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

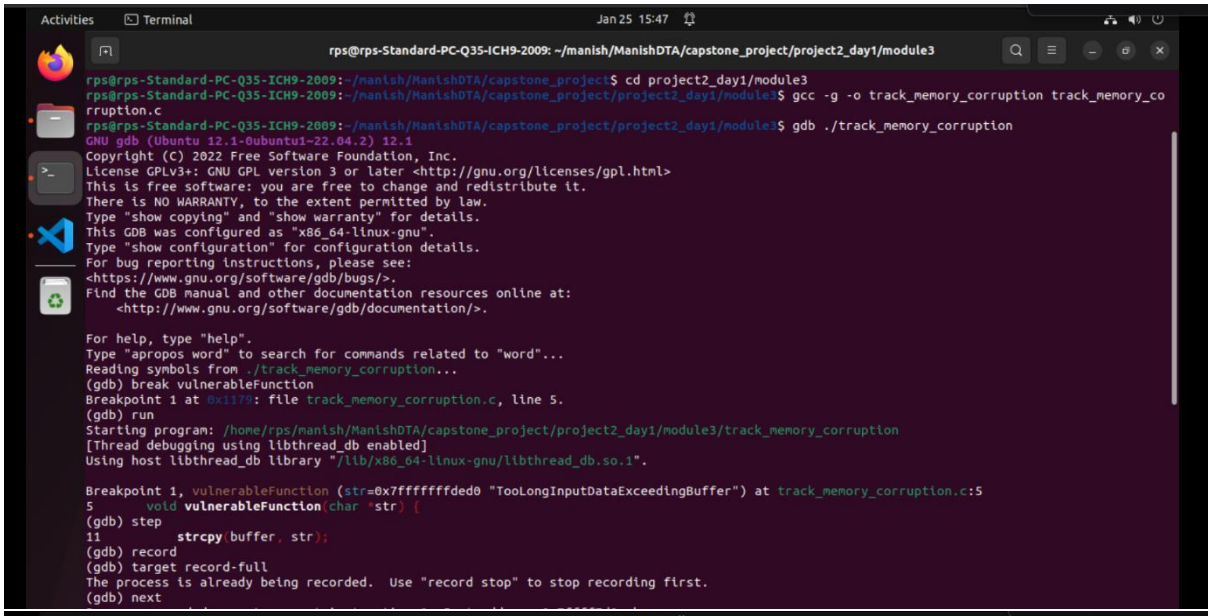
next

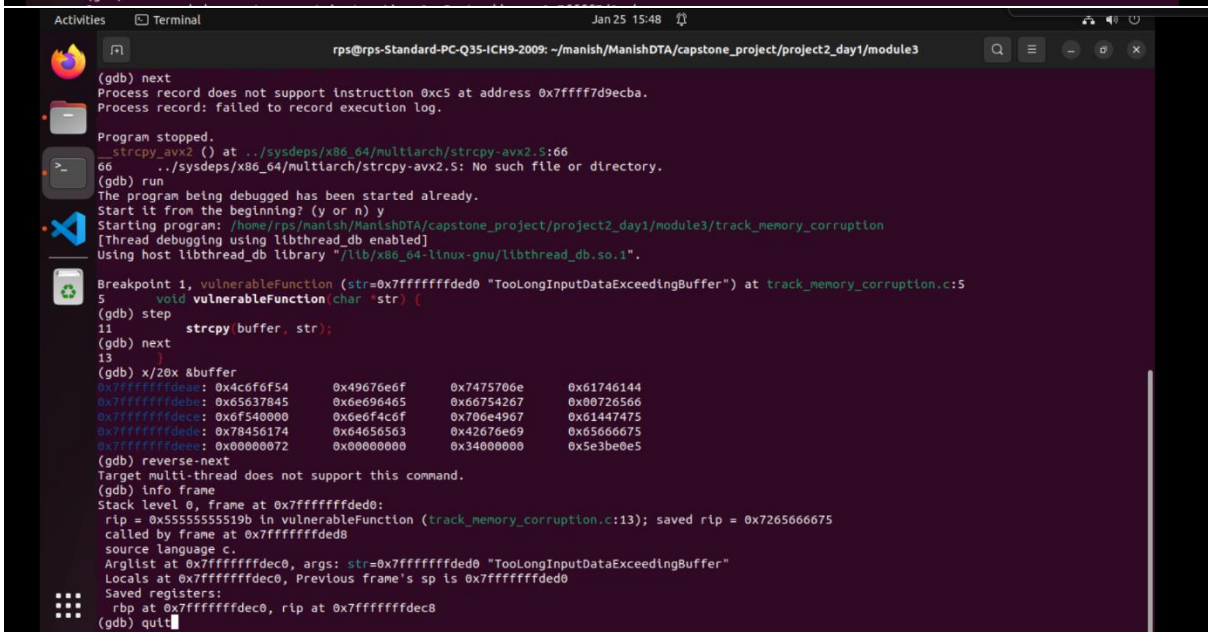
x/20x &buffer

reverse-next

reverse-step

info frame





Fix track_memory_corruption:-----

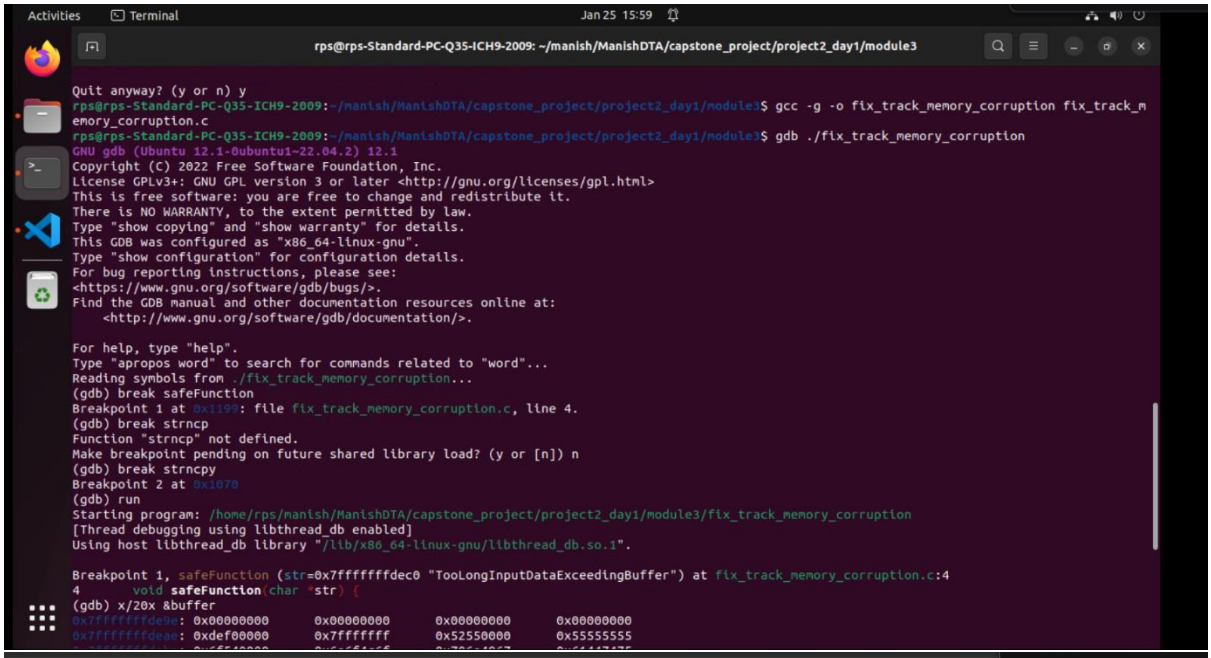
```
#include <stdio.h>
#include <string.h>

void safeFunction(char *str) {
```

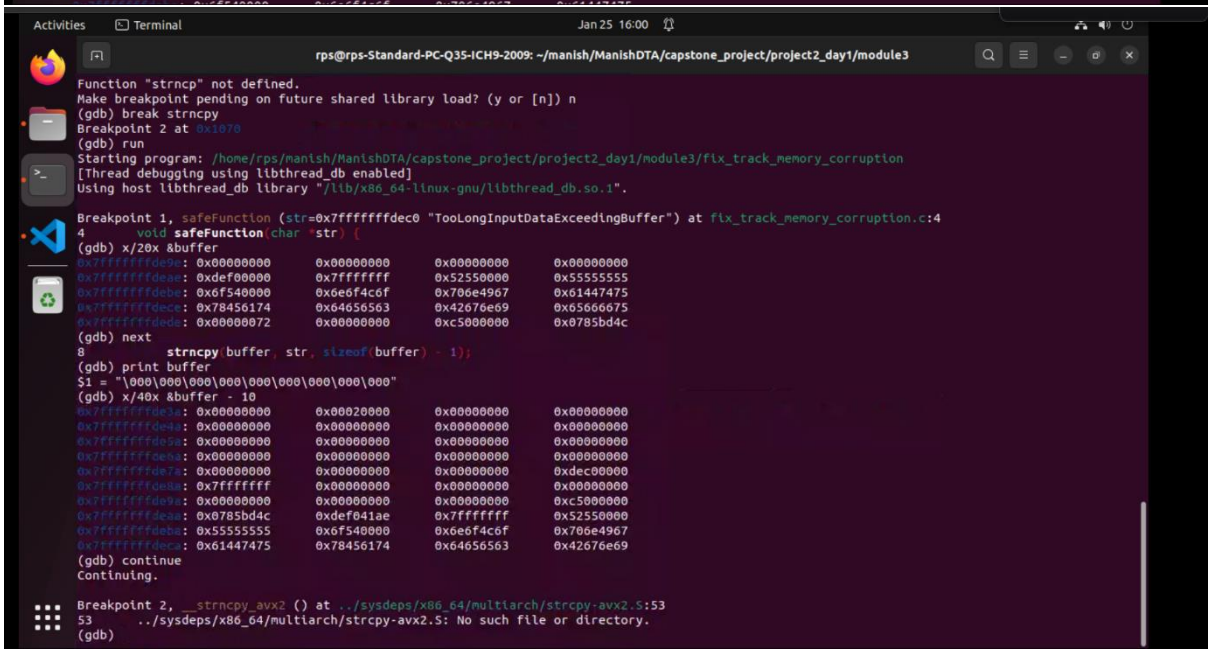
```
    char buffer[10];

    // 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
```


Activities
Terminal
Jan 25 15:59
rps@rps-Standard-PC-Q35-ICH9-2009: ~/manish/ManishDTA/capstone_project/project2_day1/module3
Quit anyway? (y or n) y
rps@rps-Standard-PC-Q35-ICH9-2009:~/manish/ManishDTA/capstone_project/project2_day1/module3$ gcc -g -o fix_track_memory_corruption fix_track_m
emory_corruption.c
rps@rps-Standard-PC-Q35-ICH9-2009:~/manish/ManishDTA/capstone_project/project2_day1/module3$ gdb ./fix_track_memory_corruption
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04.2) 12.1
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./fix_track_memory_corruption...
(gdb) break safeFunction
Breakpoint 1 at 0x1198: file fix_track_memory_corruption.c, line 4.
(gdb) break strncp
Function "strncp" not defined.
Make breakpoint pending on future shared library load? (y or [n]) n
(gdb) break strncpy
Breakpoint 2 at 0x1070
(gdb) run
Starting program: /home/rps/manish/ManishDTA/capstone_project/project2_day1/module3/fix_track_memory_corruption
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Breakpoint 1, safeFunction (str=0x7fffffffdec0 "TooLongInputDataExceedingBuffer") at fix_track_memory_corruption.c:4
4 void safeFunction(char *str) {
(gdb) x/20x &buffer
0x7fffffffde9e: 0x00000000 0x00000000 0x00000000 0x00000000
0x7fffffffdeae: 0xdef00000 0x7fffffff 0x52550000 0x55555555


Activities
Terminal
Jan 25 16:00
rps@rps-Standard-PC-Q35-ICH9-2009: ~/manish/ManishDTA/capstone_project/project2_day1/module3
Function "strncp" not defined.
Make breakpoint pending on future shared library load? (y or [n]) n
(gdb) break strncpy
Breakpoint 2 at 0x1070
(gdb) run
Starting program: /home/rps/manish/ManishDTA/capstone_project/project2_day1/module3/fix_track_memory_corruption
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Breakpoint 1, safeFunction (str=0x7fffffffdec0 "TooLongInputDataExceedingBuffer") at fix_track_memory_corruption.c:4
4 void safeFunction(char *str) {
(gdb) x/20x &buffer
0x7fffffffde9e: 0x00000000 0x00000000 0x00000000 0x00000000
0x7fffffffdeae: 0xdef00000 0x7fffffff 0x52550000 0x55555555
0x7fffffffdebe: 0x6f540000 0x6e6f4c6f 0x706e4967 0x61447475
0x7fffffffdece: 0x78456174 0x64656563 0x42676e69 0x65666675
0x7fffffffdede: 0x00000072 0x00000000 0xc5000000 0x0785bd4c
(gdb) next
8 strncpy(buffer, str, sizeof(buffer) - 1);
(gdb) print buffer
$1 = "\000\000\000\000\000\000\000\000\000\000"
(gdb) x/40x &buffer - 10
0x7fffffffde5e: 0x00000000 0x00020000 0x00000000 0x00000000
0x7fffffffde6e: 0x00000000 0x00000000 0x00000000 0x00000000
0x7fffffffde7e: 0x00000000 0x00000000 0x00000000 0x00000000
0x7fffffffde8e: 0x00000000 0x00000000 0x00000000 0x00000000
0x7fffffffde9e: 0x7fffffff 0x00000000 0x00000000 0x00000000
0x7fffffffdeae: 0x0785bd4c 0xdef041ae 0x7fffffff 0x52550000
0x7fffffffdebe: 0x55555555 0x6f540000 0x6e6f4c6f 0x706e4967
0x7fffffffdece: 0x61447475 0x78456174 0x64656563 0x42676e69
(gdb) continue
Continuing.
Breakpoint 2, __strncpy_avx2 () at ../sysdeps/x86_64/multiarch/strcpy-avx2.S:53
53 ../sysdeps/x86_64/multiarch/strcpy-avx2.S: No such file or directory.
(gdb)