Yifei Liang
727002177

Programing Assignment 0
Report

GDB version:

Step 1:

Adding #include <vector> and using namespace std at the header, and set the node member to public

Step 2:

Correct the statement by changing . to ->

Step 3:

Compile with following statement:

g++ buggy.cpp -g
gdb a.out
run

found a segmentation fault at line 15
using **break 15** and **run**
and **print mylist[i]**
found out that mylist[i] is pointing to a nullptr
fixed the issue by using:

mylist[i] = new node()

Compiled and ran again
found another error at sum function
the issue was at the second part of the create_LL:

the "next" var of the tail node will be assign to a random location

there for I needed to change **i < node_num. to i < node_num** – 1

step 4:

delete all the node in the vector

AddressSanitizer:

Step 1:

Adding #include <vector> and using namespace std at the header, and set the node member to public

Step 2:

Correct the statement by changing . to ->

Step 3:

Compile and run with following statement:

g++ buggy.cpp -fsanitize=address -g

./a.out

Found a **SEGV on** #0 0xaaaacc6fe5f7 in create_LL(std::vector<node*, std::allocator<node*> >&, int) /home/ubuntu/service/buggy.cpp:63

Fixed it by mylist[i] = new node()

Compiled and run again

Found a **heap-buffer-overflow on** #0 0xaaaabd6ed7c3 in create_LL(std::vector<node*, std::allocator<node*> >&, int) /home/ubuntu/service/buggy.cpp:69

the issue was at the second part of the create_LL:

the "next" var of the tail node will be assign to a random location there for I needed to change i < node_num. to i < node_num – 1

step 4:

compiled and ran again

Found a

detected memory leaks

fixed it by deleting all node in the vector

Conclusion:

After finishing this programming assignment, one thing I leanred is that the Address Sanitizer is really good at finding the potential problem. For example, when the running the program with gdb the issue at the second part of the create_LL didn't show up until the function sum function. However, when I ran with Adress Sanitizer it showed up more quickly than using GDB