НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ ТА ПРОГРАМУВАННЯ»

«Безпечне програмування»

*Звiт з лабораторної роботи №13-15*

*Тема: «Використання вразливості переповнення буферу в мові C»*

Виконав:

ст. гр. 16Б

Башкатов В.Є.

Харків – 2019

***Мета:*** навчитися використовувати вразливості переповнення буферу в мові C.

***Завдання***

Використовуючи відлагодник, не вводячи вірний пароль, а скориставшись помилкою переповнення буфера, зруйнувати / переписати стек викликів так, щоб програма при неправильній введеної послідовності символів видала Passed.

**Текст програми**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

bool isPasswordOk() {

char password[12];

gets(password);

return 0 == strcmp(password, "goodpass");

}

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

bool status;

puts("Enter password: ");

status = isPasswordOk();

if (!status) {

puts("Access denied");

system("pause");

exit(-1);

}

puts("Passed");

system("pause");

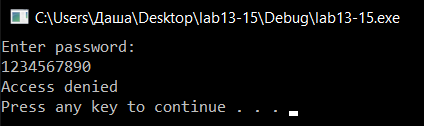
exit(-1);

return 0;

}

**Хід роботи:**

1. Руйнування стека відбувається, коли переповнення буфера перезаписує дані в пам'яті, виділеної для стека виконання. Це може мати серйозні наслідки для надійності і безпеки програми. Переповнення буфера в сегменті стека може дозволити зловмиснику змінювати значення автоматичних змінних або виконувати довільний код. Була створена програма, яка це показує. Вигляд програми у її нормальному стані представлено на рис. 1.



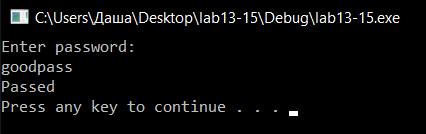


Рисунок 1 – Результат роботи програми

1. Запускаємо програму у режимі “Debug” та шукаємо адресу на яку треба “стрибнути”. Ця адреса представлена на рис. 2.

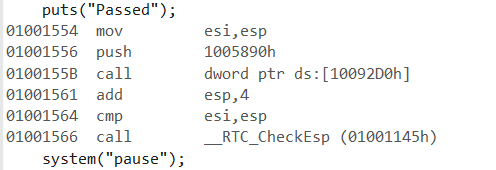


Рисунок 2 – Виклик функції

З рис. 2 можна побачити, що виклик функції “puts(‘Passed’)” починається за адресою 0x00001554. Зараз ми знаємо на яку адресу потрібно перейти. Далі цю адресу треба перевести у десятковий погляд. Це представлено на рис. 3.

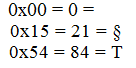


Рисунок 3 – Конвертація адреси

1. Тепер треба ввести строку у програму, яка приведена на рис. 4. Ця строчка зруйнує та перепише стек й у кінці буде отримано повідомлення про те, що пароль вірний. Це представлено на рис. 5.



Рисунок 4 – Строчка для руйнування стеку

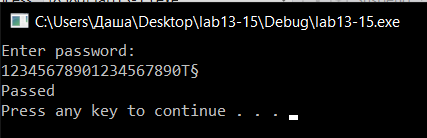


Рисунок 5 – Результат роботи програми

**Висновки:**

В результаті виконання лабораторної роботи було використано помилку переповнення буферу, зруйновано стек викликів програми з перевіркою паролю так, що програма при неправильній введеної послідовності символів видала Passed.