

PES UNIVERSITY EC CAMPUS

SECURE PROGRAMMING

WITH C

# MEMBER DETAILS :

*Neelesh S PES2UG19CS251 D*

*Ritwik Shivam PES2UG19CS333 E*

*Soumya Shridhar Hegde PES2UG19CS401 F*

Problem statement:

*To develop a C code providing hotel room booking service facilities, and dealing with the several possible vulnerabilities that could exploit the application.*

Compilers and static analysis tools used:

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

*The****GNU Compiler Collection****(****GCC****) is an optimizing compiler supporting various languages, hardware architectures and operating systems. GCC C is a free software distributed by Free Software Foundation(FSF).*

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. Splint is free software released under the terms of the GNU General Public License****.***

Data Structures and data types used:

* Array of structures- *to store the personal information and details of each customer.*
* Character array- *to store the names of customers*
* Long int*- to store the customer’s phone number*
* Integer array- *to store the room numbers of the customer which are randomly generated*
* Double- *to store and display the final bill amount*

Possible vulnerabilities we will deal with:

* gets() -To limit the input size in case of phone numbers

*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()andUncontrolled 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.*

# Random number generator attack- To generate room numbers

*The security of****cryptographic****systems depends on some secret data that is known to authorized persons but unknown and unpredictable to others. To achieve this unpredictability, some****randomisation****is typically employed. Modern****cryptographic protocols****often require frequent generation of random quantities. Cryptographic attacks that subvert or exploit weaknesses in this process are known as****random number generator attacks****.*

*x=rand();*

*This code generated bad random numbers because when you call rand() before a seed has been established with srand(), it uses the value 1 as a default seed. Anyone else on the same machine with the same compiler who calls rand() with a seed of 1 will get the same random number as you just did.*

*srandom (time (0));*

*x=random();*

*This code uses the BSD random() and srandom() functions, which generate much better random numbers than their ANSI C predecessors. However, this code still uses time() to generate the seed number. A much better source for random numbers on BSD and Linux systems is the /dev/random device.*

*Random number vulnerabilities are of interest to hackers when they can be utilized to determine input values to****cryptographic****functions. This can be utilized in cryptanalysis.*

*Improper use of the function calls rand() and random() are the normal causes of random number vulnerabilities.*

*Functions used:*

*Main()- consists of the switch case for the menu driven program*

*Login() – It creates a new* ***user id*** *by accepting user details such as* ***Name*** *and* ***Contact details****.*

*Signin() – It displays the user details along with booked rooms if-any and points to two other sub-functions:*

* *Book\_rooms() – It allows the user to book rooms of different types according to user needs for example:*

*Single(1 person), Double(2 people), Quad(4 people), Suite.*

*(Room number allocation is done by any random number generator function )*

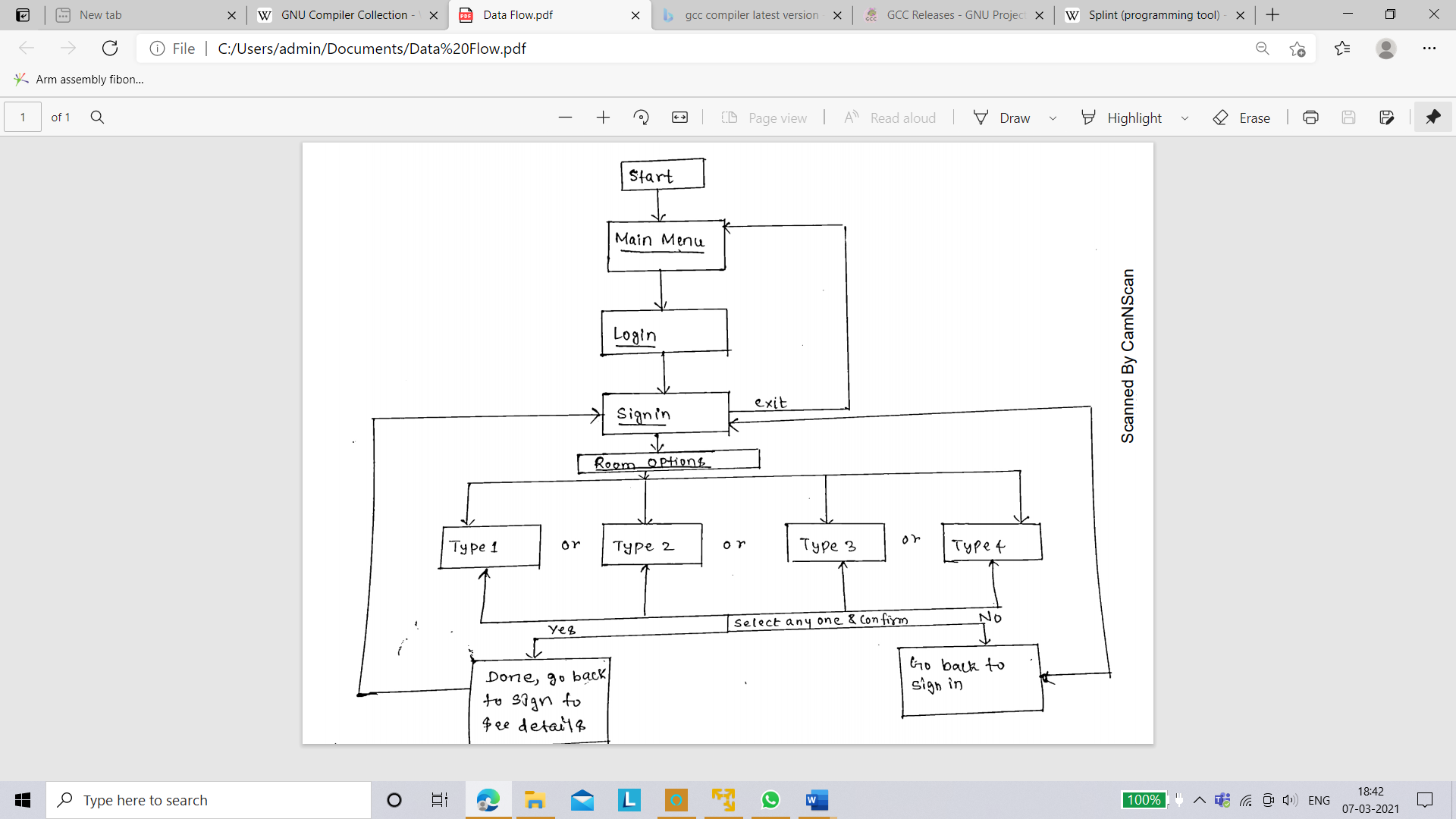
*After selecting suitable rooms, if user wants to reserve the room/s, Book\_rooms() points to Signin() which displays no. of booked rooms, the total fare and an option to quit from the process.*

*If user wants to repeat the process without reserving the rooms, Book\_rooms() points to itself which allows the user to repeat the process.*

* *Exit()- Allows the user to quit from the process by pointing to Main() with switch cases.*

*Terminate()- Terminates the program.*

***DATA FLOW DIAGRAM:***



*\*\*\*\*\*\**