**LAB 7**

**AIM:**

To manipulate the password program using the disassembled code to get access to the application.

**SOFTWARE REQUIRED:**

Code::Blocks, IDA Freeware

**INTRODUCTION:**

Assembly language is a low-level programming language for a computer or other programmable device.

The Interactive Disassembler (IDA) is a disassembler for computer software which generates assembly language source code from machine-executable code. It supports a variety of executable formats for different processors and operating systems.

IDA is used widely in software reverse engineering, including for malware analysis and software vulnerability research.

**Program**

**C++ code:**

#include<iostream>

using namespace std;

int main() {

string pass;

cout<<"Enter the password to continue:";

cin>>pass;

if (pass=="dinesh") {

cout<<"Welcome Dinesh!"; }

else {

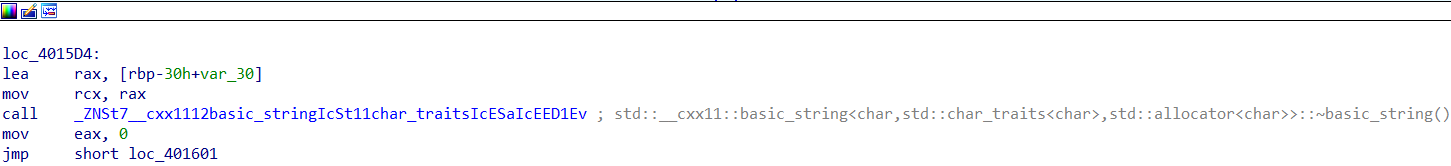
cout<<"Wrong Password!"; } }

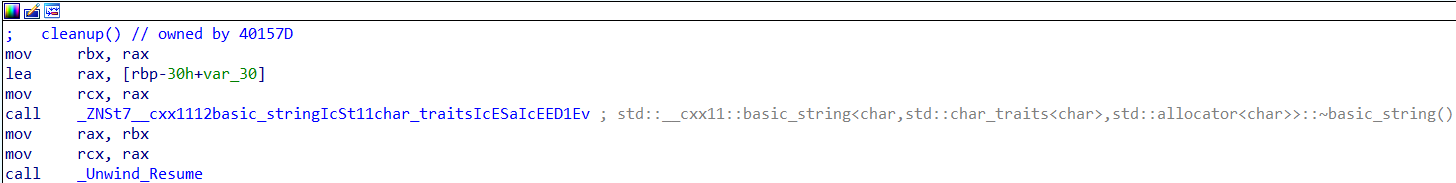
**Disassembled code:**

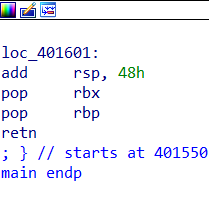
****

****

****

****

****

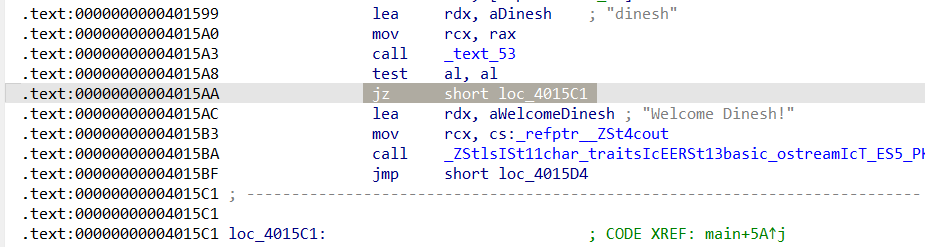
****

**Explanation:**

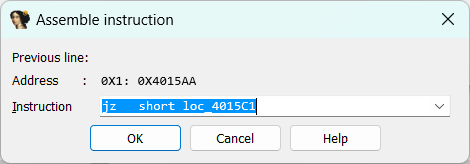
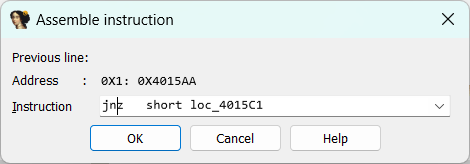
The code reads a password entered by the user, compares it with a string "dinesh", and displays a corresponding message based on the match or mismatch of the password. If matched, it shows “Welcome Dinesh!” and if mismatched, it shows “Wrong Password!”.

**Manipulated Code:**

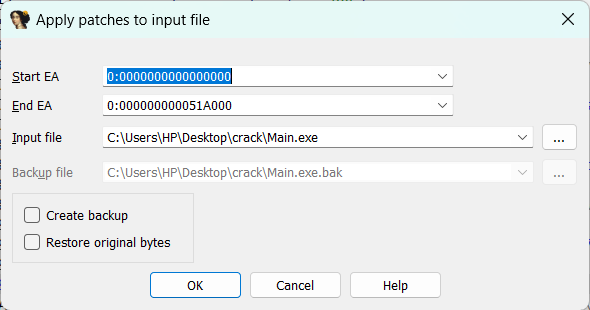
At loc\_4015AA, the disassembled code shows that if ‘zero’ is encountered, then jump to loc\_4015C1.



There we change the instruction at loc\_4015AA, replace jz jump with a jnz jump to jump to the loc\_4015C1 when the al register is not zero.

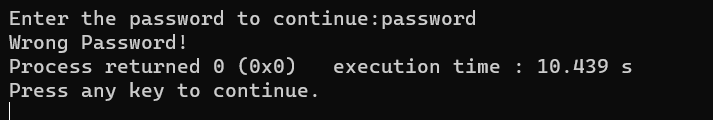
 

We apply the patch to the input file to make the change to the original file.

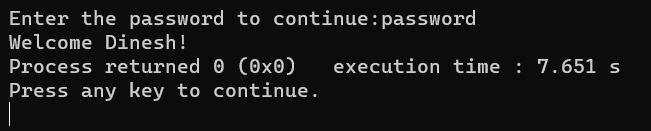


**Output:**

Before applying the Patch:



After applying the Patch:

****