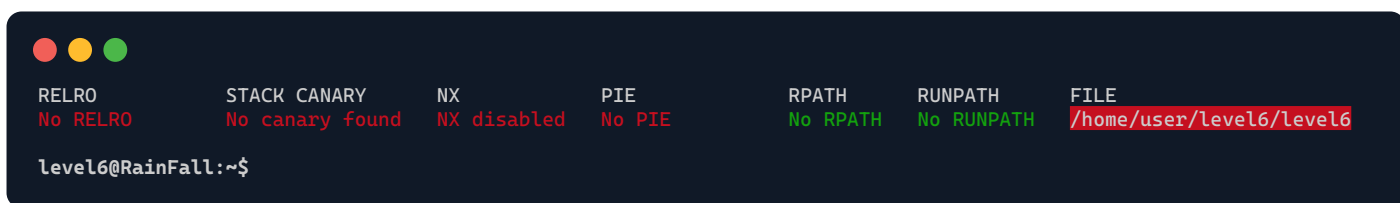


./level6



Decompiled file with *Ghidra*:

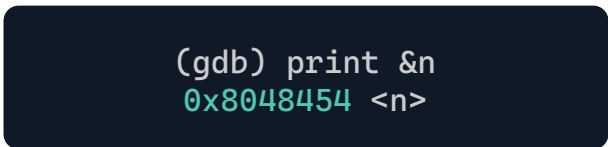


This time, our main function allocates a **buffer** of 64 bytes and also allocates space for a **function pointer**.

The **funcPtr** points to the **m()** function, which currently does nothing.

We need to modify it so that it points to the **n()** function, which will execute the cat command on the *level7.pass* file.

First, we will find the address of the **n()** function:



Since **strcpy** does not check **buffer** boundaries, we can *overflow* the buffer using **argv[1]** and overwrite the **funcPtr** value to point to the **n()** function. Both **buffer** and **funcPtr** are located in the heap, and since **funcPtr** was declared after the **buffer**, they are contiguous in memory.

Because **malloc()** pads out the memory allocated to multiples of 8 bytes, when the **funcPtr malloc(4)** allocates memory, it provides 8 bytes for user data. Before this user data, it reserves another 8 bytes for internal *bookkeeping*, which typically includes *metadata* about the size of the allocation and possibly pointers for managing free blocks in the heap.

Therefore, to reach the **funcPtr** after the buffer, we need to write 64 characters to fill the buffer, then an additional 8 bytes to override the bookkeeping data, before we can overwrite the value of **funcPtr**.

