Report Case 2

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 2 "a" is partially uninitialized. There is no value initialized to a[9].

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

int main(int argc, char** argv){
  int i;
  int a[10];
  for (i = 0; i < 9; i++)
    a[i] = i;

  for (i = 0; i < 10; i++){
    printf("%d ", a[i]);
  }
  printf("\n");
  return 0;
}
```

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

```
==16297== Memcheck, a memory error detector
==16297== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==16297== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==16297== Command: ./case2
```

```
==16297==
==16297== Conditional jump or move depends on uninitialised value(s)
==16297==    at 0x48EAB56: __vfprintf_internal (vfprintf-internal.c:1516)
==16297==    by 0x48D481E: printf (printf.c:33)
==16297==    by 0x1091F1: main (case2.c:11)
==16297==
==16297== Use of uninitialised value of size 8
==16297==    at 0x48CE33B: _itoa_word (_itoa.c:177)
==16297==    by 0x48E9B3D: __vfprintf_internal (vfprintf-internal.c:1516)
==16297==    by 0x48D481E: printf (printf.c:33)
==16297==    by 0x1091F1: main (case2.c:11)
==16297==
==16297== Conditional jump or move depends on uninitialised value(s)
==16297==    at 0x48CE34C: _itoa_word (_itoa.c:177)
==16297==    by 0x48E9B3D: __vfprintf_internal (vfprintf-internal.c:1516)
==16297==    by 0x48D481E: printf (printf.c:33)
==16297==    by 0x1091F1: main (case2.c:11)
==16297==
==16297== Conditional jump or move depends on uninitialised value(s)
==16297==    at 0x48EA643: __vfprintf_internal (vfprintf-internal.c:1516)
==16297==    by 0x48D481E: printf (printf.c:33)
==16297==    by 0x1091F1: main (case2.c:11)
==16297==
==16297== Conditional jump or move depends on uninitialised value(s)
==16297==    at 0x48E9C85: __vfprintf_internal (vfprintf-internal.c:1516)
==16297==    by 0x48D481E: printf (printf.c:33)
==16297==    by 0x1091F1: main (case2.c:11)
==16297==
0 1 2 3 4 5 6 7 8 0
==16297==
==16297== HEAP SUMMARY:
==16297==     in use at exit: 0 bytes in 0 blocks
==16297==   total heap usage: 1 allocs, 1 frees, 1,024 bytes allocated
==16297==
==16297== All heap blocks were freed -- no leaks are possible
==16297==
==16297== Use --track-origins=yes to see where uninitialised values come from
==16297== For lists of detected and suppressed errors, rerun with: -s
==16297== ERROR SUMMARY: 5 errors from 5 contexts (suppressed: 0 from 0)
```

References

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