**FINDING PASSWORD IN EXECUTABLE USING GDB**

**SUBJECT NAME**: CRYPTOGRAPHY AND NETWORK SECURITY

**SUBJECT CODE:** CS6008

**MODULE:** 1

|  |  |
| --- | --- |
| **NAME** | BHUVANESHWAR S |
| **REG.NO** | 2019103513 |
| **DATE** | 05/04/2022 |

**AIM:**

To find the password stored in an executable binary file using GDB debugger.

**TOOLS INVOLVED:**

* GCC
* GDB
* KALI- LINUX TERMINAL (WSL SUBSYSTEM)

**PROBLEM DESCRIPTION:**

An executable binary file where the password getting from user to compared with an actual password to login. Using gdb debugging to find to what actual password from exact location where memory saved. This memory location will find through gdb various command from gdb degugging to access.

**INPUT:**

Getting an input password from user in executable binary files.

**OUTPUT:**

Debug the binary code to find the what actual password is it.

**SCREEN SHOT:**

Filename : auth.c

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int check\_auth(char \*password\_buffer)

{

    int auth\_flag = 0;

    char \*p = "bhuvan";

    if (strcmp(password\_buffer, p) == 0)

    {

        auth\_flag = 1;

    }

    return auth\_flag;

}

int main()

{

    char password\_buffer[16];

    printf("[-] Enter the password : ");

    scanf("%s",password\_buffer);

    if (check\_auth(password\_buffer))

    {

        printf("\n-----------------------------------\n");

        printf("\tACCESS GRANTED\t");

        printf("\n-----------------------------------\n");

    }

    else

    {

        printf("\n-----------------------------------\n");

        printf("\tACCESS DENIED\t");

        printf("\n-----------------------------------\n");

    }

    return 0;

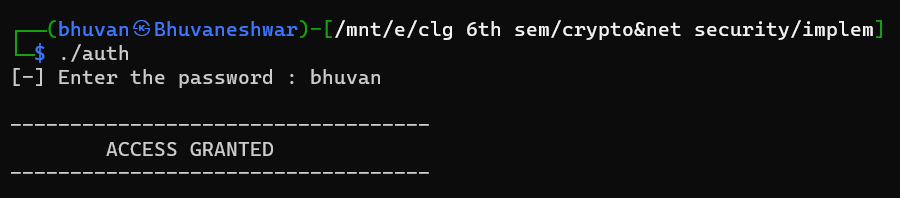
}

The program accepts a password from a command line argument and then call function check\_auth().This function compare the password whether it is equal it return 1 otherwise it return 0.

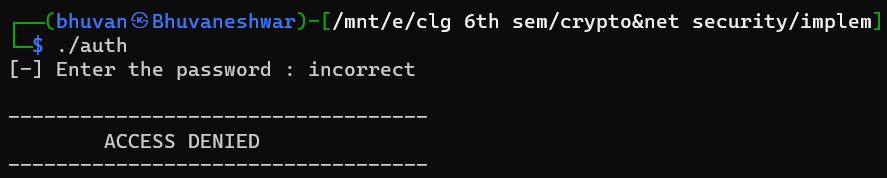
Compile this program using this command : gcc -g -o auth auth.c

Then run this program : ./auth

When user type correct password

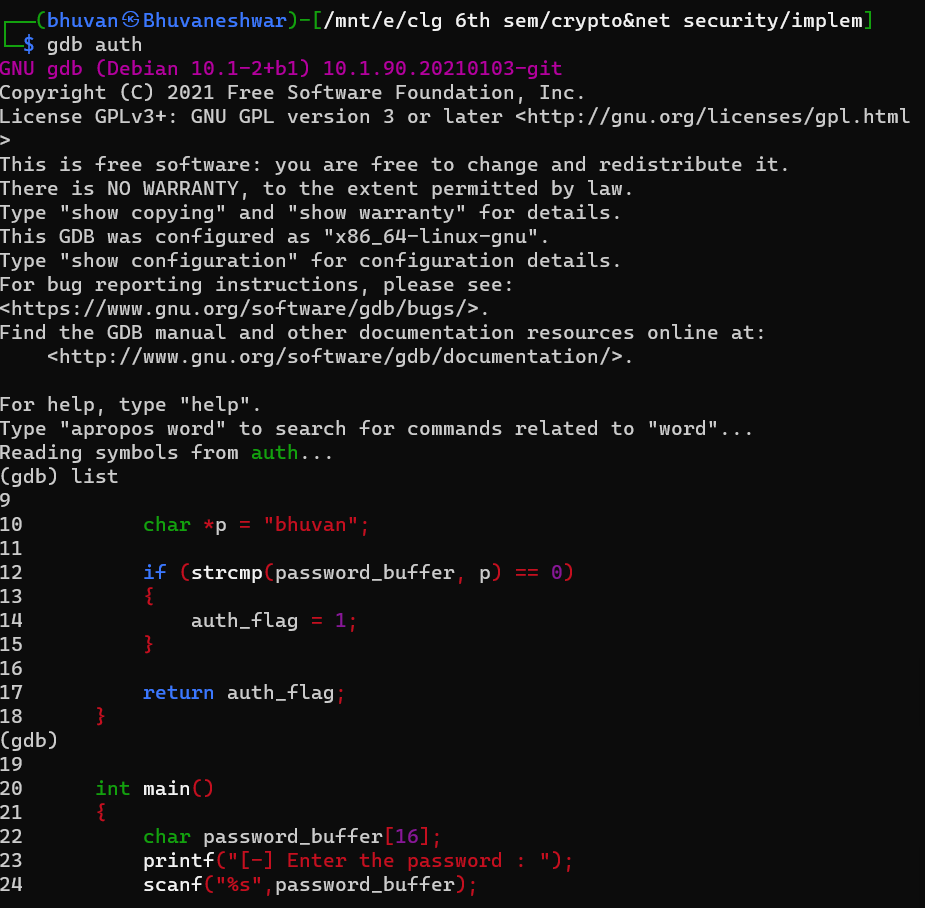


When user type incorrect password.

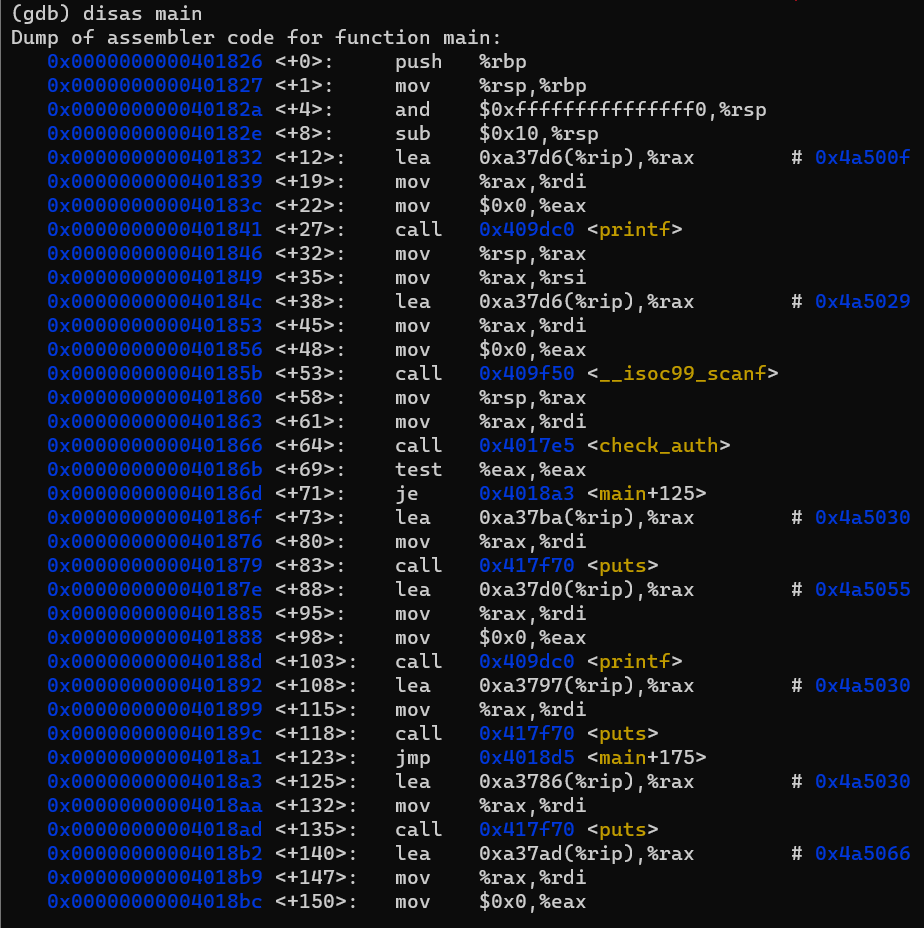


1. **CRACKING USING GDB DEBUG**

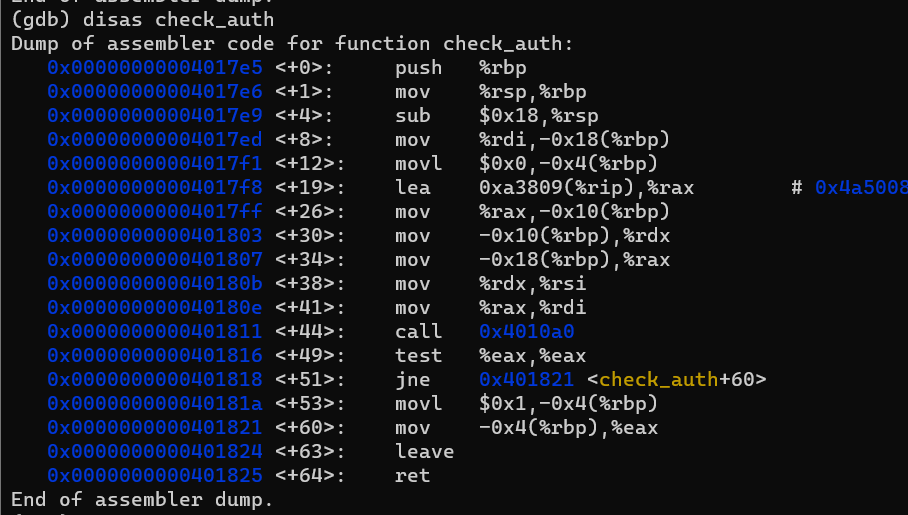
Let start with gdb debugging using this command : **gdb auth**

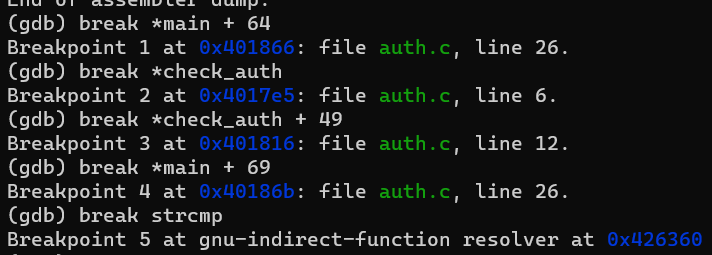


Disassemble main



Disassemble check\_auth





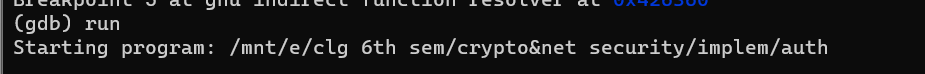
Break points are set **as \*main+64,\*check\_auth, check\_auth + 49, \*main + 69, strcmp**.

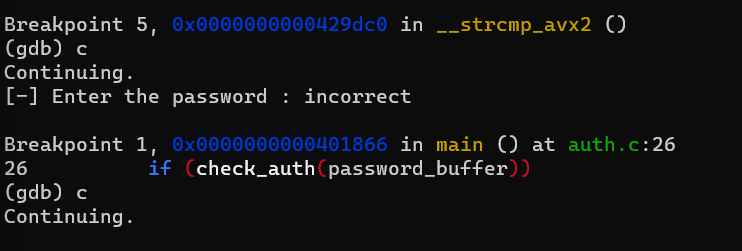
Where

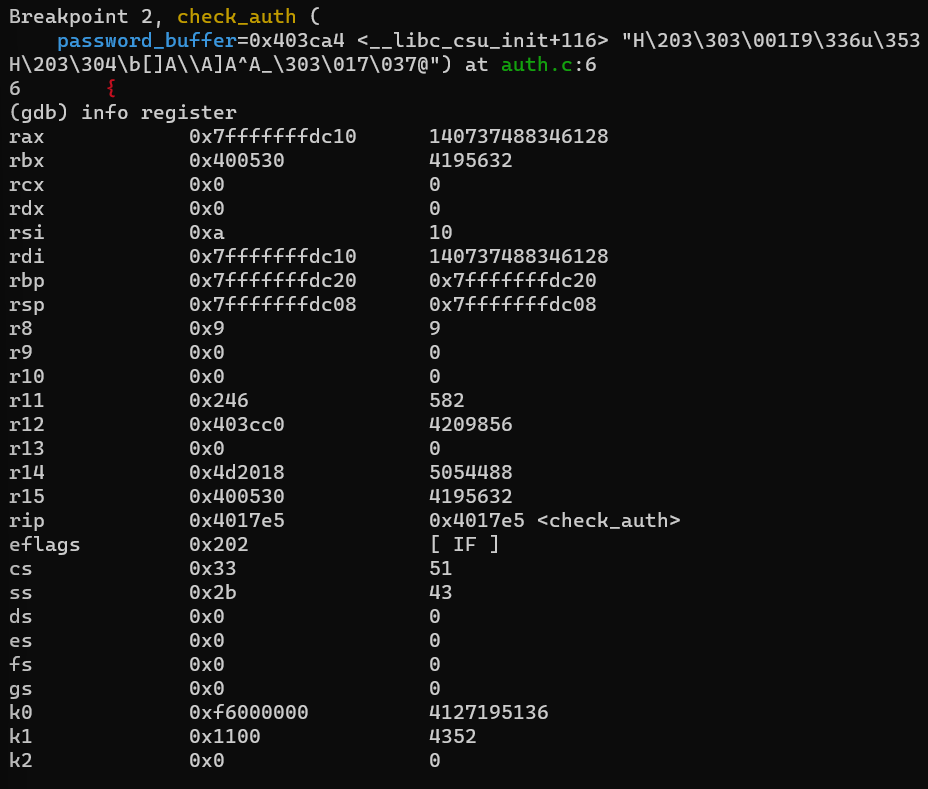
* **\*main + 64** => call functions check\_auth() to compare password
* **\*main + 69** => return an integer from check\_auth() function to check whether it is correct or not.
* **\*check\_auth + 49** => retrun an integer value from strcmp() function to compare the both string
* **Strcmp** => call strcmp() function through “string.h” library

The command **info registers** will shows the information about the state of register at that point in program execution. The **rsp** – stack pointer , show the current situation in stack memory.

Now let start run command : run





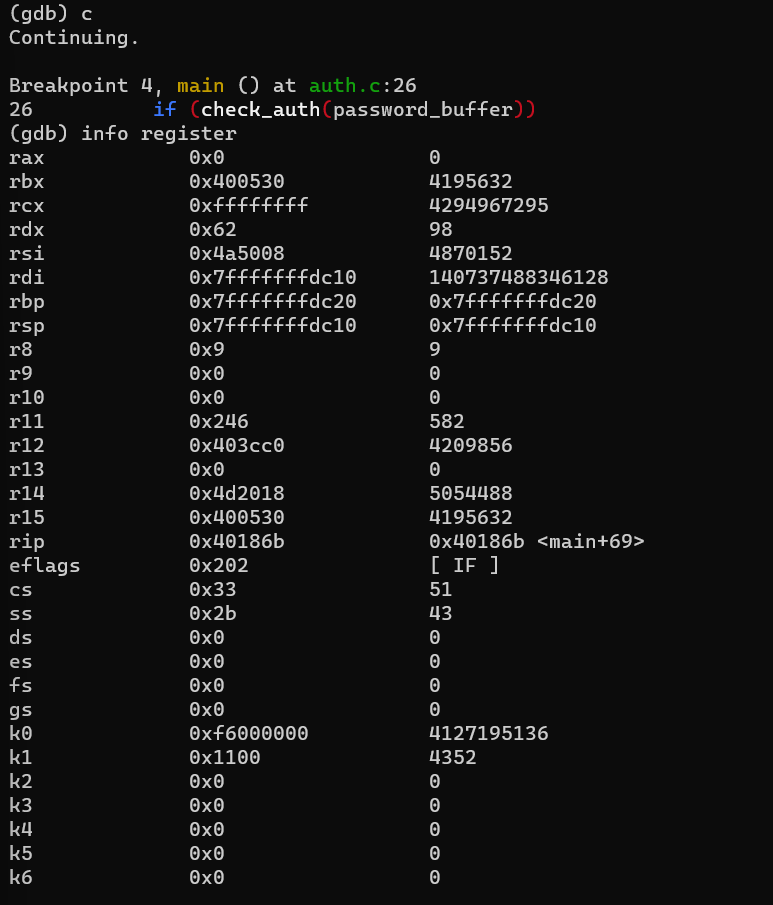


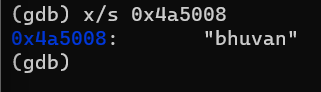


Print the stack content through x command.

$rax stores the values from user\_input password

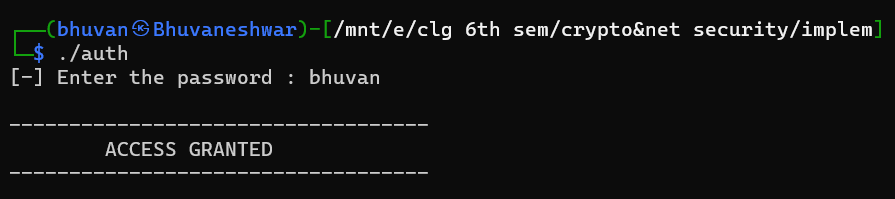
Continue the program





Finally we crack the password through $rbp address.

Run again to program using “bhuvan” as user input



Finally we cracked the password using gdb command.

1. **USING STRING COMMAND**

A string is any sequence of 4 or more printable characters that end with a new line or null characters. Only if developer mentioned password in the code can see via strings in terminal, otherwise we can’t see the password.

