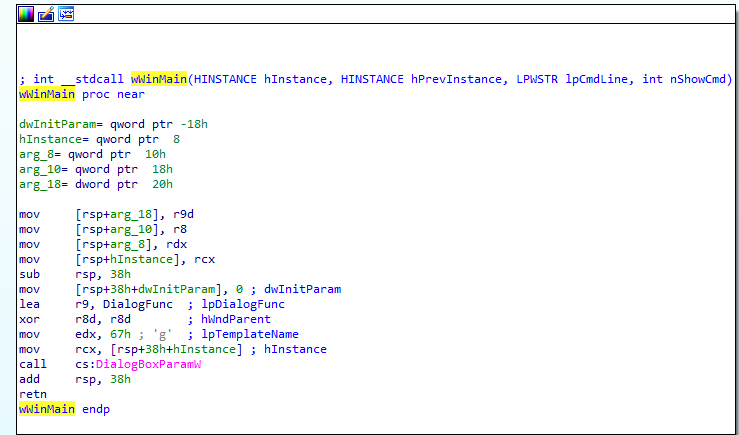
פתרון לcrackme64

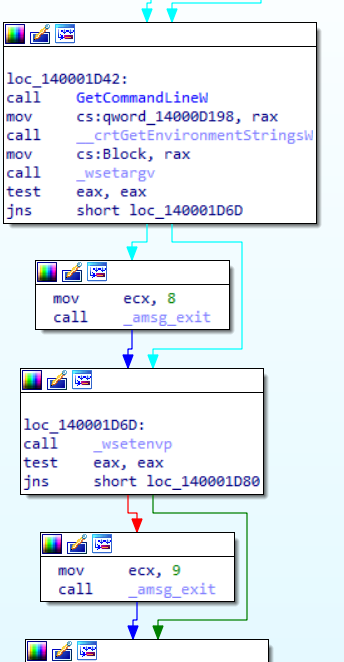
רז אברג'ל 313575185

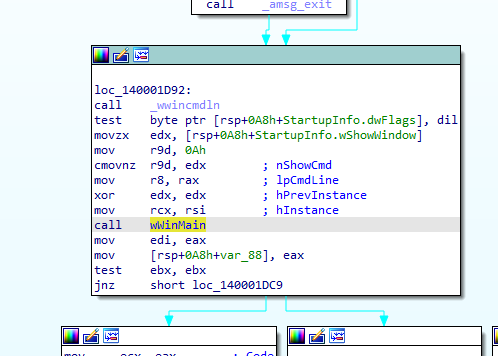
מציאת המיין היא דיי פשוטה ,אם נכנס לida נוכל לראות שהגרף נפתח לנו למעשה במיין עצמו.

wWinMain זוהי למעשה המיין של כל תוכנית עם ממשק גרפי שכתובה בC (לפי msdn )



נוכל לראות גם מאיפה קוראים למיין ונוכל לראות שלפני הקריאה למיין אנו מטפלים בפרמטרים

שמיין מקבלת 

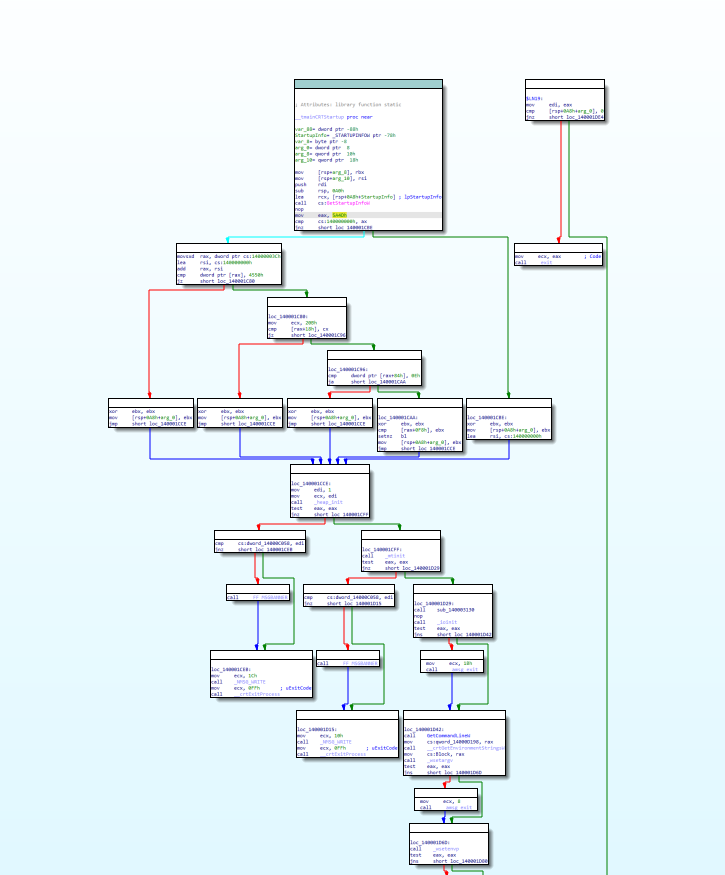


**? MAIN האם רץ קוד לפני ה**

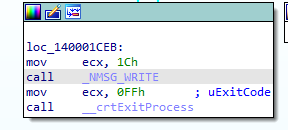
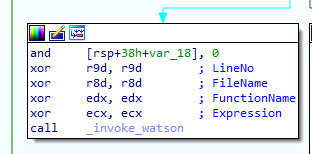
**פירוט מנגנוני אנטידיבאג, מה הם בודקים?**

נוכל לראות שיש קטע קוד גדול

לפני הקריאה למיין

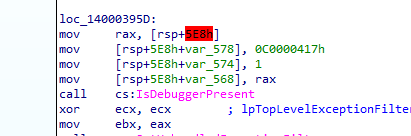


כשחוקרים את הקוד ב איידה אפשר למצוא את הפונקציה לפני הקריאה למיין

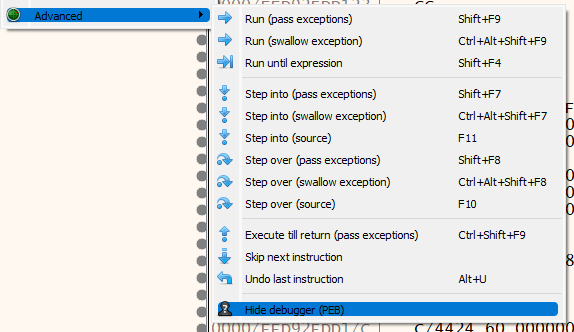
נכנס לכאן  ואז לכאן

נראה שיש כמה קריאות לפונקציה invoke Watson **לפני** הקריאה למיין.

נכנס אליה ונוכל לראות את המנגנון IsDebuggerPresent

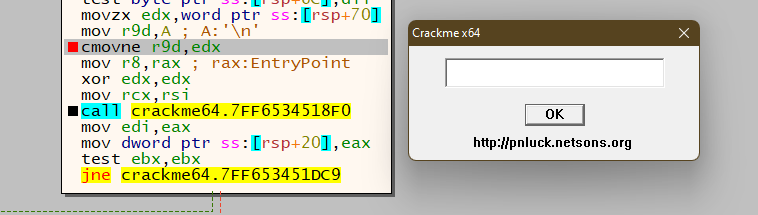


כדי לנטרל את המנגנון הזה נוכל לפצפץ את הקוד או שנוכל פשוט לשנות השדה שמודיע על דיבוג במבנה הנתונים PEB נוכל לשנות את הערך בזכרון אבל לא נעשה זאת באופן הזה אלא פשוט נשתמש בפיצ'ר המובנה של x32 שמאפשר לנו לאפס את השדה



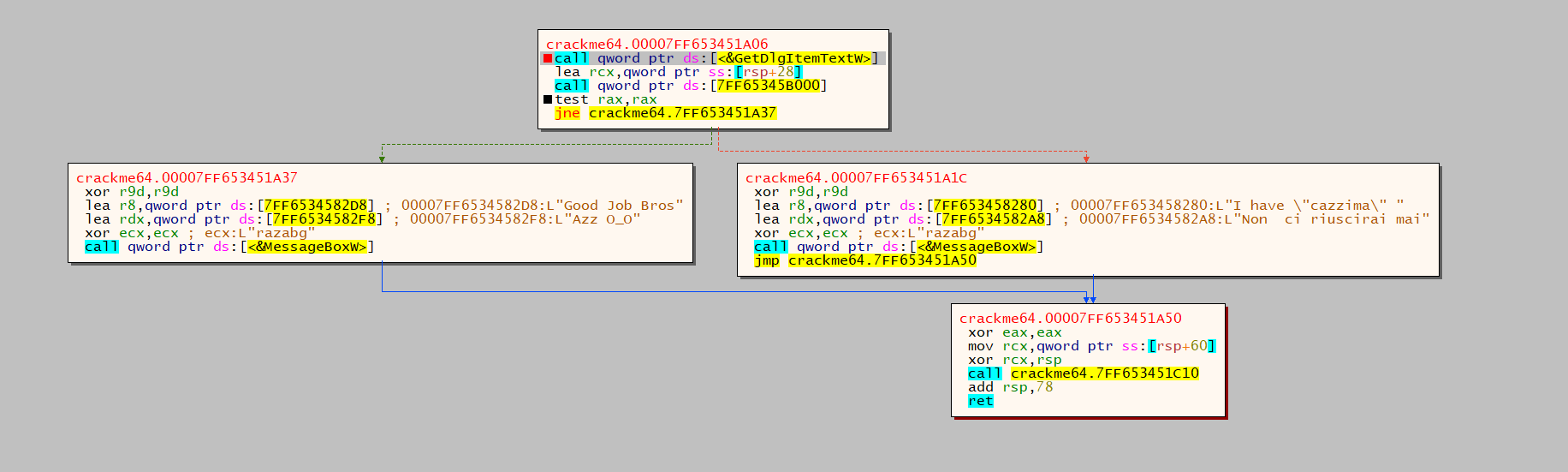
עכשיו שנריץ את התוכנית נוכל לראות שהחלון הראשי של התוכנית עולה

מה שאומר שהתגברנו על מנגנון זה

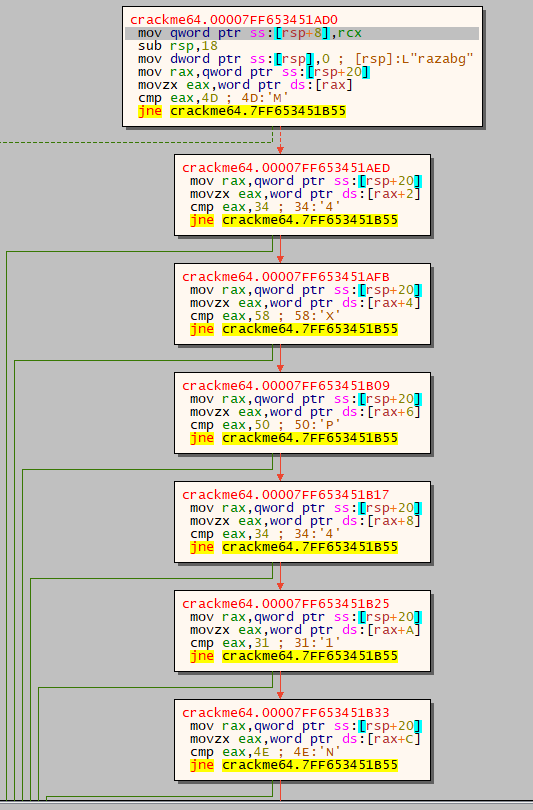


היכן נמצאת הבדיקה של הסיסמה ומה עשיתם כדי להצליח לעקוף את הבדיקה?

לאחר שהזנתי קלט רנדומלי זיהיתי שלפני הדפסת ההודעה יש לנו את הפונקציה הזאת



בזמן דיבוג נוכל לראות שהפונקציה משתנה תוך כדי ריצה והיא נראית כך



אפשר לראות שיש פה כמה השוואות לכמה קלטים

ומכאן אנחנו למדים שזוהי הסיסמא M4XP41N

אנחנו לא נזין אותה אלא נעקוף אותה ע"י פצפוץ של הקוד

נשנה את פקודת הקפיצה כדי להגיע לאזור הטוב



נשנה ל 

וככה נגיע לאזור הטוב עם ההודעה של ההצלחה

