Buffer Overflow Example

CS 485: Systems Programming
Fall 2014

Instructor:

James Griffioen

Adapted from slides by R. Bryant and D. O'Hallaron (http://csapp.cs.cmu.edu/public/instructors.html)

Instructions

1. wget http://www.netlab.uky.edu/~griff/classes/cs485/handouts/l4/bufdemo.tar
2. tar –vxf bufdemo.tar
3. cd bufdemo; cat bufdemo.c /* now read bufdemo.c */
4. make
5. ./bufdemo
6. Type “123” when prompted. You should see the expected output:

    Abuf = ‘ ‘

    Bbuf = ‘123’

Exercises

1. Find an input string that will result in the program printing:

   Abuf = ‘ ‘

   Bbuf = ‘1234’

2. Find an input string that will result in the program printing:

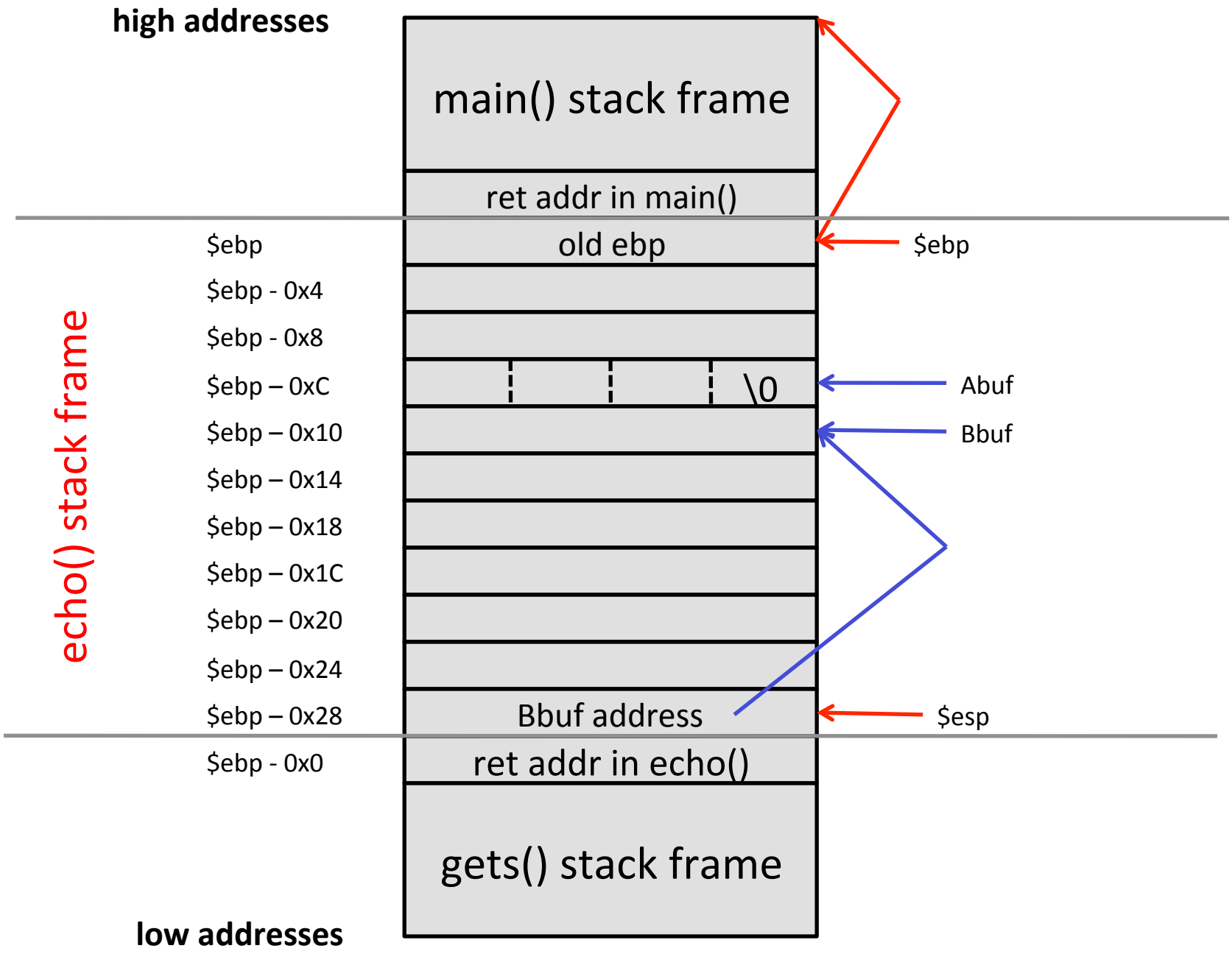
   Abuf = ‘56’

   Bbuf = ‘123456’

3. Give 123456789012345 as input. What is the output? Why?

4. Give 1234567890123456 as input. What is the output? Why?

5. Run the program under gdb and set a breakpoint just before returning from echo(). Run the program and type 12345678901234567890 as the input. Single step the program past the return from echo(). Does the echo() procedure return to main()? Explain what is happening.


high addresses
main() stack frame
ret addr in main()
$ebp
old ebp
$ebp
$ebp - 0x4
$ebp - 0x8
$ebp – 0xC
\0
Abuf
$ebp – 0x10
Bbuf
$ebp – 0x14
$ebp – 0x18
$ebp – 0x1C
$ebp – 0x20
$ebp – 0x24
$ebp – 0x28
Bbuf address
$esp
echo() stack frame
$ebp - 0x0
ret addr in echo()
gets() stack frame
low addresses