PES UNIVERSITY EC CAMPUS

SECURE PROGRAMMING WITH C

Project synopsis

MEMBER DETAILS :

Shreyas Ganesh(PES2UG19CS387)
S Preetha(PES2UG19CS343)
Rohan George(PES2UG19CS910)

Problem statement:

To develop a C program to implement a library management system which will include all the functionalities required for daily operations.

Compilers and static analysis tools used:

Compiler: **GCC C** (Version 10.2)

The GNU Compiler Collection is an optimizing compiler produced by the GNU Project supporting various programming languages, hardware architectures and operating systems.

Static analysis tool: **Splint 3.1.2**

Splint, short for Secure Programming Lint, is a programming **tool** for statically checking C programs for security vulnerabilities and coding mistakes.

Data Structures and data types used:

●Array of structures- to store the personal information and details of each member of the library

●Character array- to store the names of members/to story the book and member details

●Long int- to store the customer's phone number/member unique ID

●Integer array- To store the due date and borrowed date for each book/member

●Double- To store the penalty amount if any or the borrowing charges

Possible vulnerabilities we will deal with:

● gets() -To limit the possibilities of entering information which does not follow standard format of the given entries.

> The **fgets**() function reads at most one less than the number of characters specified by size from the given stream and stores them in the string str.
>
> The **gets**() function is equivalent to **fgets**() with an infinite size and a stream of stdin, except that the newline character (if any) is not stored in the string.

●printf() and Uncontrolled format string

A malicious user may use the %s and %x format tokens, among others, to print data from the call stack or possibly other locations in memory.

One may also write arbitrary data to arbitrary locations using the %n format token, which commands printf() and similar functions to write the number of bytes formatted to an address stored on the stack.

- strcpy

  The strcpy built-in function does not check buffer lengths and may very well overwrite memory zone contiguous to the intended destination. In fact, the whole family of functions is similarly vulnerable: strcpy, strcat and strcmp.

- File opening vulnerabilities
  Identify and mitigate potential file opening errors and the possible attacks that could arise from using these functions.

Functions used:

- main()-Main function will contain the switch case for the login and signup function.
- signup()-Signup function to register a new user.
- login()-login flow for an existing user.
    - borrow()-flow to borrow a book from the existing inventory.User will check for availability of book,if book is available,the due date and borrowed date is set.Post this,the control is returned to the main driver switch case.
    - return()-User can return a book that has been lent using this flow.Checks if the due date is lesser than the return date.If the condition is satisfied,penalty is assigned to the user.The book is added to the inventory of the library.Control is returned back to the switch case.
- exit()-exits from the main program and this gives the user the ability to close the program at any given point in time when the session is complete

Data flow diagram: