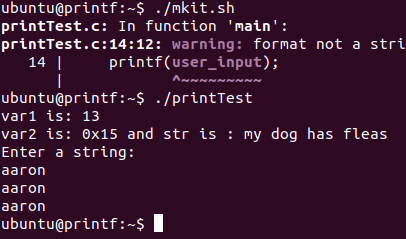
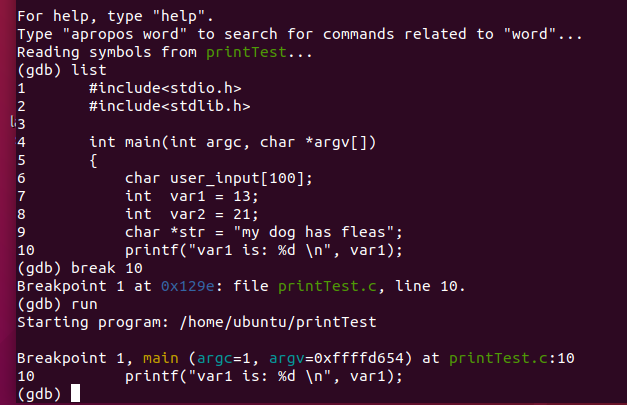
Introduction to printf memory refererences



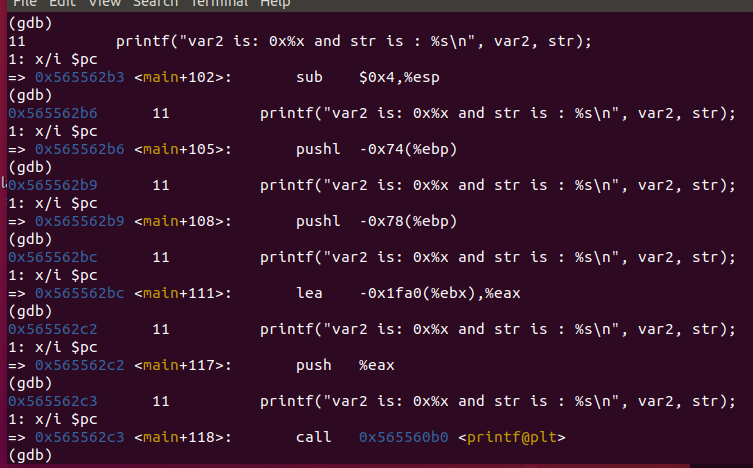
I used “nano printTest.c” to open the file.



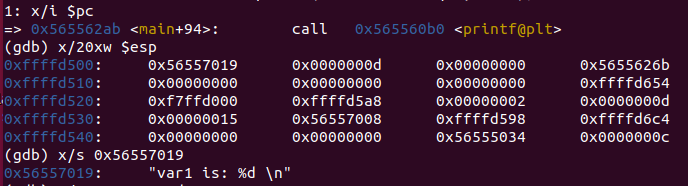
I ran “mkit.sh” to compile the program and “print Test” to run the program. The string I entered was “aaron”.



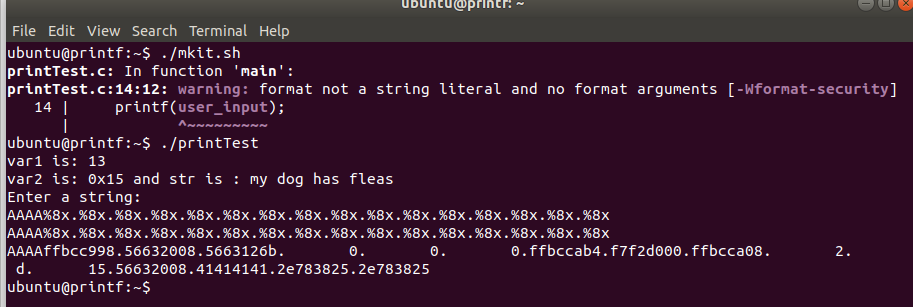
Ran “gdb printTest”, listed out the lines of the program, set breakpoint at line 10 (line where printf statement occurs), and ran the program.



Ran “nexti” until printf call.



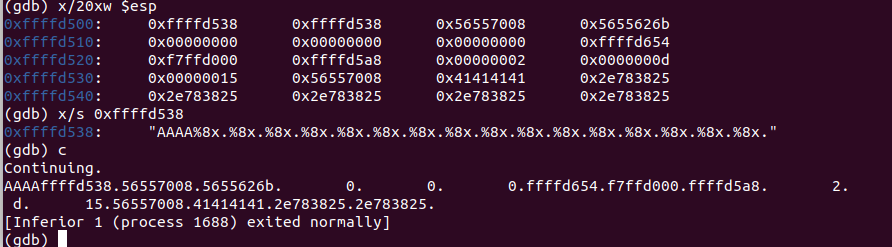
Checked that item on top of stack is the format string. Second top of stack is 13 in hex



Ran printTest again with string input of “AAAA%8x.%8x.%8x.%8x.%8x.%8x.%8x.%8x.%8x.%8x.%8x.%8x.%8x.%8x.%8x.%8x.”. Assorted other text placed in output.



Ran gdb on printTest file. Set breakpoint to line 14, where vulnerable call of printf is located. Stepped to printf call.



Displayed stack content. Confirmed that top of the stack is the given format string. Continued program. It turns out that the program outputs the addresses of the following 16 addresses on the stack. WOW!