En este código tenemos parte de la unidad 1 y 3 en el cual tenemos un código el primero obliga a un espacio de memoria aleatorio a no estar en la memoria caché del procesador para luego mida el tiempo que le toma acceder a este espacio de memoria el cual si es menor a 120 este va a guardar el espacio de memoria y guardar e imprimir en hexadecimal el contenido de la memoria. Y al final guarde esta información en un .txt llamado Robo

#include <stdio.h>

#include <stdint.h>

#include <stdlib.h>

#include <x86intrin.h>

#include <time.h>

#define ARRAY\_SIZE 256 \* 4096

unsigned char array[ARRAY\_SIZE]; //Modificador

uint64\_t access\_times[256]; //Entero sin signo de 64 bits

//-----------------------------------------------------------------------------

//Invalidar una linea especifica de cache en la memoria, esto para asegurarnos que una dirección no este

//en la memoria cache del procesador

void flush\_cache\_line(unsigned char\* addr) {

\_mm\_clflush(addr);

}

//Imprimir el contenido de la memoria a partir de una direccion de memoria dada y para una

//longitud especifica de bytes

void print\_memory\_contents\_to\_file(FILE\* fp, unsigned char\* addr, size\_t length) {

for (size\_t i = 0; i < length; i++) {

fprintf(fp, "Address %p: 0x%02x\n", (void\*)(addr + i), addr[i]);

}

}

//-----------------------------------------------------------------------------

int main(int argc, char\*\* argv) {

int junk = 0;

register uint64\_t time1, time2;

volatile unsigned char\* ptr;

unsigned char\* secret\_data;

// Inicializa el array

for (int i = 0; i < 256; i++) {

flush\_cache\_line(&array[i \* 4096]);

}

// Inicializar la semilla del generador de números aleatorios

srand(time(NULL));

// Asignar memoria para los datos secretos dentro del array

secret\_data = array + (rand() % (ARRAY\_SIZE - 1));

// Acceder a los datos secretos

junk = \*secret\_data;

// Medir el tiempo que toma acceder a los elementos del array

for (int i = 0; i < 256; i++) {

time1 = \_\_rdtscp(&junk);

junk = array[i \* 4096];

time2 = \_\_rdtscp(&junk) - time1;

access\_times[i] = time2;

}

// Abrir archivo para escritura

FILE\* fp;

fp = fopen("Robo.txt", "w");

if (fp == NULL) {

perror("Error al abrir el archivo");

return 1;

}

// Escribir tiempos de acceso en el archivo

for (int i = 0; i < 256; i++) {

fprintf(fp, "Time for array[%d \* 4096]: %lu\n", i, access\_times[i]);

}

// Escribir contenido de memoria alrededor de los datos secretos

fprintf(fp, "\nMemory contents around the secret data:\n");

print\_memory\_contents\_to\_file(fp, secret\_data, 64);

// Cerrar el archivo

fclose(fp);

return 0;

}