**תרגיל בית 2 – פירוט**

לפי התרשים safe.pdf המטרה הסופית לפיצוח הכספת היא Safe Password, שאליה אין לנו גישה. אבל נוכל להשיג דרך הכספת של ה-wizard את הסיסמה כשהיא מוצפנת, את המקודד ומפתח ההצפנה נשיג דרך goblin ואת הפרמטרים לקידוד דרך giant

* **:wizard**

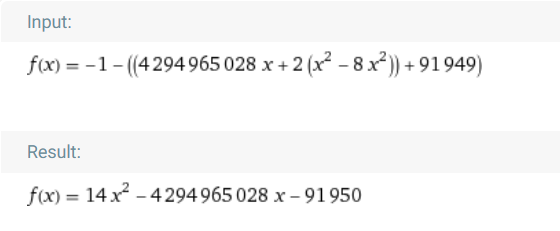
הורדנו את safe.exe מהכספת הפרטית של wizard וניתחנו את הקובץ עם IDA.

ניתן לראות כי בתחילת הקובץ קוראים לפונקציה TlsCallback, מה שלהבנתנו מסבך את המעבר עם דיבאגר, זאת היות ו-TlsCallback לוקח שליטה לפני נקודת הכניסה לתכנית. לכן, המשכנו בניתוח סטטי של הקוד ומצאנו לוגיקה המתחילה בהכנסת ה-magic value DEADBEEF למחסנית. ניתן לראות כי מצפים לקלט ראשוני, שכרגע לערכו אין חשיבות, מלבד להיותו מספר בייצוג hexa, ולאחר מכן ממשיכים. בעת הזנת המספר, בודקים אם הוא שלילי ואם כן הופכים אותו לחיובי ע"י neg. במידה ונכניס מספר אקראי, בסוף יודפס לנו ג'יבריש. הדרך לבצע unhash היא ע"י פונקציה שאליה נכנסים רק בהמשך במידה והפרמטר הנ"ל שלילי, מה שלא יקרה לרוב כי ביצעו לו neg. לכן, המספר היחיד שישאר עדיין שלילי גם לאחר ביצוע neg הוא MAX\_INT. מאחר והקלט בשלב זה הוא ב-hexa, נכניס 80000000, עליו יבוצע neg מבלי לשנותו, וכך נגיע גם לחלק שעושה unhash בהמשך.

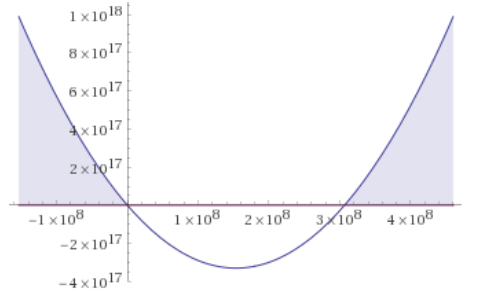
כעת קוראים שלושה מספרים למחסנית אל ebp-10, ebp-14, ebp-18. מבצעים מספר פעולות אריתמטיות, מעתיקים את המספר הראשון שנקלט ומשווים אותו ל-196 (C4). אם הוא שונה, מסתיימת התכנית. לכן המספר הראשון שנכניס הוא 196.

אחרי כן יש בדיקה שהמספר השני בקלט הוא קטן מאפס, אחרת התכנית מסיימת. לכן, המספר השני שנכניס הוא -1.

לאחר מכן, בודקים אם תוצאת הפעולה האריתמטית שביצענו, בעת חיסורה מהמספר השני שקלטנו, גדול אפס, ואם כן מסיימים את הריצה. בעזרת WolframAlpha חישבנו את הפולינום המתקבל מהאריתמטיקה וראינו מתי הוא מתאפס:



והגרף המתקבל הוא:



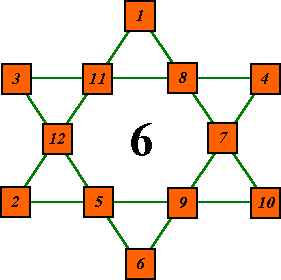
עבור מספרים בעלי ייצוג ארוך, למשל מאוד גדולים, יתכן שיחתכו חלק מהביטים בעת ביצוע הפעולות האריתמטיות, אך בקירוב ניתן לראות כי עבור מספרים גדולים מאוד נעבור את בדיקת הקלט.

במידה ועמדנו בכל תנאי הקלט הנ"ל, נקבל הדפסה של מגן דוד עם סימני שאלה מפוזרים. עם סיום ההדפסה, התוכנית מחכה לקלט של 12 מספרים בין 0 ל-11 ומבצעת בדיקת קלט עבור כל אחד מהם. בנוסף, עבור כל מספר שמוזן התכנית בודקת שלא הוזן קודם לכן, כך שיוזנו כל המספרים בין 0 ל-11.

ניתן לראות כי קודם לכן, נדחפו על המחסנית המספרים 1-12 בסדר מסוים, כך שבעת הזנת הקלט שלנו, כל מספר שנזין מוחלף במספר כלשהו מהמחסנית לפי התאמתו, כלומר – המספר 0 מתאים למספר הראשון על המחסנית וכך הלאה עד שלבסוף המספר 11 מתאים למספר האחרון מקבוצת המספרים על המחסנית.

לאחר מכן, הקוד ממשיך בוידוא שהקלט שלנו פותר את החידה כך שהוא סוכם את ארבעת המספרים בכל צלע במגן דוד ומוודא שסכומם הוא 26 כאשר על כל שורה שעמדה בתנאי הוא מגדיל מונה ב-1. לבסוף, הוא סוכם גם את כל המספרים על היקפו של המגן דוד ומוודא שגם סכומם 26, כך שאם עמדנו בתנאים המונה יעמוד על 7 (6 צלעות + היקף).

ניתן למצוא את פתרון החידה באינטרנט ולהגיע לפתרון הבא:



C:\Program Files\Microsoft VS Code\bin\codeכעת, כל שעלינו לעשות הוא למצוא את המספרים שעלינו להזין על מנת שהחלפת המספרים תביא לפתרון הנ"ל, כאשר סדר המספרים (לפי הוידוא של הקוד) הוא מלמעלה למטה ומשמאל לימין. על מנת לקבל את הרצף:

1 3 11 8 4 12 7 2 5 9 10 6, עלינו להזין 7 11 5 0 3 6 9 10 1 8 4 2 ובעת הזנת רצף זה נקבל הודעת הצלחה.

לאחר מכן על המחסנית טוענים את שתי מחרוזות,

3+U3waA$tMzvU$lX=]k>3|E , hacked this

התוכנית מחכה לקלט, וכשמקבלת משווה אותו למחרוזות האלה כל פעם עד אורך המחרוזת המקורית. אז הקלט המכיל את שתיהן מופרדות ברווח נותן

f82e1d49F17D72136a7d7b7b4017731E

* **goblin**

מפתח הצפנה

לאחר בחינה מהירה של safe.exe ניתן לראות שהתוכנית קוראת לrand ובוחרת ערך רנדומלי מהמילון (dictionary.txt). אז רצינו לחפש באיזה מסלולי חישוב יש קריאה פונקציה שתחזיר פלט, אז חיפשנו מחרוזות והבחנו בשתי מחרוזות \\\\.\\pipe\\SuperSecretPipe \\\\.\\pipe\\SecretAnswerPip

חיפשנו את הגישות אליהן בקובץ ומתברר שהן נשלחות כפרמטר לפונקציות winapi: CreateFileA ו-CreateNamedPipeA. חיפשנו בתיעוד של מיקרוסופט והבנו שאחת יוצרת pipe server והשניה מתחילה להאזין ל-pipe server קיים כלומר התוכנית מאזינה על SuperSecretPipe ושולחת דרך SecretAnswerPipe.

כתבנו תוכנית שמבצעת אותו הדבר בתפקידים הפוכים: מאזינה ל-SecretAnswerPipe ושולחת דרך SuperSecretPipe.

(התוכנית קודם אתחלה את ה-pipe server ואחר כך בדקה אם קיים ה-pipe המרוחק)

כעת בדקנו מה מצפה לקבל התוכנית, שכן היא קודם כל מאזינה לבקשות.

שמנו לב שהיא משווה את המחרוזת שהתקבלה מה-pipe לזאת שבחרה מוקדם יותר מהמילון ואם אינן זהות היא מחכה לבקשה נוספת.

אז הסקנו שנוכל לעבור על כל המחרוזות מהמילון עד שנפגע בזו שנבחרה באקראי. בתוכנית שכתבנו פתחנו את קובץ המילון ושלחנו לתוכנית את כל המחרוזות ממנו.

הוספנו המתנה בתחילת התוכנית כדי שנוכל להפעילן במקביל, הרצנו וקיבלנו

The encryption super secret key is (?FRcQ,AfcjZrdkv