Colby Crutcher

Lab2

Secure Coding

1. (5 points) What type of file is this, and what kind of security does it have?

- An exe file, and just has read permissions for regular users.

2. (5 points) What are the files that make up this binary, and which one contains main?

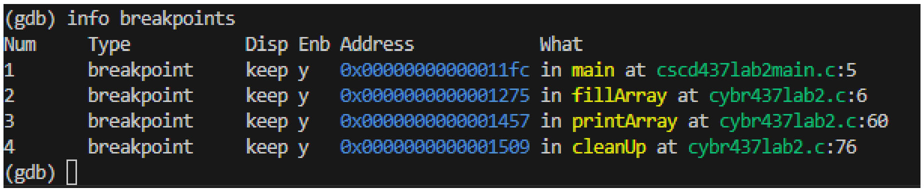
- cscd437main.c, cybr437lab2.c, cybr437.h.

cscd437main.c contains teh main function.

3. (5 points) What are the type(s) and name(s) of parameter(s) being passed to main?

- Types - int, and char pointer
- Names - int argc, and char**argv

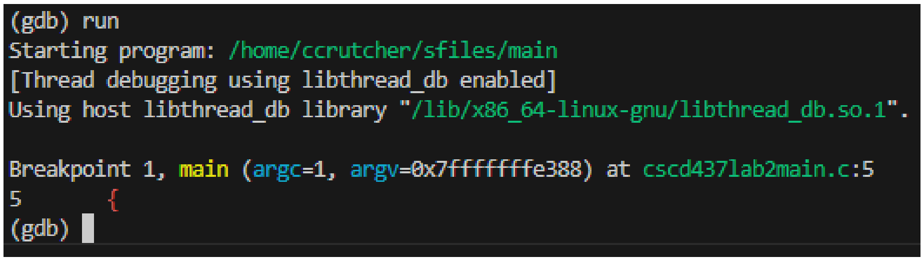
4. (5 points) Set a breakpoint on each function and display the breakpoints.



```
(gdb) info breakpoints
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x00000000000011fc in main at cscd437lab2main.c:5
2       breakpoint     keep y   0x0000000000001275 in fillArray at cybr437lab2.c:6
3       breakpoint     keep y   0x0000000000001457 in printArray at cybr437lab2.c:60
4       breakpoint     keep y   0x0000000000001509 in cleanUp at cybr437lab2.c:76
(gdb) 
```

5. (5 points) Begin running the program. How many arguments are passed to main?

- argc = 1 was passed through main, and that is it.

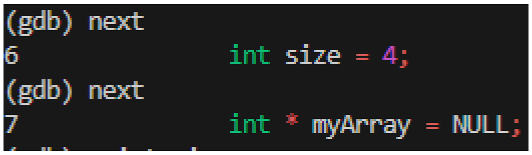


```
(gdb) run
Starting program: /home/ccrutcher/sfiles/main
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main (argc=1, argv=0x7fffffffe388) at cscd437lab2main.c:5
5       {
(gdb) 
```

6. (5 points) Step twice and display the contents of the constant and two variables in main.

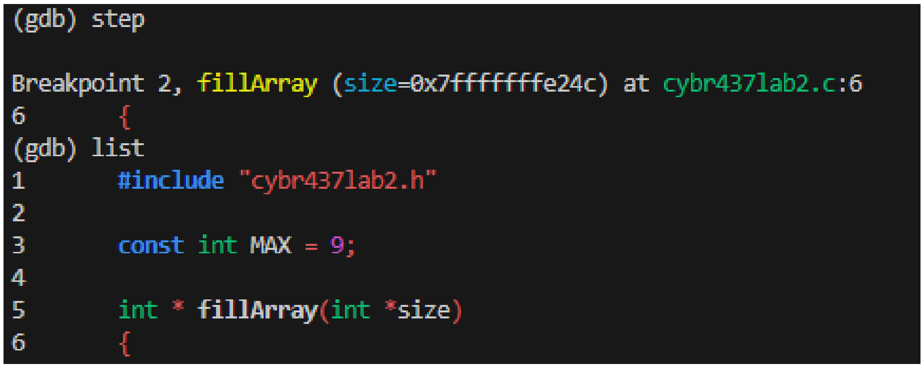
- int size = 4, and the int pointer myArray is null.



```
(gdb) next
6               int size = 4;
(gdb) next
7               int * myArray = NULL;
```

7. (5 points) Step until you enter the first function called in main. What is the type and name of the constant? Where is the constant declared?

- The constant is an int, name “MAX”, and has a value of 9. It is declared globally in cybr437lab2.c



```
(gdb) step

Breakpoint 2, fillArray (size=0x7fffffffe24c) at cybr437lab2.c:6
6       {
(gdb) list
1       #include "cybr437lab2.h"
2
3       const int MAX = 9;
4
5       int * fillArray(int *size)
6       {
```

8. (5 points) Explain the difference between printing var and var[x]?

- Printing var outputs the memory address, and printing var[x]

9. (5 points) Instead of continuing this function, return to main and print the current line.
10. (5 points) Step into the second function called from main. What is the name and starting value of the counting variable?
11. (5 points) Use the disassemble command in GDB to display the assembly code for the function. What does the output show, and how does it correlate with the C source code? Page 2 of 2
12. (5 points) Delete the breakpoint for the first function called in main and disable the breakpoint for the last function called.
13. (5 points) Without starting over, AKA your current location in a function that is not main, print the memory location of the first variable passed to main.
14. (5 points) Print the information on the current running threads. How many threads are running?
15. (5 points) Enable the breakpoint for the third function called by main. What is the type, name, and memory location of the variable passed to it?