מבוא

אהלן, אני עמית כהן בן 24 סטודנט שנה ב' באוניברסיטת בר אילן למדעי המחשב.

תקופת המבחנים נגמרה, ופתאום גיליתי שיש לי הרבה זמן פנוי אז החלטתי לאתגר את עצמי ולפצח קוד בעזרת $reverse\ engineering$.

$reverse\ engineering$ - תהליך שמטרתו לחלץ מתוכנת מחשב את האלגוריתם שבבסיסה כאשר זמין למהנדס קובץ ההרצה בלבד (exe). – ויקיפדיה.

חיפשתי אתגר בגוגל ומצאתי את האתר *http://reversing.kr/index.php*, והחלטתי לפתור את אחד האתגרים שם $easy\ crack$, קובץ הרצה על $windows$.

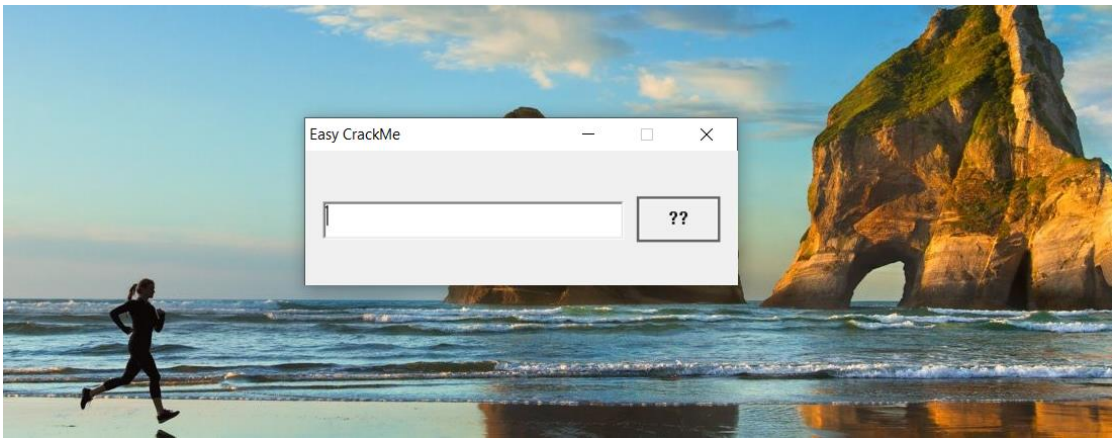
במסמך זה אפרט את יומן המסע שלי בדרך לפיצוח הקובץ.

במהלך הפתרון שלי, השתמשתי בתוכנת IDA ($interactive\ disassembler$) תוכנה שלמעשה לוקחת קובץ exe ועושה לו $disassembly$.

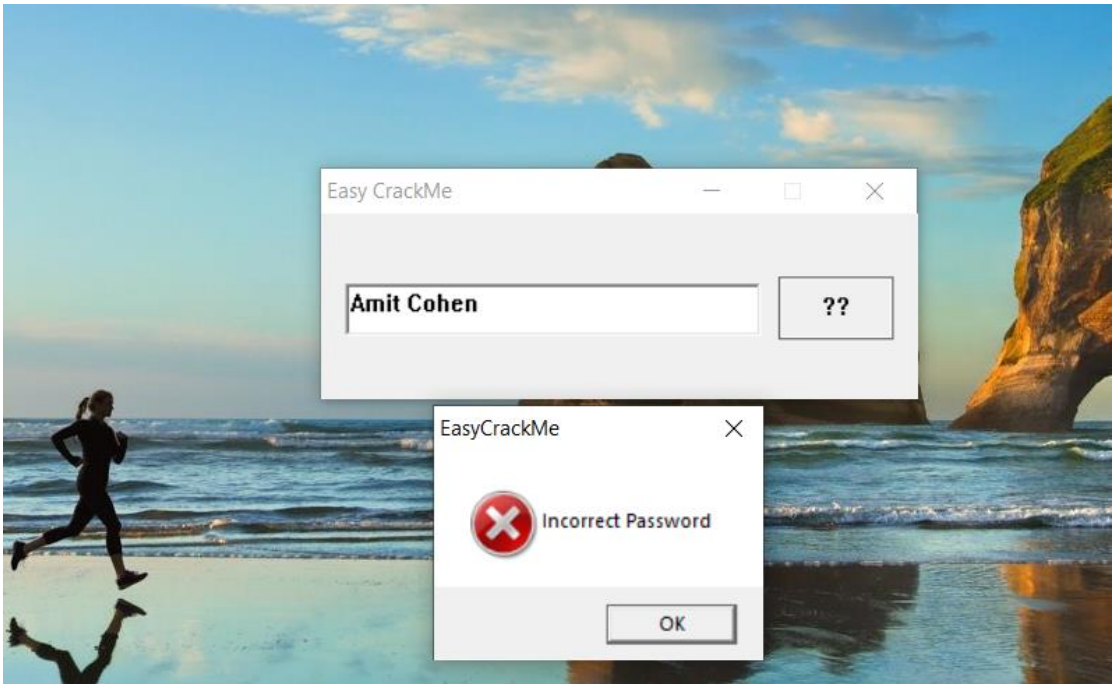
תוכנת הIDA שהשתמשתי בה היא $IDA\ Freeware$, תוכנה חינמית שניתן להוריד אותה מהכתובת הבאה: *https://hex-rays.com/ida-free/*.

יומן מסע

ראשית, הורדתי את הקובץ $Easy\_crackme.exe$ מהאתר, הרצתי את הקובץ ונפתח החלון הבא:



נפתח חלון דו שיח ($dialog\ box$), שמבקש להכניס משהו שנראה כמו סיסמא.
כמובן שבשלב זה אין לי מושג או כיוון למה הסיסמא תהיה, ננסה להכניס סתם מחרוזת ונראה אולי נוכל להסיק עוד מידע:



הכנסנו סיסמא לא נכונה ונפתח חלון חדש שאומר שהסיסמא אינה נכונה, ואין שום דבר מעבר לכך בקובץ ההרצה.

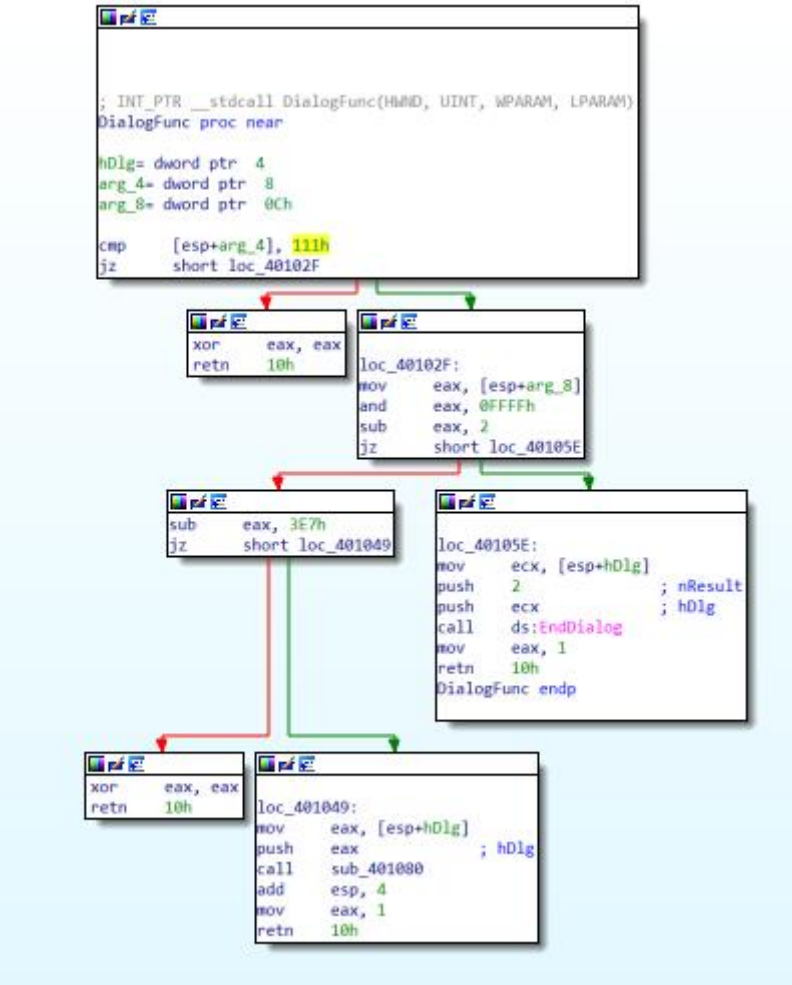
נעשה לקובץ ההרצה $disassembly$ בעזרת הIDA:

```
sub_401000 proc near

hInstance= dword ptr  4

mov     eax, [esp+hInstance]
push    0                       ; dwInitParam
push    offset DialogFunc ; lpDialogFunc
push    0                       ; hWndParent
push    65h ; 'e'               ; lpTemplateName
push    eax                     ; hInstance
call    ds:DialogBoxParamA
xor     eax, eax
retn    10h
sub_401000 endp
```

אנחנו רואים שהקובץ מתחיל לרוץ בפונקציה sub_401000, ושבשורה הרביעית לפני האחרונה הוא קורה לפונקציה $DialogBoxParamA$, שבחיפוש קצר בגוגל נגיע לדף הדקומנטציה של $microsoft$ לגבי הפונקציה הזו - https://docs.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-dialogboxparama

לפי דף התיעוד של הפונקציה, ניתן להבין כי זאת הפונקציה שאחראית לפתיחת תיבת הדו שיח.

ברור כי באיזה שהוא שלב הפונקציה צריכה לבדוק האם הסיסמא נכונה, אבל הפונקציה $DialogBoxParamA$ היא פונקציה מובנית של מערכת ההפעלה ומובן שלא היא זו שתבדוק האם הסיסמא נכונה, לכן ניתן להסיק כי הארגומנט הרביעי $DialogFunc$ הוא הפונקציה שבודקת האם הסיסמא נכונה! נצלול לתוך פונקציה זו וננסה לגלות רמזים לגבי הסיסמא הנכונה:



זה האסמבלי של הפונקציה $DialogFunc$ הפונקציה האחראית לבדיקת תקינות הסיסמא כאמור.
ניתן להשים לב כי הענף loc_40105E, הוא הענף המסיים את הדיאלוג כי הוא קורא לפונקציה $enddialog$, ולא הופיע עד סיום הדיאולוג איזה שהיא השוואה של סיסמאות.
לכן נסיק כי הענף השני הוא הענף האחראי לבדיקת הסיסמא, ובסופה היא קוראת לפונקציה

sub_401080 כאשר היא מעביר לו את הארגומנט $hDlg$ על המחסנית, נמשיך להסתכל על דף הדקומנטציה של $microsoft$ על הפרמטר הזה:

Parameters

`[in] hDlg`

Type: **HWND**

A handle to the dialog box that contains the control.

לכן פרמטר זה הוא הפרמטר ששולט על תיבת הדו שיח ולכן מאוד הגיוני שהפונקציה sub_401080 שיש לה את $hDlg$ תבדוק את תקינות הסיסמא (כי היא יכולה למשל לשלוף את הסיסמא מתוך תיבת הדו שיח בעזרת הפרמטר).

נצלול לפונקציה sub_401080:

```
sub_401080 proc near

String= byte ptr -64h
var_63= byte ptr -63h
var_62= byte ptr -62h
var_60= byte ptr -60h
hDlg= dword ptr  4

sub     esp, 64h
push    edi
mov     ecx, 18h
xor     eax, eax
lea     edi, [esp+68h+var_63]
mov     [esp+68h+String], 0
push    64h ; 'd'        ; cchMax
rep stosd
stosw
stosb
mov     edi, [esp+6Ch+hDlg]
lea     eax, [esp+6Ch+String]
push    eax              ; lpString
push    3E8h             ; nIDDlgItem
push    edi              ; hDlg
call    ds:GetDlgItemTextA
cmp     [esp+68h+var_63], 61h ; 'a'
jnz     short loc_401135
```

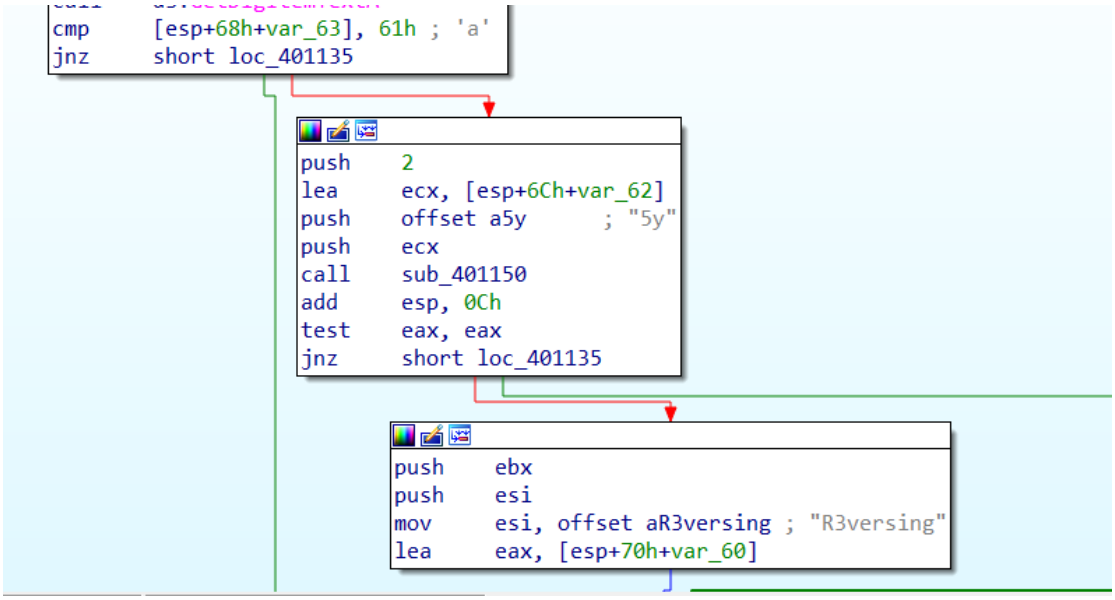
אנחנו רואים קריאה לפונקציה $GetDlgItemTextA$, שהיא הפונקציה שאכן שולפת את הסיסמא שהקלדנו בתוך תיבת הדו שיח, אנחנו בכיוון הנכון!

נראה כי שלושת הפרמטרים שהפונקציה $GetDlgItemTextA$ מקבלת הם $hDlg, 0x3E8, String$, ונסיק שלאחר הקריאה לפונקציה זו הסיסמא שהקלדנו בתיבת דו השיח נשמרת במשתנה $String$.

אם נסתכל על המשתנים שהוגדרו, כלומר על $String, var\_63, var\_62, var\_60$, ונסתכל על ה$offset$ שלהם נוכל להסיק כי הם נשמרו בצמוד אחד לשני, כלומר var_63 הוא בית אחד מתחת ל$String$, var_62 2 בתים מתחת ל$String$, או בצורה יותר אינטואיטיבית:

$$var\_63 = String[1], var\_62 = String[2], var\_60 = String[4]$$

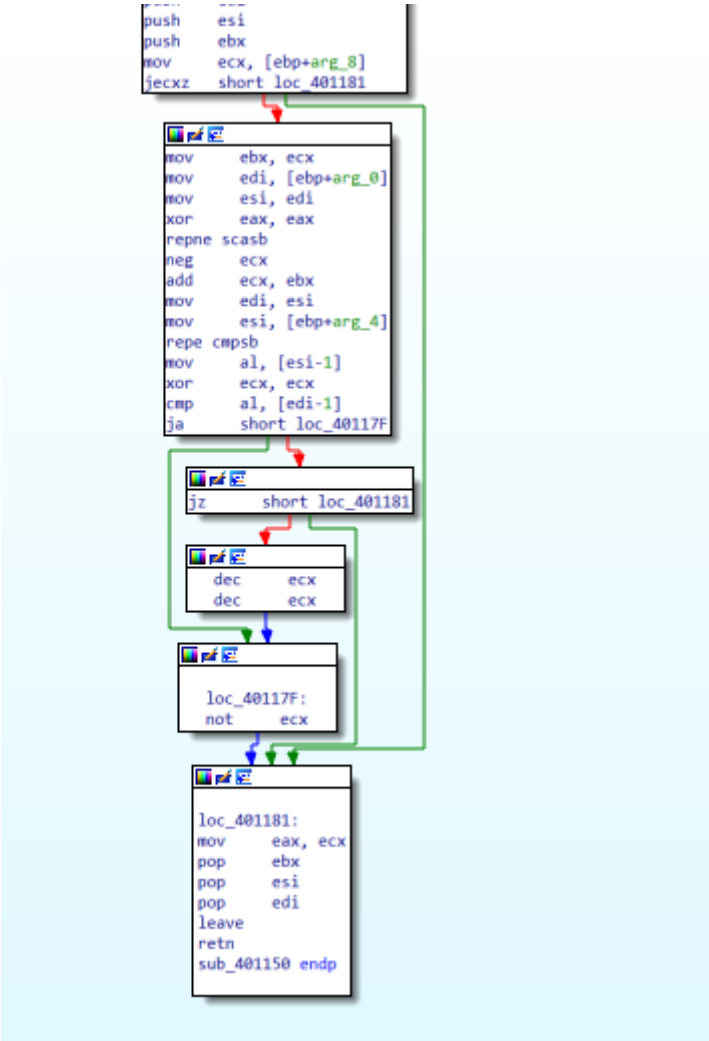
כעת בשתי השורות האחרונות, יש השוואה בין var_63 ל $0x61 = 'a'$ לפי קידוד $'ASCII'$, ולאחר מכן הקפיצה jnz. נסתכל לאיפה הקפיצה תוביל אותנו:



כפי שניתן לראות, אם var_63 או הבית הראשון של הסיסמא הוא לא $'a'$ אז התכנית תקפוץ לloc_401135, ששם יודפס "$wrong\ password$", לכן נסיק כי הענף loc_401135, הוא הענף שבו הסיסמא לא נכונה, ולכן הבית הראשון של הסיסמא הוא $String[1] = 'a'$.

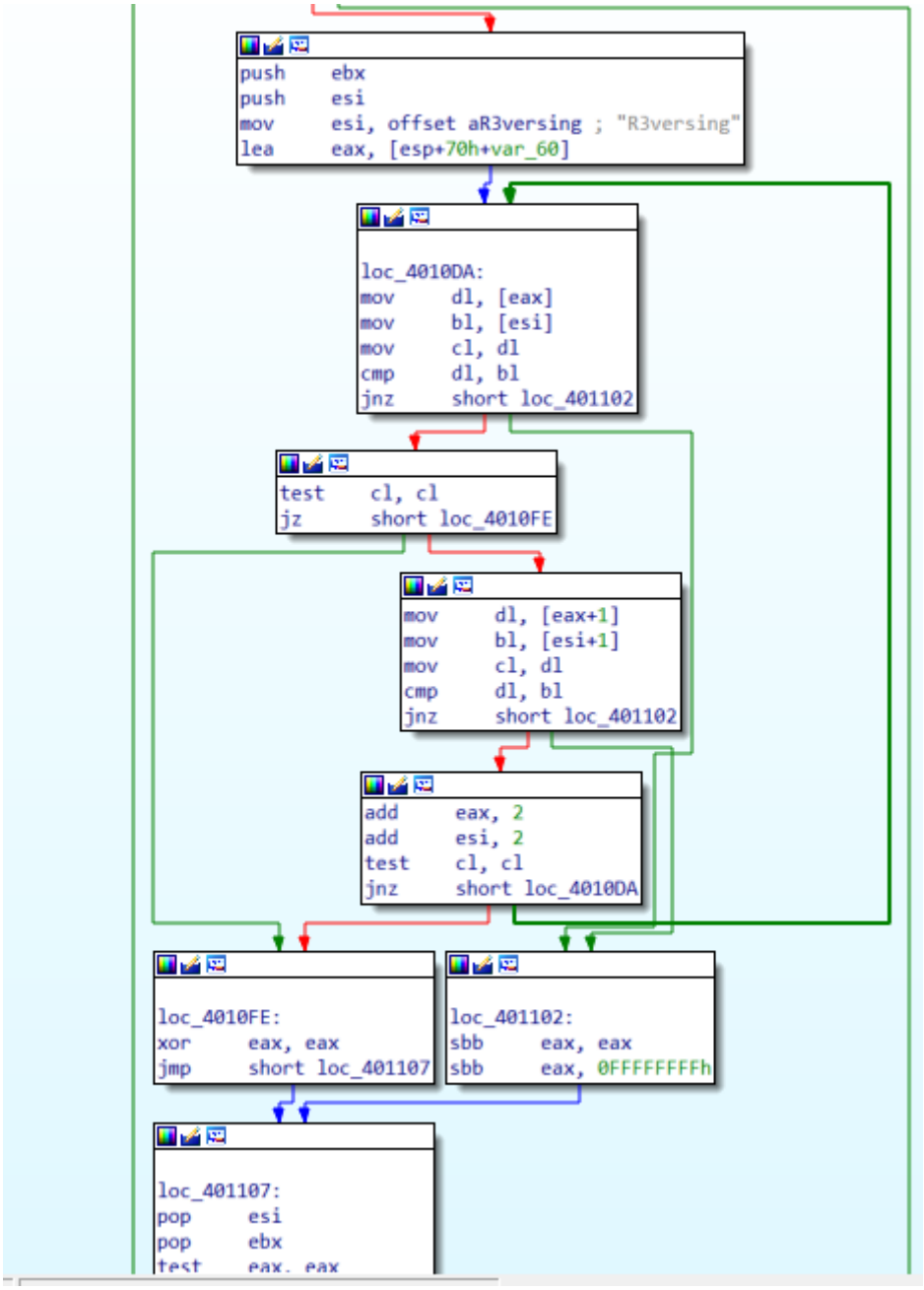
נמשיך לתיבה הימנית, אנחנו רואים שקוראים לפונקציה sub_401150, כאשר מעבירים לה כפרמטר את המחרוזת "$5y$", אם הפונקציה sub_401150, מחזירה 0 אז אנחנו נמשיך בבדיקה, אחרת נקפוץ לloc_401135 שכאמור הוא הענף שנקפוץ אליו כאשר הסיסמא תהיה לא נכונה.

נצלול לפונקציה sub_401150:

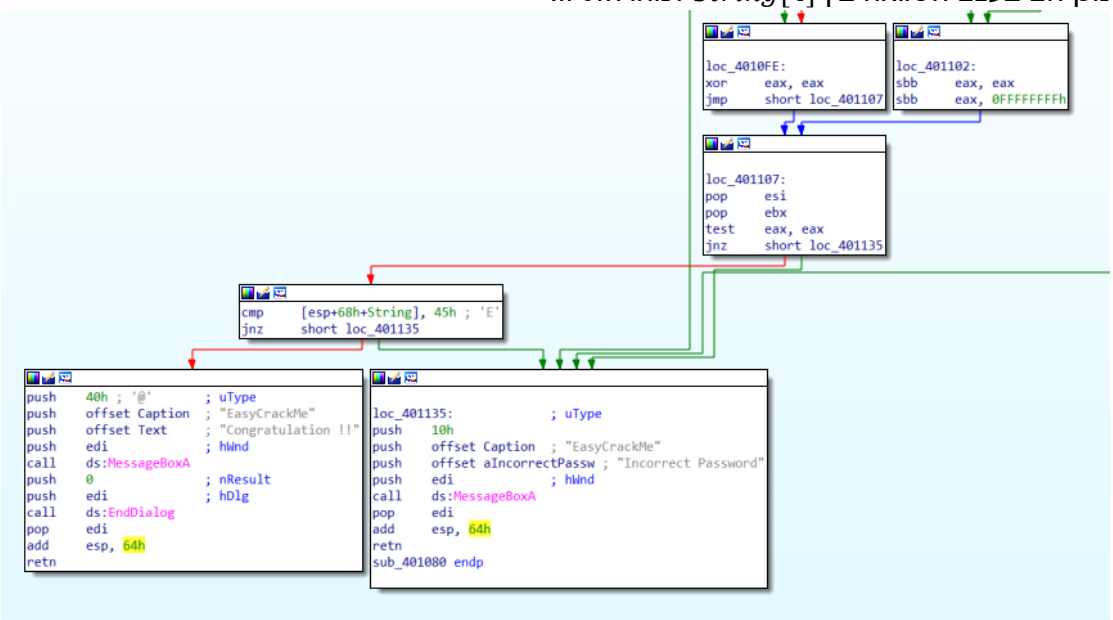


קודם כל מעבירים את $String[2]$ לedi, ואז את המחרוזת "$5y$" לesi, והמעבד מבצע את הפקודה $repe\ cmpsb$, שבלי להתעכב יותר מדי על הפקודה הזאת, מה שיקרה זה שלאחר הפקודה edi יצביע על התו הראשון ב$String[2]$ ששונה מ"$5y$", ולאחר מכן בודקים האם edi ו esi זהים, אם כן הפונקציה תחזיר 0 (בדיוק כמו שרצינו ע"מ שלא נקפוץ לענף של סיסמא לא נכונה), וסה"כ נוכל להסיק כי 2 הבתים הם $String[2] =' 5', String[3] = 'y'$.

נחזור ל sub_401080:

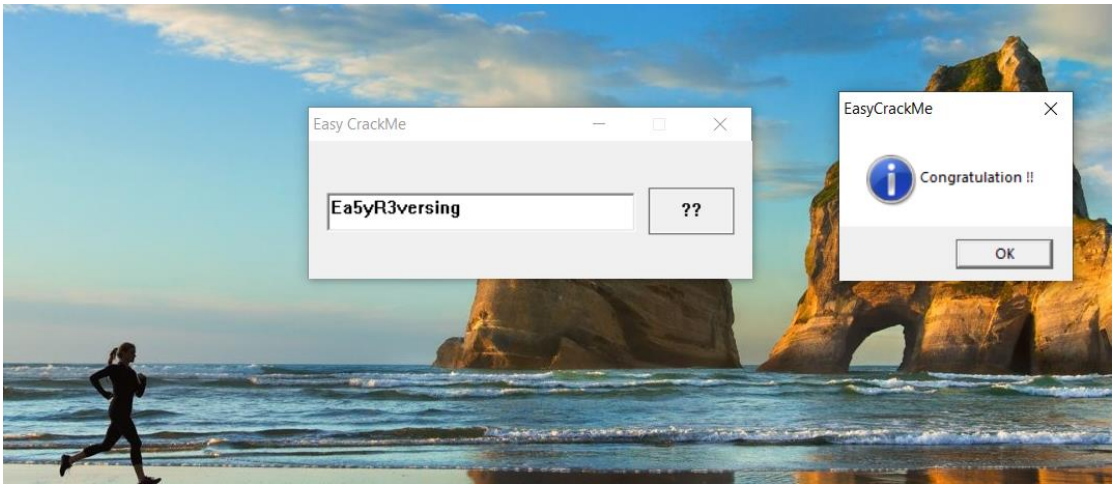

push ebx
push esi
mov esi, offset aR3versing ; "R3versing"
lea eax, [esp+70h+var_60]
loc_4010DA:
mov dl, [eax]
mov bl, [esi]
mov cl, dl
cmp dl, bl
jnz short loc_401102
test cl, cl
jz short loc_4010FE
mov dl, [eax+1]
mov bl, [esi+1]
mov cl, dl
cmp dl, bl
jnz short loc_401102
add eax, 2
add esi, 2
test cl, cl
jnz short loc_4010DA
loc_4010FE:
xor eax, eax
jmp short loc_401107
loc_401102:
sbb eax, eax
sbb eax, 0FFFFFFFFh
loc_401107:
pop esi
pop ebx

קודם כל ניתן לראות כי האוגר esi מצביע על המחרוזת "$R3versing$", והפקודות שמתבצעות לאחר מכן הם בעצם השוואה בין $String[4]$ למחרוזת זו:



ניתן לראות שאם אחד מהתווים במהלך ההשוואה אינו יהיה זהה המעבד יקפוץ לענף loc_401135 שהוא הענף שנגיע אליו עם הסיסמא לא נכונה, לכן נוכל להסיק כי "$String[4] = "R3versing$, ועד כאן הסיסמא הנכונה היא: "$\#a5yR3versing$" כאשר # התו הראשון אינו ידוע.

לבסוף, אנחנו משווים ב[@], בין התו הראשון לבין $0x45 = 'E'$ לפי קידוד $'ASCII'$, אם התו הראשון אינו $'E'$ נקפוץ לענף של הסיסמא הלא נכונה, ואם הוא כן $'E'$ יודפס לנו "$Congratulation$!!" ונסיק כי הסיסמא היא $Ea5yR3versing$. נחזיק אצבעות וננסה את הסיסמא הזו:



והצלחנו לפצח את הסיסמא באמצעות $reverse\ engineering$.

מקווה שהצלחתי לעניין אתכם ולהדגים לכם באיזו קלות אפשר לפרוץ סיסמאות.