XIAMEN UNIVERSITY MALAYSIA
厦門大學马来西亚分校

Debug with Debugger

1. Study the code >> primeProgram-buggy.c
2. Run the program
3. Debug
4. List all the issues/bug found and rectify them one by one

Example:

Instructions:

1. Compile the Code
2. Run debugger

REPORT:

BUG #1	Program received signal SIGINT, Interrupt. isPrime (x=2) at main.c:68 68 if(x % 2 == 0) { it does not identify 2 as prime number
How did you resolve it?	
Issue solved? Yes or No	

BUG #2	Program received signal SIGINT, Interrupt. isPrime (x=2) at main.c:68 68 if(x % 2 == 0) { it does not identify 2 as prime number
How did you resolve it?	
Issue solved? Yes or No	

And so on…

Debug with GDB Debugger

1. vuln.c

a) Compile the program below. Name you file vuln.c.

```
/vuln.c
#include <stdio.h>
#include <string.h>
int main(int argc, char* argv[]) {
    /* [1] */ char buf[256];
    /* [2] */ strcpy(buf,argv[1]);
    /* [3] */ printf("Input:%s\n",buf);
    return 0;
}
```

Use GDB to identify the offending function and code line causing the error. What is the cause of the error? How can the error be rectified?

2. Stack/Heap Overflow

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

int main(void) {
   // stack corruption
  char buf2[16] = "overwriteme";
  char buf1[16];

   strcpy(buf2, "");

   printf("buf1 val:  %s\n", buf1);
   printf("buf2 val:  %s\n", buf2);
   printf("buf1 addr: %p\n", (void *)buf1);
   printf("buf2 addr: %p\n", (void *)buf2);

   // heap corruption
   /*
   char *buf3 = malloc(12 * sizeof(char));
   char *buf4 = malloc(12 * sizeof(char));

    strcpy(buf4, "mywordshere");
    strcpy(buf3, "");

   printf("buf3 val:  %s\n", buf3);
   printf("buf4 val:  %s\n", buf4);
   printf("buf3 addr: %p\n", (void *)buf3);
   printf("buf4 addr: %p\n", (void *)buf4);
   */
}
```

Demonstrate the stack and heap overflow. Capture the screenshots to document these events, and provide an explanation of the processes involved.

Upload your answers in Moodle. Make sure it is in PDF format.