Reo Matsuda

825001809

CSCE 313-508

PA0 Report

Steps taken to debug:

1. Compile program using g++.  
   $ g++ buggy.cpp -g -o buggy
2. Run program to check any compile errors.  
   $ ./buggy
3. Add correct include statements.
4. Make node class variables public.
5. Correct pointer value accessing method.  
   mylist[i].val 🡨 NO  
   mylist[i]->val 🡨 YES
6. Compile program again.
7. Run program.
8. Run program using gdb.  
   $ gdb buggy  
   $ gdb) run
9. Find line number that is causing segfault.
10. Add a break point.  
    $ gdb) break 17
11. Run program in gdb.
12. Print content of array.  
    $ gdb) print mylist  
    $ $1 = std::vector of length 3, capacity 3 = {0x0, 0x0, 0x0}
13. Fix array elements not being initialized properly.  
    mylist[i] = new node();
14. Run program using gdb.
15. Find line number that is causing segfault.
16. Add a break point.
17. Run program in gdb.
18. Step in gdb until segfault reached.  
    $ gdb) step
19. Fix bugged linkedlist creation.  
      
    Explanation for segfault:  
     When the linked list was created from the vector, the last node’s next pointer was assigned to an element of the vector that was outside the index. Because of this the last node’s next pointer was assigned to a garbage value, for me this was 0x21.
20. Compile program again but adding -fsanitize flag, this uses AddressSanitizer.
21. Run program.

$ ==4388==ERROR: LeakSanitizer: detected memory leaks

1. Make sure all allocated node pointers are deleted at the end of program.  
    Use an iterator to iterate through the vector and delete allocated node pointer.

Debugging within IDE, Visual Studio Code

A screenshot of a computer

Description automatically generated