי דיבאג כדי לנסות לראות איפה אני קורס ראיתי שאני מגיע לחלק הזה:



כמובן שהמשכתי להריץ וראיתי שאחרי כמה הרצות אני מגיע לצד שמאל וזה מקריס אותי כמו כאן:



כמובן שאני לא רוצה לקרוס ולכן אפשרות אחת לנטרל את מנגנון האנטי דיבאג תהיה לבטל את הקריאה שיש שם לפונקציה או לבטל את הבדיקה וישר ללכת לצד ימין או כל אפשרות אחרת...

בכל מקרה הקוד כאן מעניין מכמה סיבות:

```
sub rsp,38
mov rax,qword ptr ds:[<&ExitProcess>]
add rax,2
mov qword ptr ds:[7FF61E1FC040],rax
```

דבר ראשון אנחנו לוקחים את הכתובת של היציאה מתהליך ושמים את זה בזיכרון במקום מסוים שנשתמש בהמשך.

```
crackme64.00007FF61E1F1AB4
xor ecx,ecx
call qword ptr ds:[7FF61E1FC040]
```

זה התמונה של צד ימין שאנחנו יוצאים, למעשה זה המקום שאליו אנחנו עלולים להגיע וזה די ברור שזה המקום הרע כי זה בדיוק איפה ששמרנו את הכתובת של יציאה מהתהליך.

עוד דבר מעניין שיש פה זה:

```
mov rax,qword ptr gs:[30]
mov rax,qword ptr ds:[rax+60]
movzx eax,byte ptr ds:[rax+2]
mov byte ptr ss:[rsp+20],al
movzx eax,byte ptr ss:[rsp+20]
test eax,eax
jne crackme64.7FF61E1F1AB4
```

יש לנו כאן גם קוד ארוך יחסית שבודק את הבייט המסויים של EAXואנחנו מריצים הרבה כתובות ומצביעים להגיע לשם.

למעשה אחרי שהרצתי את הקוד והסתכלתי על הערך של הרגיסטר ראיתי ש:

```
הסתר FPU
RAX   0000000000000001
RBX   00007FF61E1F8248   crackm
RCX   00007FF61E1F1A70   crackm
RDX   0000000098844E55
```

כאן למעשה אפשר לראות ש EAXשווה 1 שזה מחשיד כי הביט שאומר אם אנחנו בדיבא הוא גם 1.

למעשה כל הגישה הארוכה הזאת לזיכרון כנראה אומרת לתוכנית אם אנחנו מדובגים, למעשה הפתרון שלי לפצ'פץ היא תהיה פשוט לדלג על הבדיקה ולקפוץ לצד הטוב: למשל:

```
mov rax,qword ptr ds:[rax+60]
movzx eax,byte ptr ds:[rax+2]
mov byte ptr ss:[rsp+20],al
movzx eax,byte ptr ss:[rsp+20]
nop
nop
nop
nop
lea rax,qword ptr ds:[7FF61E1F1AD0]
mov qword ptr ds:[7FF61E1FB000],rax
jmp crackme64.7FF61E1F1ABC
```

וככה ייראה הקוד לאחר הפצ'פוץ'. למעשה אני הרווחתי שהקוד של האזור הטוב מיד אחרי התנאי אז במקום לשנות אותו לקפיצה לא מותנה שיניתי את זה שהוא לא יקפוץ בכלל ופשוט ימשיך לצד הטוב...

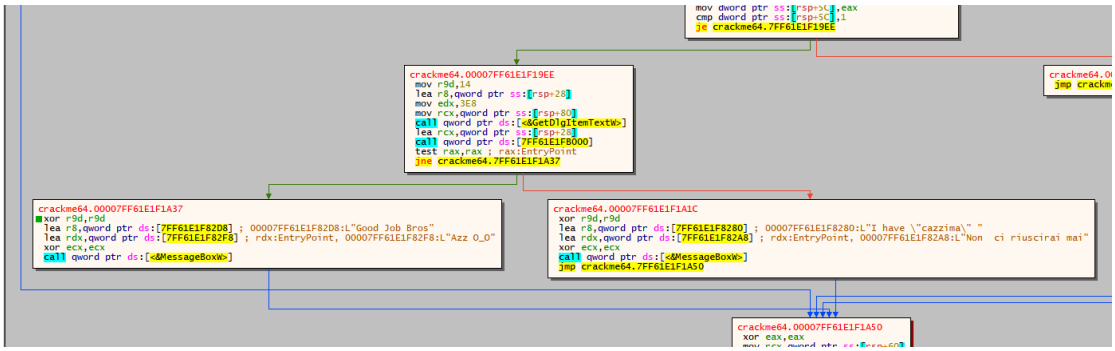
עכשיו אחרי שעשיתי את הפצ'פוץ' נשאר רק לעשות את הבדיקה של הגעה למסר הטוב:

כדי להגיע למסר הטוב אני מתחיל עם בדיקה של המחרוזות שיש בקובץ בתקווה כמובן שהן לא מוצפנות, ואכן אני יכול לזהות כמה מחרוזות טובות:

```
כתובת             disassembly                                          מחרוזת
00007FF61E1F1A1F   lea r8,qword ptr ds:[7FF61E1F8280]                   L"I have \"cazzima\" "
00007FF61E1F1A26   lea rdx,qword ptr ds:[7FF61E1F82A8]                 L"Non  ci riuscirai mai"
00007FF61E1F1A3A   lea r8,qword ptr ds:[7FF61E1F82D8]                  L"Good Job Bros"
00007FF61E1F1A41   lea rdx,qword ptr ds:[7FF61E1F82F8]                 L"Azz O_O"
00007FF61E1F1FFC   lea rcx,qword ptr ds:[7FF61E1F8328]                 L"mscoree.dll"
00007FF61E1F200E   lea rdx,qword ptr ds:[7FF61E1F8318]                 "CorExitProcess"
00007FF61E1F242E   lea r8,qword ptr ds:[7FF61E1F8910]                  "Runtime Error!\n\nProgram: "
00007FF61E1F2479   lea r8,qword ptr ds:[7FF61E1F88F8]                  "<program name unknown>"
00007FF61E1F24D2   lea r8,qword ptr ds:[7FF61E1F88F4]                  "..."
00007FF61E1F2505   lea r8,qword ptr ds:[7FF61E1F88F0]                  "\n\n"
00007FF61E1F255F   lea rdx,qword ptr ds:[7FF61E1F88C8]                 "Microsoft Visual C++ Runtime Library
00007FF61E1F4086   lea rcx,qword ptr ds:[7FF61E1F8A48]                 "USER32.DLL"
00007FF61E1F409F   lea rdx,qword ptr ds:[7FF61E1F8A38]                 "MessageBoxA"
00007FF61E1F40C0   lea rdx,qword ptr ds:[7FF61E1F8A28]                 "GetActiveWindow"
00007FF61E1F40DF   lea rdx,qword ptr ds:[7FF61E1F8A10]                 "GetLastActivePopup"
00007FF61E1F40FE   lea rdx,qword ptr ds:[7FF61E1F89F0]                 "GetUserObjectInformationA"
00007FF61E1F412C   lea rdx,qword ptr ds:[7FF61E1F89D8]                 "GetProcessWindowStation"
00007FF61E1F4C15   lea rax,qword ptr ds:[7FF61E1FBD00]                 &"Sun"
```

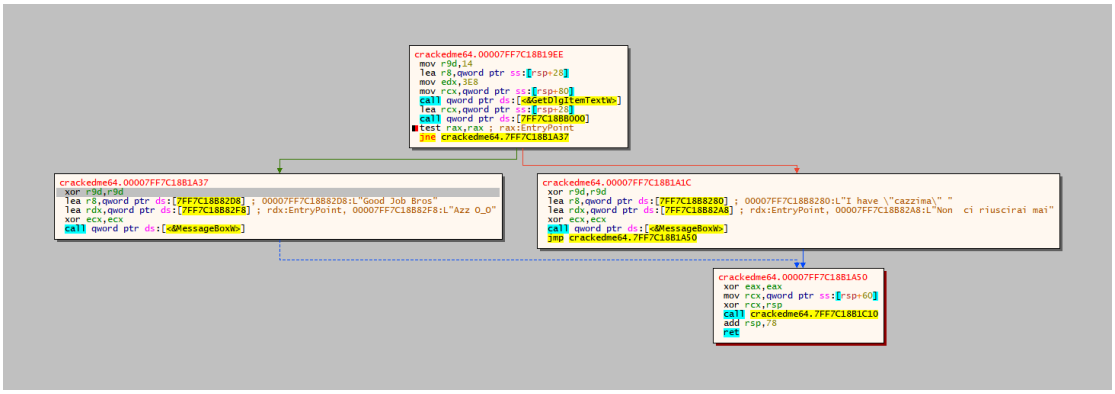
כמובן שאני אתעניין באזור של ההודעה אם זה הצלחה או לא ובהודעה על כישלון(הרצתי את זה ואני יודע שהיא הטקסט המוזר למעלה, אבל בכל מקרה זה הטקסט היחיד פה)

אחרי שהלכתי למיקום שקיבלתי את הטקסט ובניתי את העץ המלא יצא שיש אזור שאני מעונין בו במיוחד:



למעשה די ברור שזה החלק שמעניין אותי להגיע לאזור של ההצלחה.

אני מעריך שבפונקציה לפני זה בודק את הסיסמה, אבל זה ממש לא מעניין אותי מה היא, אלא אני אפצ'פץ' את EIP כדי לקפוץ להדפסה של הסיסמה:



בשביל להגיע לאזור הטוב צריך שתודפס התיבה הראשונה קודם אז אני אעצור לפני הבדיקה אם זה סיסמה טובה או לא ואז אני אקפוץ למקום של ההדפסה של הצלחה:

והנה המסר שהצלחתי! יאיי!

```
00007FF7C18B1A37   45:33C9                xor r9d,r9d
00007FF7C18B1A3A   4C:8D05 97680000       lea r8,qword ptr ds:[7FF7C18B82D8]      00007FF7C18B82D8:L"Good Job Bros"
00007FF7C18B1A41   48:8D15 B0680000       lea rdx,qword ptr ds:[7FF7C18B82F8]     00007FF7C18B82F8:L"Azz O_O"
00007FF7C18B1A48   33C9                   xor ecx,ecx
00007FF7C18B1A4A   FF15 A8670000          call qword ptr ds:[<&MessageBoxW>]
00007FF7C18B1A50   33C0                   xor eax,eax
00007FF7C18B1A52   48:8B4C24 60           mov rcx,qword ptr ss:[rsp+60]
00007FF7C18B1A57   48:33CC                xor rcx,rsp
00007FF7C18B1A5A   E8 B1010000            call crackedme64.7FF7C18B1C10
00007FF7C18B1A5F   48:83C4 78             add rsp,78
00007FF7C18B1A63   C3                     ret
00007FF7C18B1A64   CC                     int3
```

הקובץ אחרי הקפיצה.

ד"א כמובן שיש אינספורר דרכים לשנות את הקוד כך שיודפס הודעת הצלחה אבל הרבה זמן לא היה פתרון שכלל קפיצה עם EIPאז אני מנצל את זה בשביל זה.... כמובן שיש אפשרות לבטל את התנאי לבטל את הבדיקה של הסיסמה, לשנות את הבדיקה שתמיד תחזיר כאילו זה סיסמה טובה וכו'.

השלמות לשאלות שיש ותשובות מרוכזות:

מנגנוני האנטי דיבאג: היה בדיקה של הביט שאומר אם מדבגים אותנו, מחקתי את ההשוואה שהם עשו

איך מצאתי את המיין:

יש אפשרות עם IDAבדכ למצוא את זה בצורה פשוטה:

על פי IDA המיין נמצא כאן F000000001400018, איך שפתחתי את התוכנה זה ישר מביא את המיין:

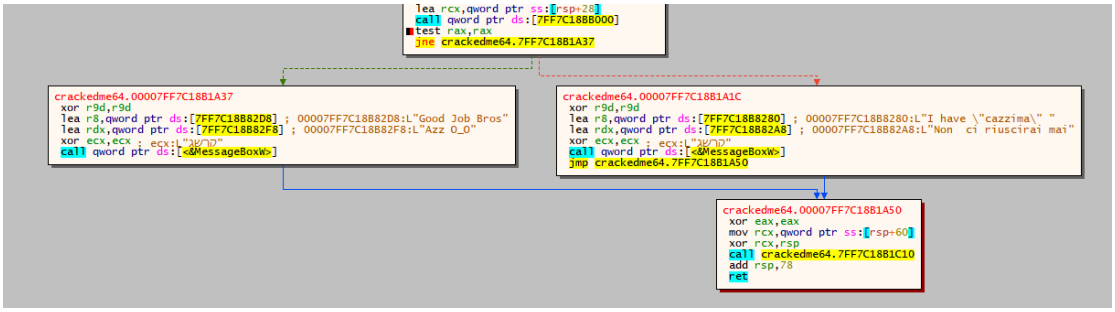
```
.text:00000001400018F0 ; int __stdcall wWinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPWSTR lpCmdLine, int nShowCmd)
.text:00000001400018F0 wWinMain        proc near               ; CODE XREF: __tmainCRTStartup+183↓p
.text:00000001400018F0
.text:00000001400018F0 dwInitParam     = qword ptr -18h
.text:00000001400018F0 hInstance       = qword ptr  8
.text:00000001400018F0 arg_8           = qword ptr  10h
.text:00000001400018F0 arg_10          = qword ptr  18h
.text:00000001400018F0 arg_18          = dword ptr  20h
.text:00000001400018F0
.text:00000001400018F0                 mov     [rsp+arg_18], r9d
.text:00000001400018F5                 mov     [rsp+arg_10], r8
.text:00000001400018FA                 mov     [rsp+arg_8], rdx
.text:00000001400018FF                 mov     [rsp+hInstance], rcx
.text:0000000140001904                 sub     rsp, 38h
.text:0000000140001908                 mov     [rsp+38h+dwInitParam], 0 ; dwInitParam
.text:0000000140001911                 lea     r9, DialogFunc  ; lpDialogFunc
.text:0000000140001918                 xor     r8d, r8d        ; hWndParent
```

האם רץ קוד לפני המיין: כן, אחרי הכל יש את הבדיקה של האנטי דיבאג לפני.

איפה בודקים את הסיסמה ואיך עקפתי:

```
00007FF7C18B1ACF   CC                 int3
00007FF7C18B1AD0   48:894C24 08       mov qword ptr ss:[rsp+8],rcx
00007FF7C18B1AD5   48:83EC 18         sub rsp,18
00007FF7C18B1AD9   C70424 00000000    mov dword ptr ss:[rsp],0
00007FF7C18B1AE0   48:8B4424 20       mov rax,qword ptr ss:[rsp+20]
00007FF7C18B1AE5   0FB700             movzx eax,word ptr ds:[rax]
00007FF7C18B1AE8   83F8 4D            cmp eax,4D                          4D:'M'
00007FF7C18B1AEB   75 68              jne crackedme64.7FF7C18B1B55
00007FF7C18B1AED   48:8B4424 20       mov rax,qword ptr ss:[rsp+20]
00007FF7C18B1AF2   0FB740 02          movzx eax,word ptr ds:[rax+2]
00007FF7C18B1AF6   83F8 34            cmp eax,34                          34:'4'
00007FF7C18B1AF9   75 5A              jne crackedme64.7FF7C18B1B55
00007FF7C18B1AFB   48:8B4424 20       mov rax,qword ptr ss:[rsp+20]
00007FF7C18B1B00   0FB740 04          movzx eax,word ptr ds:[rax+4]
00007FF7C18B1B04   83F8 58            cmp eax,58                          58:'X'
00007FF7C18B1B07   75 4C              jne crackedme64.7FF7C18B1B55
00007FF7C18B1B09   48:8B4424 20       mov rax,qword ptr ss:[rsp+20]
00007FF7C18B1B0E   0FB740 06          movzx eax,word ptr ds:[rax+6]
00007FF7C18B1B12   83F8 50            cmp eax,50                          50:'P'
00007FF7C18B1B15   75 3E              jne crackedme64.7FF7C18B1B55
00007FF7C18B1B17   48:8B4424 20       mov rax,qword ptr ss:[rsp+20]
00007FF7C18B1B1C   0FB740 08          movzx eax,word ptr ds:[rax+8]
00007FF7C18B1B20   83F8 34            cmp eax,34                          34:'4'
00007FF7C18B1B23   75 30              jne crackedme64.7FF7C18B1B55
00007FF7C18B1B25   48:8B4424 20       mov rax,qword ptr ss:[rsp+20]
00007FF7C18B1B2A   0FB740 0A          movzx eax,word ptr ds:[rax+A]
00007FF7C18B1B2E   83F8 31            cmp eax,31                          31:'1'
00007FF7C18B1B31   75 22              jne crackedme64.7FF7C18B1B55
00007FF7C18B1B33   48:8B4424 20       mov rax,qword ptr ss:[rsp+20]
00007FF7C18B1B38   0FB740 0C          movzx eax,word ptr ds:[rax+C]
00007FF7C18B1B3C   83F8 4E            cmp eax,4E                          4E:'N'
00007FF7C18B1B3F   75 14              jne crackedme64.7FF7C18B1B55
00007FF7C18B1B41   48:8B4424 20       mov rax,qword ptr ss:[rsp+20]
00007FF7C18B1B46   0FB740 0E          movzx eax,word ptr ds:[rax+E]
00007FF7C18B1B4A   85C0               test eax,eax
00007FF7C18B1B4C   75 07              jne crackedme64.7FF7C18B1B55
00007FF7C18B1B4E   C70424 01000000    mov dword ptr ss:[rsp],1
00007FF7C18B1B55   48:630424          movsxd rax,dword ptr ss:[rsp]
00007FF7C18B1B59   48:83C4 18         add rsp,18
00007FF7C18B1B5D   C3                 ret
00007FF7C18B1B5E   CC                 int3
```

זה הבדיקה של הסיסמה כמו שאפשר לראות שבודקים רגע לפני ההכרעה אם זה סיסמה טובה או לא:



השורה הראשונה שמודגשת בכחול זה הקריאה לבדיקה ואחריה בדיקה של הערך המוחזר.

איך עקפתי: פשוט שיניתי את EIP שיצביע על האזור הטוב, אפשרויות אחרות היו להעלים את הבדיקה וישר לקפוץ בלי תנאי לאזור הטוב.