Report Case 1

Pamela Salazar

May 2023

1 Introduction

Valgrind is an instrumentation framework for building dynamic analysis tools. It comes with a set of tools each of which performs some kind of debugging, profiling, or similar task that helps you improve your programs. Valgrind's architecture is modular, so new tools can be created easily and without disturbing the existing structure. Memcheck is a memory error detector in Valgrind. It helps you make your programs, particularly those written in C and C++, more correct. [1].

For example the C code of the case 1 allocate 40 bytes in memory to the pointer a∗. However in the for loop 44 bytes are required, that cause a memory error.

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

int main(int argc, char** argv){
  int i;
  int *a = malloc(sizeof(int) * 10);
  if (!a) return -1; /*malloc failed*/
  for (i = 0; i < 11; i++){
    a[i] = i;
  }
  free(a);
  return 0;
}
```

The memory error is reported by the Valgrind in the next way:

```
==11790== Memcheck, a memory error detector
==11790== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==11790== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==11790== Command: ./case1
```

```
==11790==
==11790== Invalid write of size 4
==11790==    at 0x1091B8: main (case1.c:9)
==11790==  Address 0x4a9f068 is 0 bytes after a block of size 40 alloc'd
==11790==    at 0x4848899: malloc (in /usr/libexec/valgrind/vgpreload_memcheck-amd64-linux.s
==11790==    by 0x109185: main (case1.c:6)
==11790==
==11790==
==11790== HEAP SUMMARY:
==11790==     in use at exit: 0 bytes in 0 blocks
==11790==   total heap usage: 1 allocs, 1 frees, 40 bytes allocated
==11790==
==11790== All heap blocks were freed -- no leaks are possible
==11790==
==11790== For lists of detected and suppressed errors, rerun with: -s
```

References

[1] Valgrind Manual User. `https://valgrind.org/docs/manual/manual.html`, 24^{th} may 2023.