TP 2 – EL ARIF AMINE

1. Lire le fichier en ImHex

La première partie était de définir les parties intéressantes dans le fichier binaire et pour faire cela on a définie une struct.

struct fileStruct{

char metadata[0x248];

char data[0x56E6] ;

char endOfFile[0x8A] ;

};

La meta-data est la première partie du fichier contenant des informations sur le fichier.

Le data est le contenue du ficher qu’on cherche à décoder.

Le endOfFile est la partie finale du fichier qui commence à partir de STOP.

Ensuite on passe au contenu de data, on cherche à trouver les cordonnées du GPS (latitude, longitude), grâce à l’indice donné par nos profs, j’ai pu définir cet partie on utilisant le struct suivant :

struct dataframe

{

    u8 offset[26];

    u8 longitude[4];

    u8 latitude[4];

    u8 offsetTwo[15];

};

Cet partie va se répéter tout au long du partie data, grâce à cet partie :

dataframe frame[454] @ 0x248;

et finalement on doit définir la structure en codant que ca commence à partir de l’adresse 0x00 :

fileStruct var @ 0x00;

résultat :

A screenshot of a computer

Description automatically generated

1. Script python pour lire les coordonnées en csv

with open('13caaf25c97e29c8b45ae2e4f9f8ac3b\_T0\_2e526', 'rb') as file:

    file.seek(0x248)

    trames\_bin = file.read()

    trames\_size = len(trames\_bin)

    trames\_count = trames\_size // frame\_struct.size

  trames=[frame\_struct.unpack(trames\_bin[i\*frame\_struct.size:(i+1)\*frame\_struct.size]) for i in range(trames\_count)]

Cette partie du code lit le fichier binaire T0, se déplace à la position 0x248 (ignore la méta-data), lit le reste du fichier, calcule la taille et le nombre de trames, puis décode chaque trame du fichier.

with open('gpsFile.csv', 'w', newline='') as csvfile:

    fieldnames = ['Latitude', 'Longitude']

    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)

    writer.writeheader()

ensuite je crée un fichier CSV nommé ‘gpsFile.csv’, définit les en-têtes de colonnes comme ‘Latitude’ et ‘Longitude’, puis j’écris ces en-têtes dans le fichier.

 for trame in trames:

        latitude =  '{:.6f}'.format(trame[1] / 10000000)

        longitude = '{:.6f}'.format(trame[2] / 10000000)

        writer.writerow({'Latitude': latitude, 'Longitude': longitude})

Je formate ensuite les coordonnées au bon format.

Résultat ! :

A map with a red outline

Description automatically generated