**CS6008 Cryptography and Network Security.**

**Assignment No.1**

**Name:** G Vignesh **Reg No:**2018103078

**Date:** 4.4.2022

**Module-1:** Fundamental.

**External Learning-** Finding Passwords in executables using GDB.

**Aim:**

To check and find the passwords saved in the executable files of the c program by analysing the stack arguments of the string compare function which checks our input against saved password during execution of the program by disassembling the executable using GNU debugger (GDB).

**Tools-used:**

1.Linux Terminal

2.GCC compiler.

3.GNU Debugger.

4.Online Hexadecimal to ASCII converter.

**Description:**

Passwords saved in the executables are Vulnerable to attack by hackers, even if passwords are saved in a file and extracted during runtime to compare against user given passwords are unsafe. Passwords must be protected either by encrypting it or at least save them in a database and compare with it using SQL queries.

In this assignment we are going to use the vulnerability and find the password in the executable file by using disassembler function of GDB. We are given the only access of the executable file of the C program which extracts password and check it against the user input and says whether the password is correct or else wrong.

**Input:** WrongPassword given by the user.

**Output:** Correct Password in Hexadecimal form in Stack of Strncmp function.

**The C program:**

**Extractpwd.c**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

void main(){

    char pwd[25];

    FILE \*file=fopen("pwdfile.txt","r");

    if(file==NULL){

        printf("error in opening file");

    }

    while(fscanf(file,"%s",pwd)!=EOF);

    fclose(file);

    char userpwd[25];

    printf("Enter the password to login:");

    scanf("%s",userpwd);

    printf("The password that u have entered is:%s\n",userpwd);

    int cmp=strncmp(userpwd,pwd,sizeof(pwd)-1);

    if(cmp==0){

        printf("password entered is correct\n");

    }

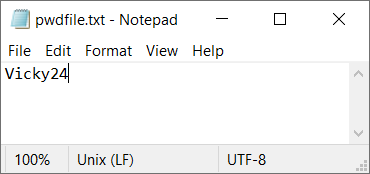
    else{

        printf("\nwrong password!!!\n");

    }

}

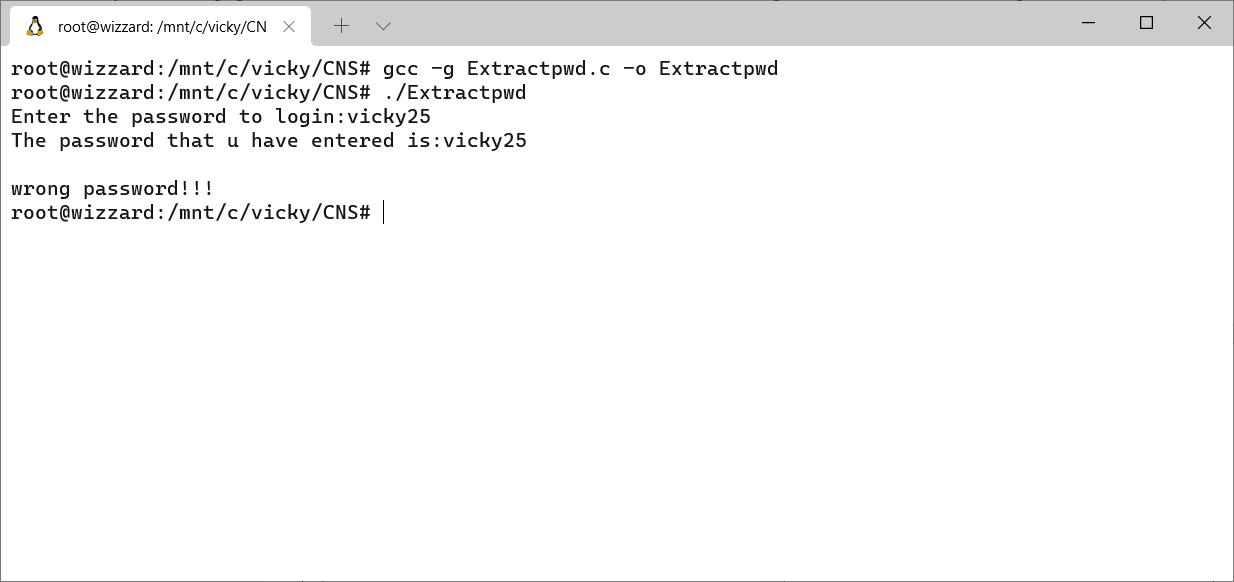
**The Password File:**



**Execution:**

**Step-1:**

The program is compiled in GDB mode using -g command and the executable file is Extractpwd is executed on giving wrong password it prompts as wrong.



**Step-2:**

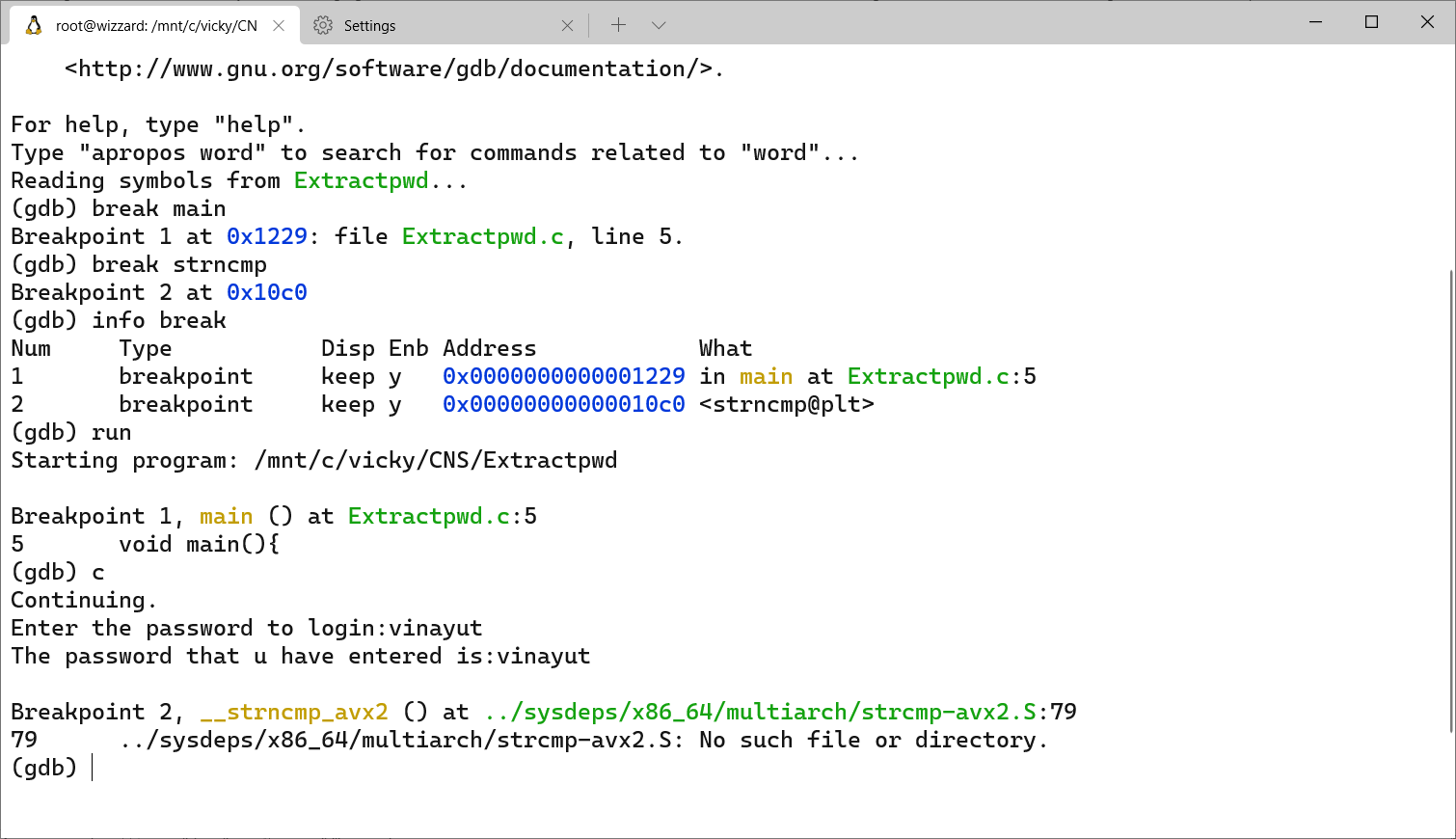
We have entered the GDB mode of the executable file and placed breakpoints in the main function and the strncmp functions. Executing the GDB until main function , where it prompts for password the any unknown wrong password is given. Then run the program till next breakpoint that is **Strncmp.**

The wrong password that we have entered is **vinayut.**

**Strncmp function:**

**int strncmp(const char \*s1, const char \*s2, size\_t n);**

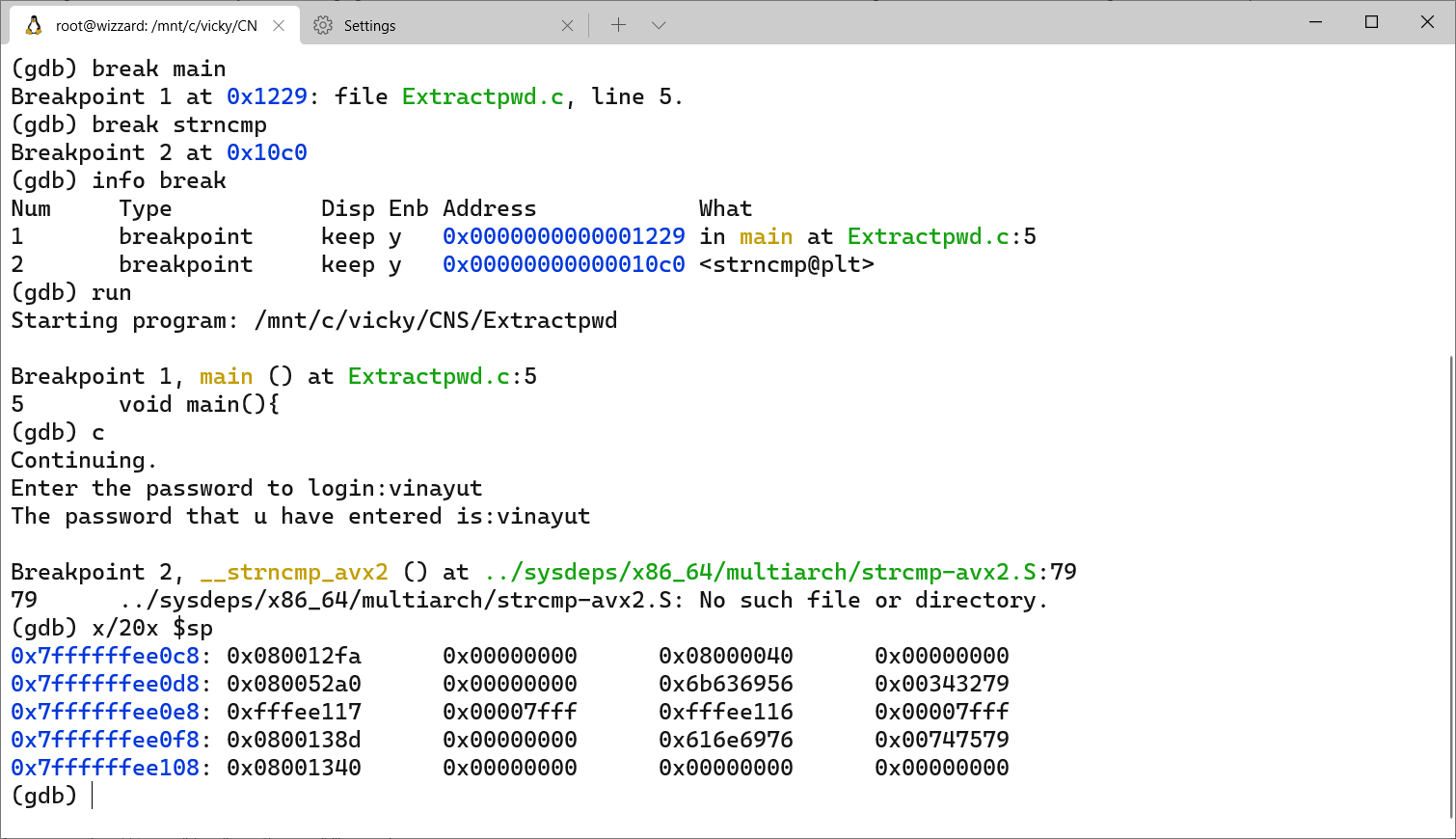
This function takes 3 values as arguments String s1 and String s2 which are to be compared and the size or length to be compared. It returns an integer less than, equal to, or greater than zero if s1 is found, respectively, to be less than, to match, or be greater than s2.



**Step-3:**

Since we have placed a break point in strncmp function, now we are going to open the stack frame of the strncmp function. We are printing the values from top of the stack.

In stack top portion contains the arguments passed , so one of the argument is the wrong password entered and the other is Corrected password stored in the executable.

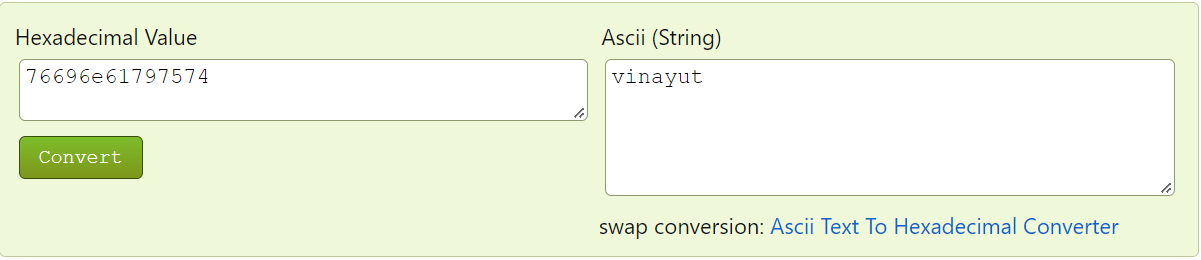


In stack values are in reverse order in hexadecimal form so we need to interpret in reverse order.

**In the fourth address we have :**

0x7ffffffee0f8: 0x0800138d 0x00000000 0x616e6976 0x00747579

76696e61797574

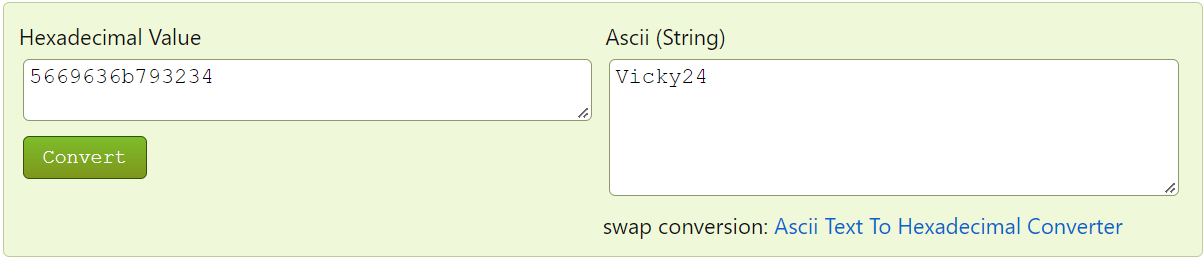


Which the wrong password given by us.

**In the second address field we have:**

0x7ffffffee0d8: 0x080052a0 0x00000000 0x6b636956 0x00343279

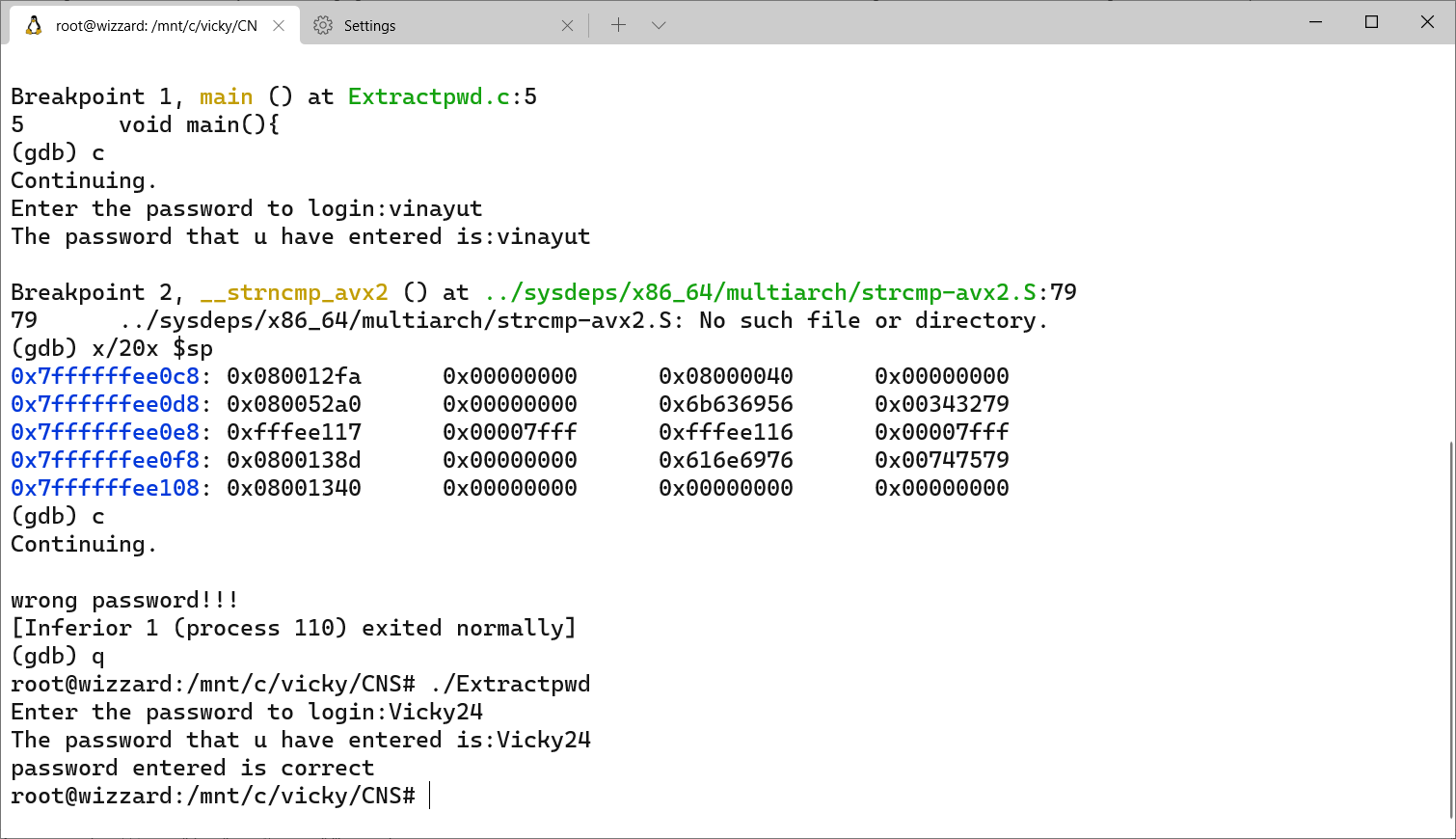
5669636b793234



Which is the correct password used by the strncmp function to compare with our given password.

**Step:4**

Now we are going to check the extracted password in the executable file and find whether its correct or not. We continue to complete the complete the program in GDB which was breaked at strncmp function and the exited GDB mode and ./Extractpwd was executed with obtained password.



The password **Vicky24** is correct.

**Conclusion:**

Suppose the reference password is stored somewhere in the program and isn't encoded in some artful manner unfortunately, then it can be found by simply looking at the binary code of the Strncmp functions stack frame pointer in which shows hexadecimal value of its arguments through which we find the password.

From this assignment we get to know that saving passwords in executable is vulnerable to attack by the hackers if they get access to the executable file, so it is advised not to save passwords in executable files rather encode it or store in databases.