Debug Memory management issues in C – Elon Nguyen

18 Tháng Tám 2023 11:06 SA

1. Debugging the memory management issues in C program using **electric fence** malloc debugger and GDB:

Electric Fence helps you detect two common programming bugs:

1. software that overruns the boundaries of a malloc() memory allocation
2. software that touches a memory allocation that has been released by free()

Below is the example code I have written to demonstrate the usage of **electric-fence malloc debugger** in conjunction with GDB and also provided the instruction to use.

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

void overrun_boundary()
{
        char *ptr=NULL;
        ptr = malloc(sizeof(char)*5);
        strcpy(ptr,"hello linux world");
        printf("%s\n",ptr);
}

void access_free()
{
        int *ptr=NULL;
        ptr = malloc(sizeof(int)*4);
        ptr[0]=100;
        free(ptr);
        printf("ptr[0]=%d\n",ptr[0]);
}

int main()
{
        overrun_boundary();
        //access_free();
        return EXIT_SUCCESS;
}

================================================================================================================

Notes:
Install by using the command:
# sudo apt-get install electric-fence

Compile and Link the electric fence by using the command:
# gcc <program_name.c> -o <program_name> -g -lefence

Run the executable:
./program_name

Electric Fence 2.2 Copyright (C) 1987-1999 Bruce Perens <bruce@perens.com>
Segmentation fault (core dumped)

Inspect using GDB
#gdb ./program_name
#run (need to type in GDB console)

GDB analysis:

GNU gdb (Ubuntu 8.1.1-0ubuntu1) 8.1.1
Copyright (C) 2018 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:
<http://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 ./lefence_demo...done.
(gdb) r
Starting program: /home/shashank/electric_fence/lefence_demo
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

  Electric Fence 2.2 Copyright (C) 1987-1999 Bruce Perens <bruce@perens.com>

Program received signal SIGSEGV, Segmentation fault.
0x0000555555554823 in overrun_boundary () at lefence_demo.c:9
9       strcpy(ptr,"hello linux world");
```

2. Using **valgrind** tool to detect the memory management issues in C:

   valgrind is used for debugging and profiling the Linux programs. By using **valgirnd** tool we can automatically detect the many memory management and threading bugs thus making your program more stable.

   To demonstrate the usage of **valgrind** tool I have written a c program which performs the operations below:
   1. usage of un-initialized memory
   2. memory leak(not freeing the allocated memory dynamically)
   3. used freed memory
   4. overshooting the memory (access the memory beyond the allocated range)
   5. not freeing the memory allocated by using realloc()
   6. using un-initialized variable
   7. double freeing of the memory

   Install the valgrind tool in system by using the below command:
   # sudo apt-get install valgrind

   verify the installation using the below command:
   # valgrind --version

   compile the program using below command :
   # gcc <program_name.c> -g std=c11 -lm -o <program_name or executable_name>

   Run the program with valgrind tool:
   # valgrind --track-origins=yes --leak-check=full ./executable_name

   It is clearly evident that from the below result, valgrind tool detected double free'd of the memory and also it shows the function and line number where exact the issue is.

   NOTE: uncomment the functions in the code to check the corresponding behavior using valgrind tool.