# פתרון CTF5

* לצורך הפיתרון נאלץ לבצע Reverse Engineering לקוד של התוכנית Minecraft .
* ניתן להשתמש בשלל תוכנות אך אני בחרתי להשתמש בתוכנה שמצאתי בעת מחקר על הנושא טרם הקורס, שפותחה על ידי הNSA בשם Ghidra. (התוכנה חסומה להורדה משום מה \*פרצוף מופתע\* לכתובות אייפי מישראל אז אפשר להוריד עם VPN).
* נפתח את הקובץ ה"זדוני" באמצעות התוכנה..
* נתבונן בפונקציית הMain שלנו -

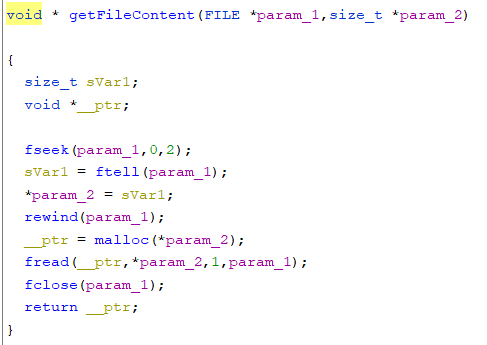


* תחילה יש אתחול של שלל משתנים (שורות קוד רבות הינן אתחולים של משתנים ואנו נרצה בעיקר להתייחס ללוגיקה של התוכנית..) ולאחר מכן ניתן לראות כי התוכנית מבצעת קריאה של שתי תיקיות – הראשונה כשמה כן היא – תיקיית קבצי המשתמש.
* התיקייה השניה תיווצר במידה ואינה קיימת וככל הנראה שייכת לתוכנית ההצפנה



* ניתן לראות כי התוכנית מקבלת את שמות הקבצים לשימוש מאוחר יותר להצפנתם..
* במידה והקובץ הינו תיקייה, התוכנית מתעלמת ממנו.
* מתבצעת קריאה ל3 פונקציות עיקריות אותן נצטרך לחקור..

נתבונן בפונקציה הראשונה – GetFileContent –

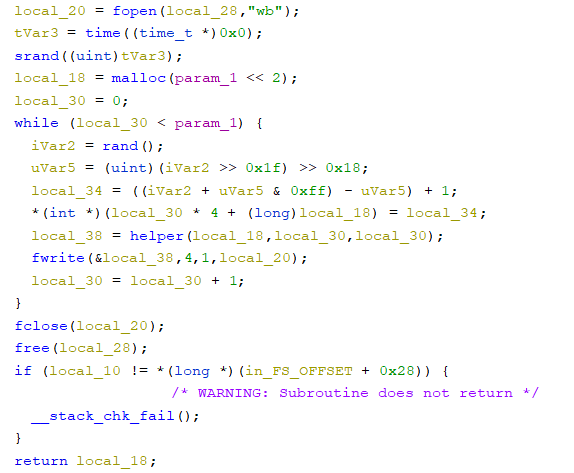


* ניתן לראות שמתבצעת קריאה לפונקציה מובנית ftell שמעבירה למשתנה את אורך תוכן הקובץ..
* לאחר מכן מתבצעת קריאה של כלל התוכן של הקובץ אל תוך מערך, וזאת אנו רואים בקריאה לfread, (קריאה בבתים כאורך המשתנה שנקבע קודם לכן).
* הפונקציה מחזירה פוינטר למערך המחזיק את התוכן בבתים ..

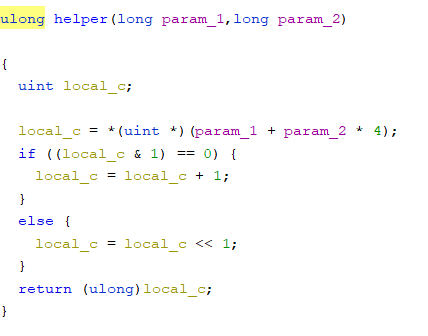
נעבור לפונקציה CreateRandomOrder -

את תחילת הפונקציה לא אציג כיוון שהיא מראה רק אתחול משתנים..

* תחילה הפונקציה פותחת קובץ לכתיבה בינארית
* קיים כאן שימוש בפונקציית rand שנעזרת בזמן הנוכחי לספק מספרים רנדומליים ככל שיהיו.
* Param1 מייצג את אורך הקובץ אותו אנו קוראים והלולאה תרוץ עד לאורך זה.
* iVar2 מקבל מספר רנדומלי גבוה מהפונקציה rand.
* Uvar5 יקבל 0 לכל ערך רדנומלי שמתקבל עבור int ולכן עבור הערך הלוקלי local34 נקבל את - (הערך הרנדומלי & 255 ) + 1 שזה שקול לפעולת מודולו 256 + 1.. (נקבל ערכים רנדומלים בין 1 ל256..)
* Local30 הינו האינדקס שלנו, local18 הינו מערך בגודל מספר הבתים שבקובץ.  
  כלומר המיקום במערך באינדקס הנוכחי מקבל את הערך הרנדומלי ממקודם..
* כעת מתבצעת קריאה לפונקציה שמחזירה ערך למשתנה חדש.. נתבונן בפונקציה



נתבונן בפונקציה helper –

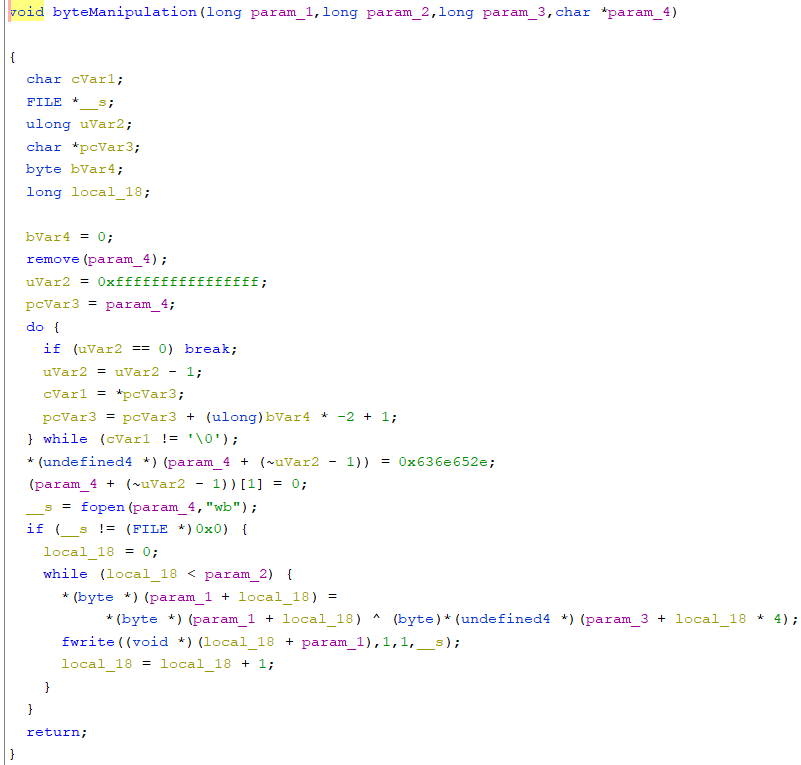


* הפונקציה מקבלת את מערך הבתים וכן את האינדקס הנוכחי מהלולאה בפונקציה הקודמת
* Local\_c מקבל את הבית הנוכחי במערך..
* באם הערך הנוכחי הינו זוגי, נוסיף לו 1.
* באם הוא אי זוגי, נכפילו ב2.

הערך החדש מוחזר לפונקציה הקודמת ונכתב לקובץ כבית אחד.

\*\* נשים לב כי המערך הרדנומלי ממקודם אינו משתנה בשום שלב!

כעת נסתכל בפונקציה האחרונה שמתבצעת בתוכנית - ByteManipulation

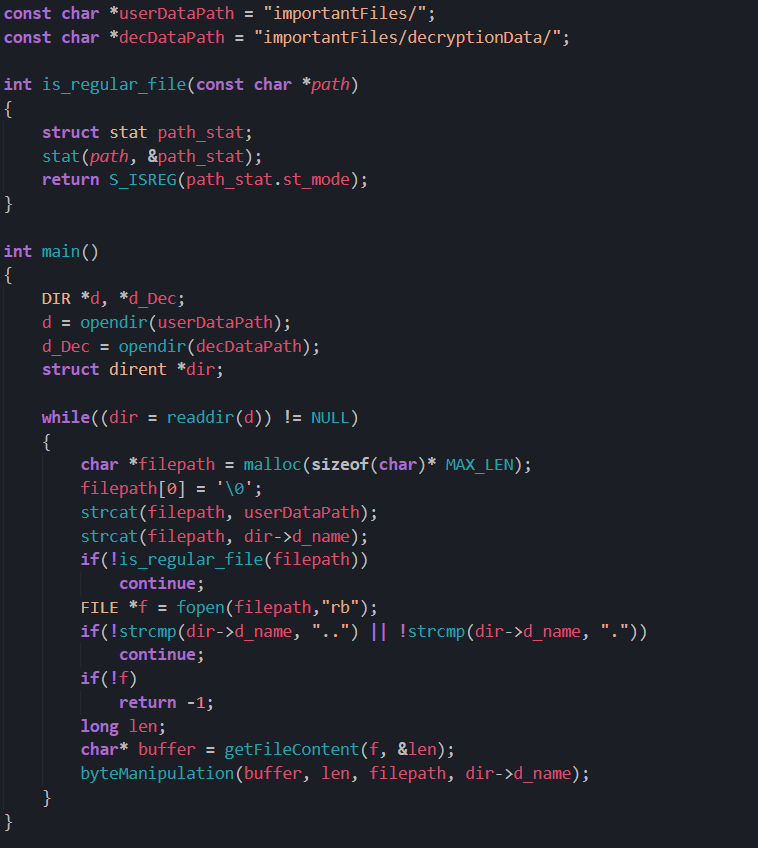


* מתבצעת פתיחה של קובץ חדש עם אותו השם ממוקדם לכתיבה בינארית.
* מהתמונה הראשונה נוכל להסיק כי param\_2 הינו אורך הקובץ ועל כן הלולאה תרוץ על מספר הבתים שבו..
* כל איבר במערך הבתים של הקובץ מבצע פעולת XOR עם מיקומו במערך הרנדומלי שנוצר קודם לכן!

ראשית ניתן לראות כי מתבצעת מחיקה של הקובץ הנוכחי המקורי ממחשב המשתמש

למעשה בשלב זה יש בידינו את כל המידע שאנו צריכים על מנת לכתוב תוכנית שמפענחת את הקבצים המוצפנים לפי הלוגיקה שנראיתה כאן. (כל מה שראינו בצורה הפוכה) .

פעולת XOR עם אותם הערכים תיתן לנו את הערכים המקוריים של הבתים בקבצים הישנים שלנו ונקבל אותם כפי שהיו.



נבנה תוכנית C שתפענח לנו את הקבצים –

התוכנית קוראת מתיקיה את כל הקבצים שלה שאינם תיקיות בעצמם, פותחת אותם לקריאה אחד  
אחרי השני..

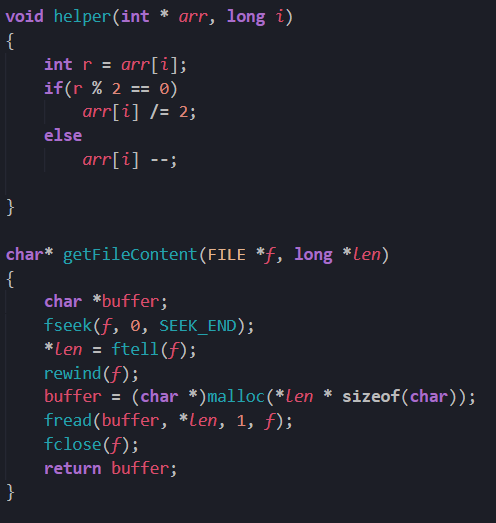
התוכנית קוראת לgetfilecontent שיעביר את תוכן  
הקובץ הנוכחי אל מערך ואת המערך של הבתים תעביר אל הפונקציה ByteManipulation..

ByteManipulation מוחק תחילה את הקובץ עליו עובדים, מוסיף לשם הקובץ אותו נייצר סיומת dec   
ואחר כך מייצר אותו.

הפונקציה קוראת לgetDecData שמחפשת בתיקית decryptionData את שם הקובץ המוצפן שם יש את המידע על הפענוח..

הפונקציה קוראת את הקובץ המתאים עד לסופו ומעבירה למערך אל הפונקציה שקראה לה..

Bytemanipulation כעת תעבור בלולאה על אורך המערך שקיבלנו ותקרא לפונקציה נוספת – Helper..

הפונקציה הזאת לוקחת את כל אחד מהערכים במערך ובהתאם לזוגיות שלו מבצעת עליו פעולה.

אם זוגי נחלק ב2, אם אי זוגי נחסיר 1.

המערך החדש חוזר אל bytemanipulation שם מתבצעת פעולת Xor בין המערך החדש לתוכן הקובץ אותו אנו מפענחים ולאחריה כתיבה לקובץ המפעונח..

התוצאה – כל הקבצים יפוענחו !

נקמפל ונריץ ונקבל את הקבצים המקוריים..





קיבלנו מתכונים בשלושה קבצי Word..

בסוף כל קובץ יש חלק מהאפיקומן –







נרכיב אותם יחדיו ונקבל את הפיתרון שלנו – **Afikoman{ChocolateIsLife}**

תוכנית ההצפנה המקורית וכן תוכנית הפענוח מופיעות בתיקיה.