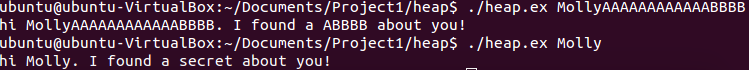
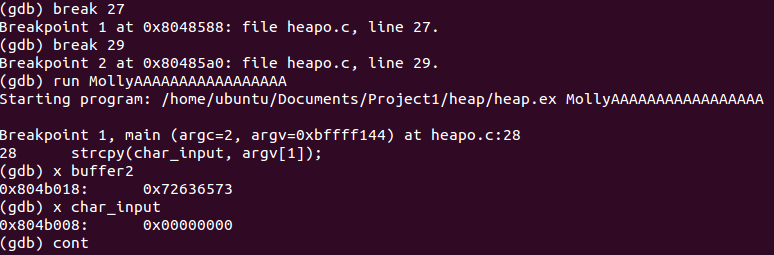
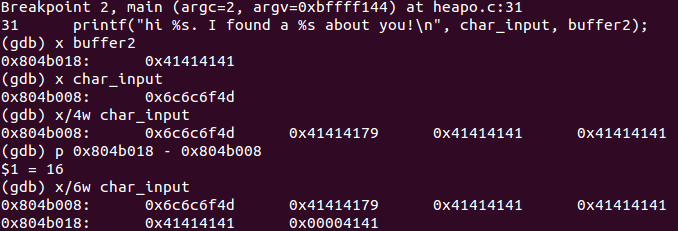
Heap overflow

1. program uses strcpy to cpy input to buffer in heap

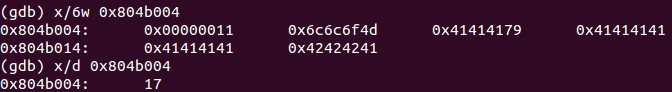
2. heap memory locations: - you can see the malloc zeroed, but not in use space bc we havent strcpy into it yet for char\_input.

3. after strcpy - -calculated space between buffers – 16 bytes – char\_input buffer is 10, buffer is 20 in the program. 



4. before the buffer is overwritten you can see char\_input cleared to zeroes, and then buffer2; the “secret” is stored in the 0x72636573 and “et” in 0x00007465.

5. The 0x00000019 in decimal is 25; this would be metadata for the heap as it shows “secret” in buffer 2 is allocated 20 bytes + 5 bytes for metadata. This explains why in order to overflow Char\_inuput we had to use 17 characters to overwrite the buffer2 heap; metadata for buffer 2 had to be overflown so that the pointer for buffer2 would contain letters. For the first heap, I viewed the address 4 bytes in front of it to attempt to locate metadata.



It looks like our “secret” input memory chunk is over-written by “ABBBBB”