Module.3

Memory Corruption in Dynamic Arrays

Description: Track memory corruption in dynamic arrays by reversing execution.

Debugging Tasks: 1.

Compile: 2. Use GDB: o Set a breakpoint before the loop. o Step into the loop and observe memory writes. o Reverse-step to identify where the out-of-bounds write occurs.

3. Fix the loop bounds.

**Find error track\_memory\_corruption:-----**

track\_memory\_corruption.c

#include <stdio.h>

#include <string.h>

void vulnerableFunction(char \*str) {

    char buffer[10];

    // The following line introduces a buffer overflow vulnerability.

    strcpy(buffer, str);

}

int main() {

    // Input string larger than the allocated buffer size.

    char largeInput[] = "TooLongInputDataExceedingBuffer";

    vulnerableFunction(largeInput);

    return 0;

}

**Steps to debug to find track\_memory\_corruption using gdb**

GDB Commands

Compile the Code with Debugging Symbols:

gcc -g -o track\_memory\_corruption track\_memory\_corruption.c

Start GDB:

gdb ./track\_memory\_corruption

break vulnerableFunction

Run

step

record

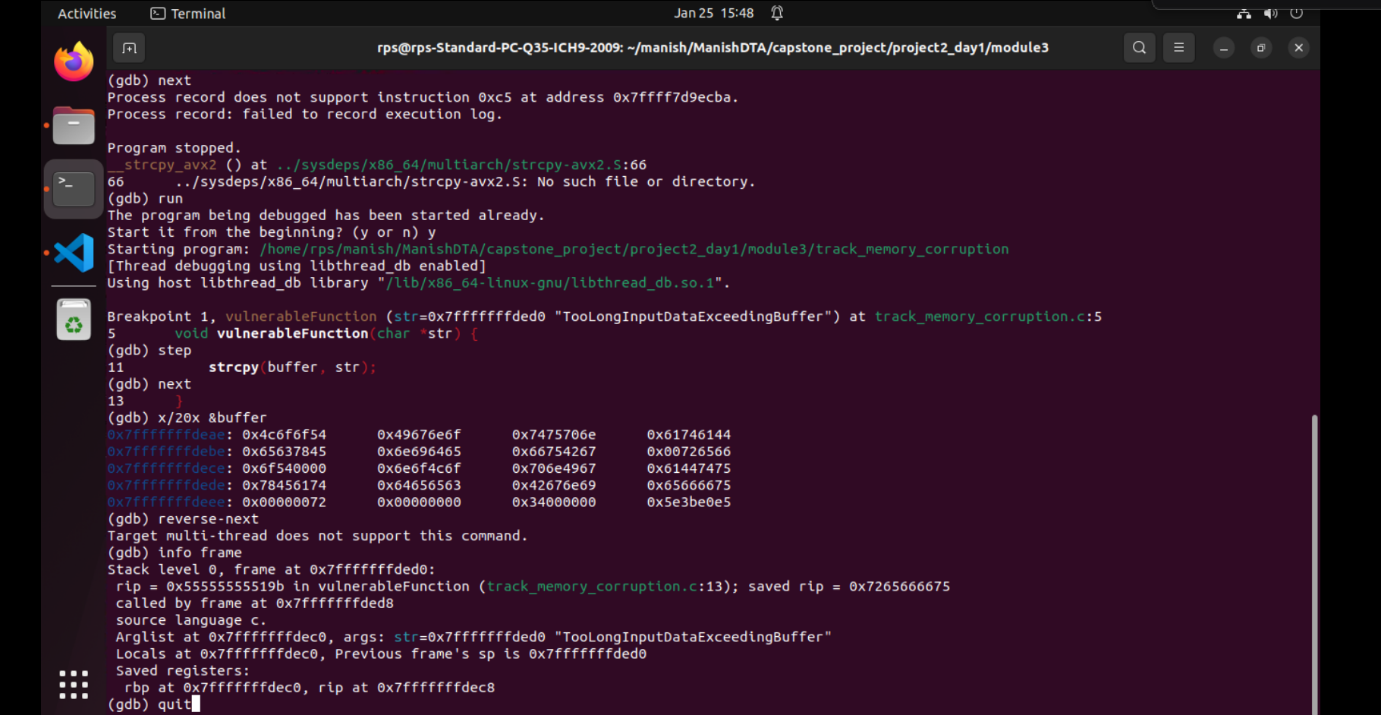
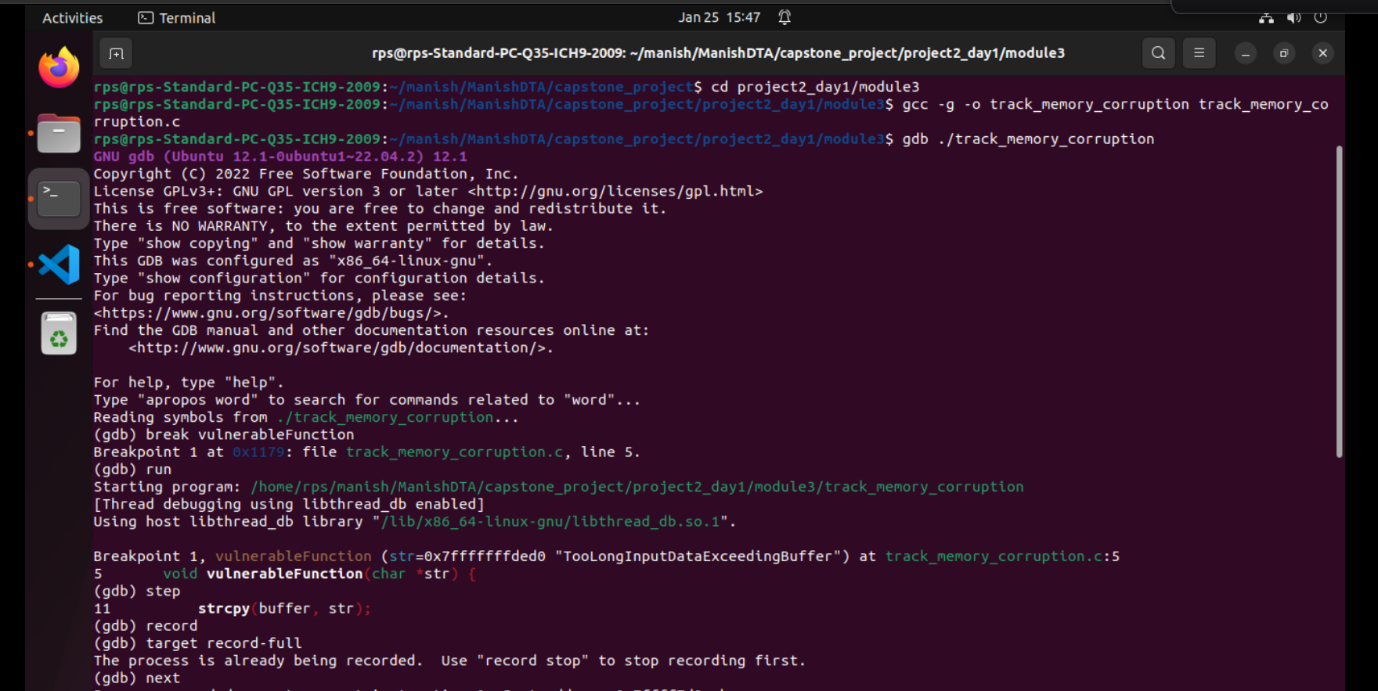
next

x/20x &buffer

reverse-next

reverse-step

info frame



**Fix track\_memory\_corruption:-----**

#include <stdio.h>

#include <string.h>

void safeFunction(char \*str) {

    char buffer[10];

    // Use strncpy to prevent buffer overflow.

    strncpy(buffer, str, sizeof(buffer) - 1);

    buffer[sizeof(buffer) - 1] = '\0'; // Ensure null termination.

    printf("Buffer content: %s\n", buffer);

}

int main() {

    // Input string larger than the allocated buffer size.

    char largeInput[] = "TooLongInputDataExceedingBuffer";

    safeFunction(largeInput);

    return 0;

}

**Steps to debug to verify to check the track\_memory\_corruption using gdb is fix or not**

Steps to Verify the Fix

gcc -g -o fix\_track\_memory\_corruption fix\_track\_memory\_corruption.c

gdb ./fix\_track\_memory\_corruption

break safeFunction

break strncpy

run

x/20x &buffer

next

print buffer

x/40x &buffer - 10

continue

