МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное учреждения высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт компьютерных технологий и информационной безопасности

Кафедра математического обеспечения и применения ЭВМ

**ЛАБОРАТОРНАЯ РАБОТА № 7**

по дисциплине

**«Безопасность информационных технологий»**

на тему:

**«Переполнение буффера»**

Выполнил:

Студент группы

КТбо2-8

Нестеренко П. А.

Проверил:

доцент кафедры

ИБТКС

Петров Д. А.

Оценка

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

**Цель работы:**

Исследование уязвимости «переполнение буфера» на примере тестовой программы и операционной системы Ubuntu/Linux.

**Порядок выполнения:**

1. Скорректировать тестовый файл «main.c» в соответствии с вариантом задания.
2. Скомпилировать программу командой «gcc -O0 -mpreferred-stack-boundary=2 -g -m32 -fnostack-protector main\_var?.c»
3. Протестировать полученную программу путём ввода неправильного пароля, правильного пароля и неправильного пароля, длина которого превышает размер выделенного буфера.
4. Запустить отладчик gdb, задать точку останова на функцию main() и получить результат дизассемблирования функции main() и buff\_overflow\_test().
5. При помощи пошаговой отладки программы в gdb посмотреть содержимое стека для буфера buff\_var для случаев произвольного короткого пароля и пароля, длина которого превышает размер буфера на 1 байт. Привести результат выполнения программы при вводе этих паролей.
6. В качестве отчёта по лабораторной работе привести результаты выполнения указанных выше пунктов, в том числе скорректированный исходный код и результаты работы отладчика gdb.

**Индивидуальный вариант задания:**

|  |  |
| --- | --- |
| Название исходного файла | main\_var2 |
| Название буфера | buff\_var2 |
| Размер буфера | 6 |
| Правильный пароль | pass2 |

**Скорректированный исходный код программы:**

#include <stdio.h>

#include <string.h>

int main(void)

{

buff\_overflow\_test();

return 0;

}

int buff\_overflow\_test()

{

char buff\_var2[6];

int pass = 0;

printf("\n Enter the password : \n");

gets(buff\_var0);

if(strcmp(buff\_var2, "pass2"))

{

printf ("\n Wrong Password \n");

}

else

{

printf ("\n Correct Password \n");

pass = 1;

}

if(pass)

{

/\* Now Give root or admin rights to user\*/

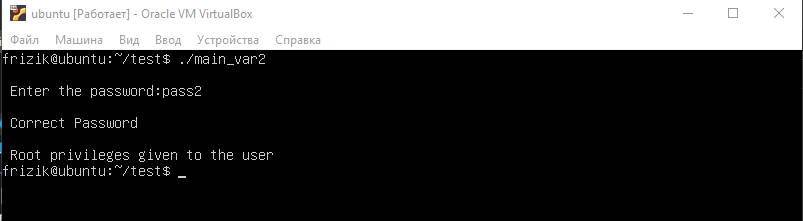
printf ("\n Root privileges given to the user \n");

}

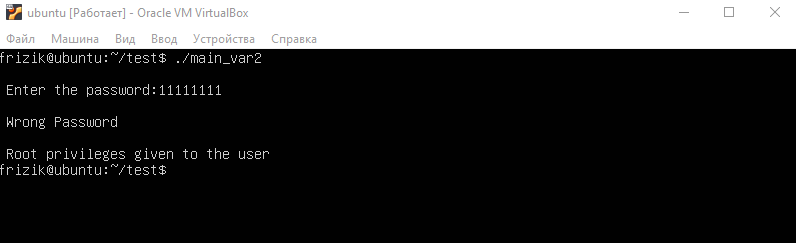
}



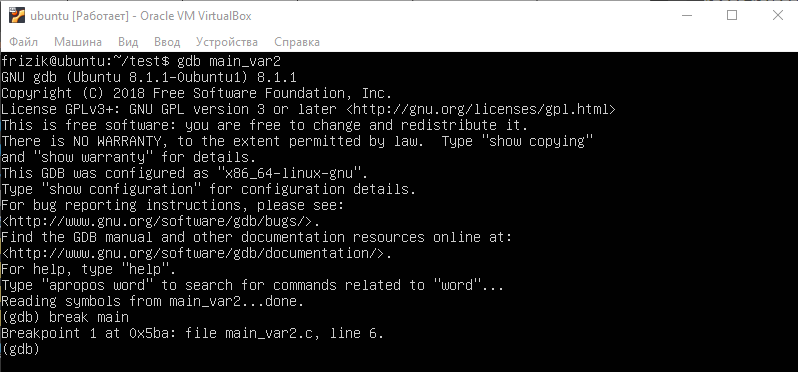
*Рисунок 1 – Попытка ввода неправильного пароля, длиной менее 6 символов*

**

*Рисунок 2 – Ввод правильного пароля*

**

*Рисунок 3 – Ввод неправильного пароля с выдачей прав*

**

*Рисунок 4 – Запуск* ***gdb*** *и добавление точки остановки программы*

**Результат команд disassemble main и disassemble buff\_overflow\_test**

|  |
| --- |
|  |
|  |

**Данные в регистре ESP при йвводе «AAA»**

|  |
| --- |
| (gdb) x/x $esp  0xffffs590: 0x41415960 |
| (gdb) x/s $esp  0xffffs590: "AAA" |
| (gdb) x/s $esp+5  0xffffs596: "" |
| (gdb) x/x $esp+5  0xffffs596: 0x00 |

**Данные в регистре ESP при вводе «AAAAAA»**

|  |
| --- |
| (gdb) x/x $esp  0xffffs590: 0x41415960 |
| (gdb) x/s $esp  0xffffs590: "AAAAAAA" |
| (gdb) x/x $esp+5  0xffffs596: 0x41 |
| (gdb) x/s $esp+5  0xffffs596: "A" |