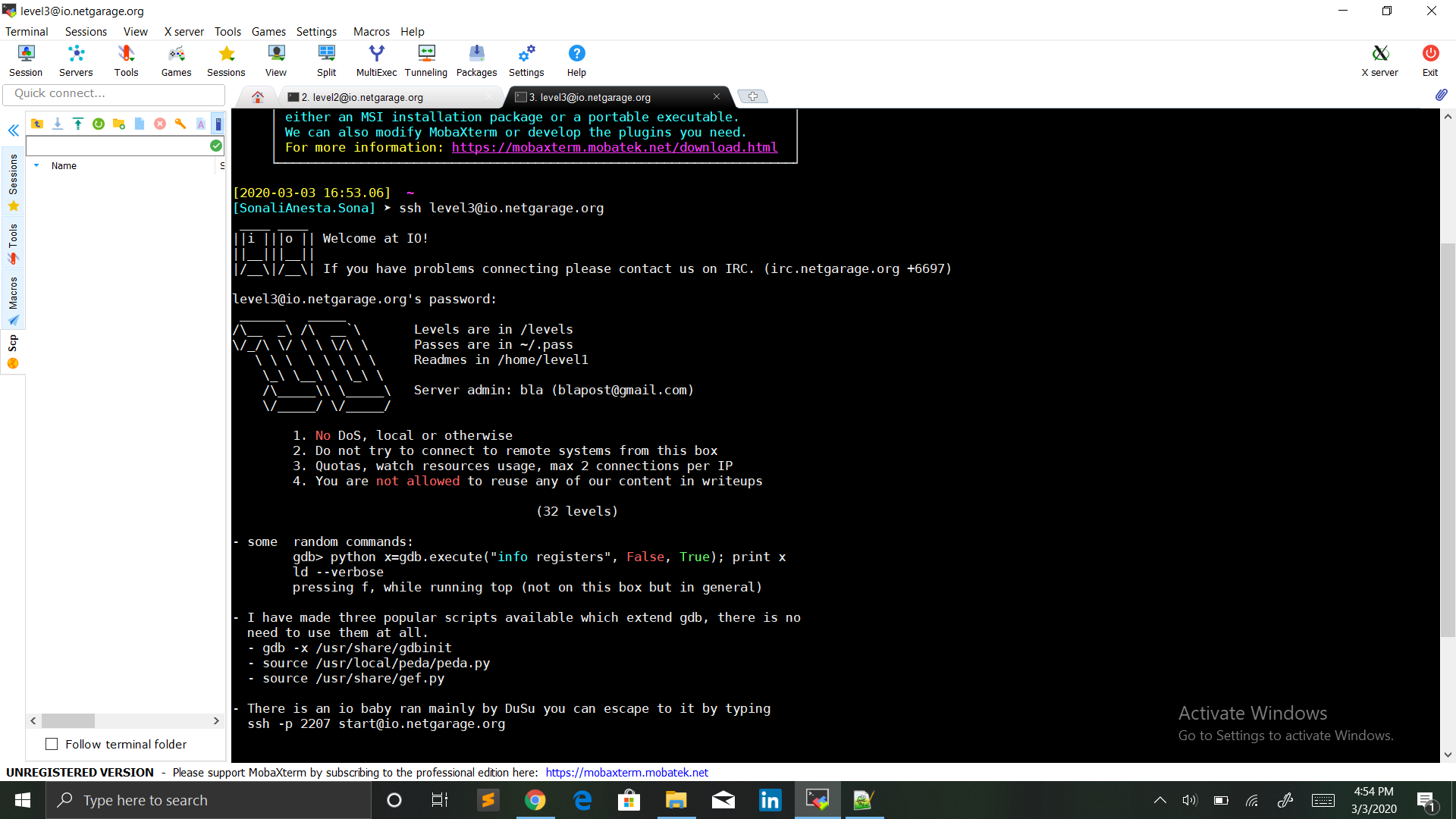
IO.NETGARAGE.ORG

LEVEL 03

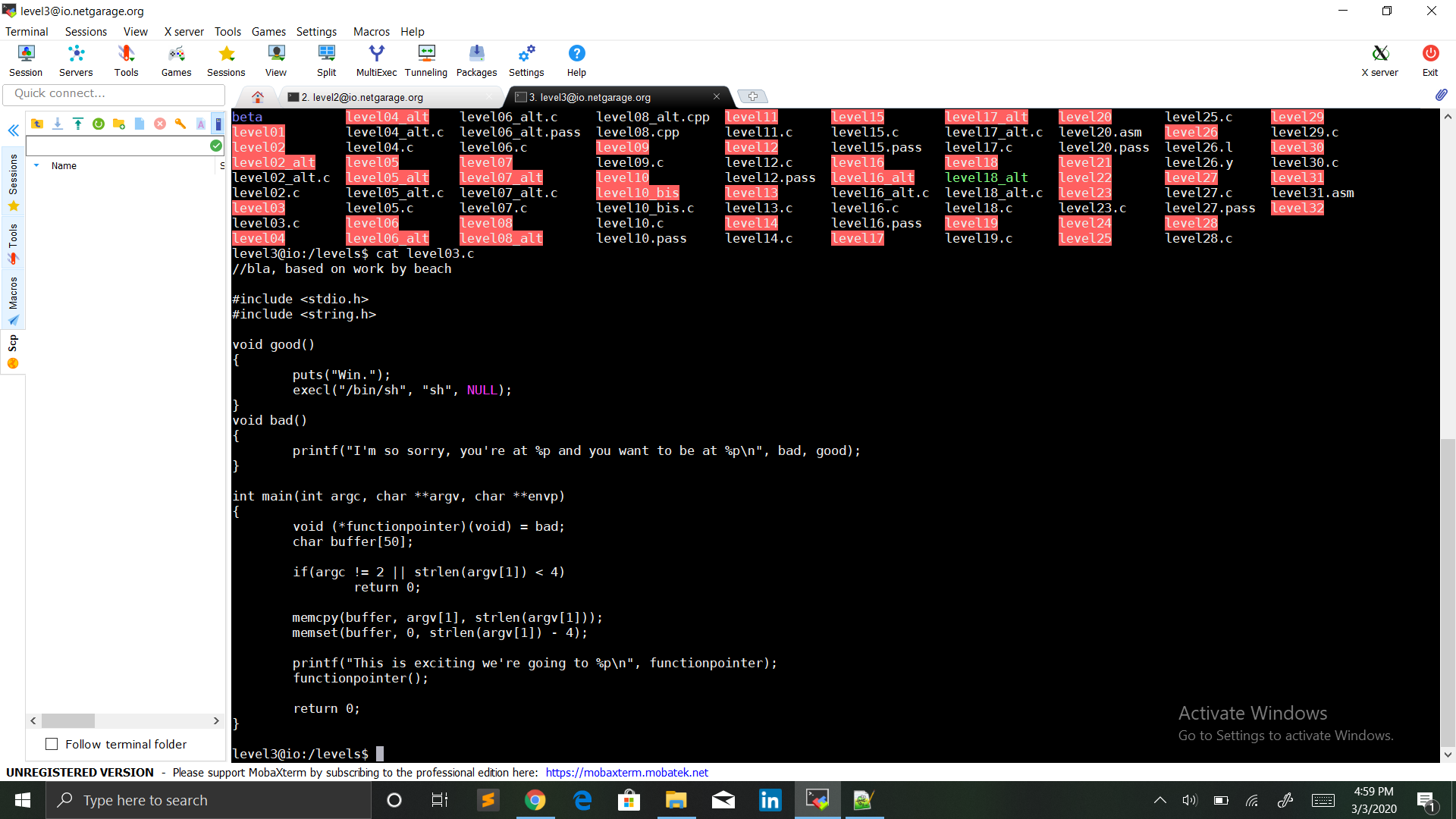
IT17098724

Anesta W.D.S

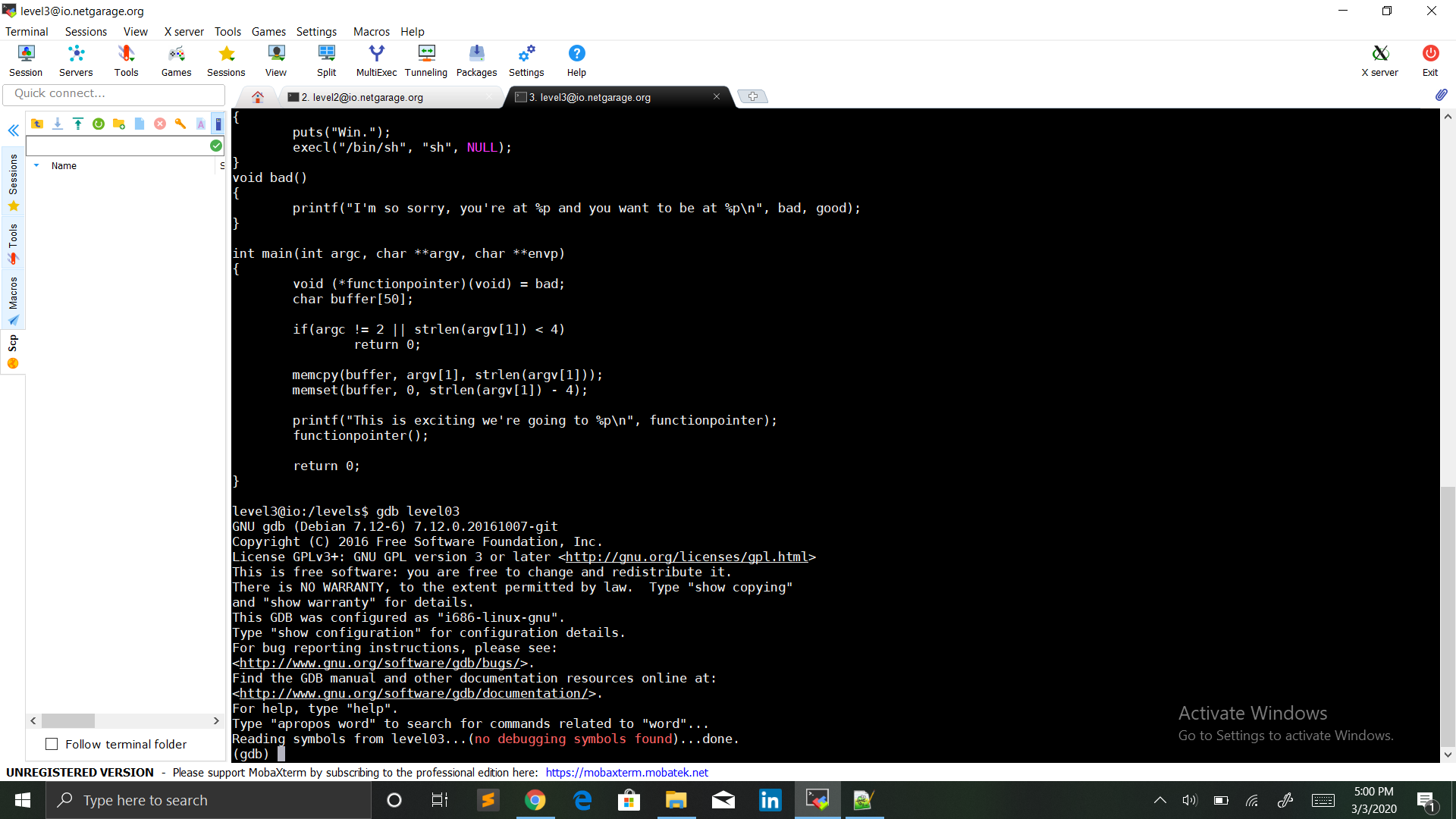
* Using link and password log in to the wargame level 03 as shown below.



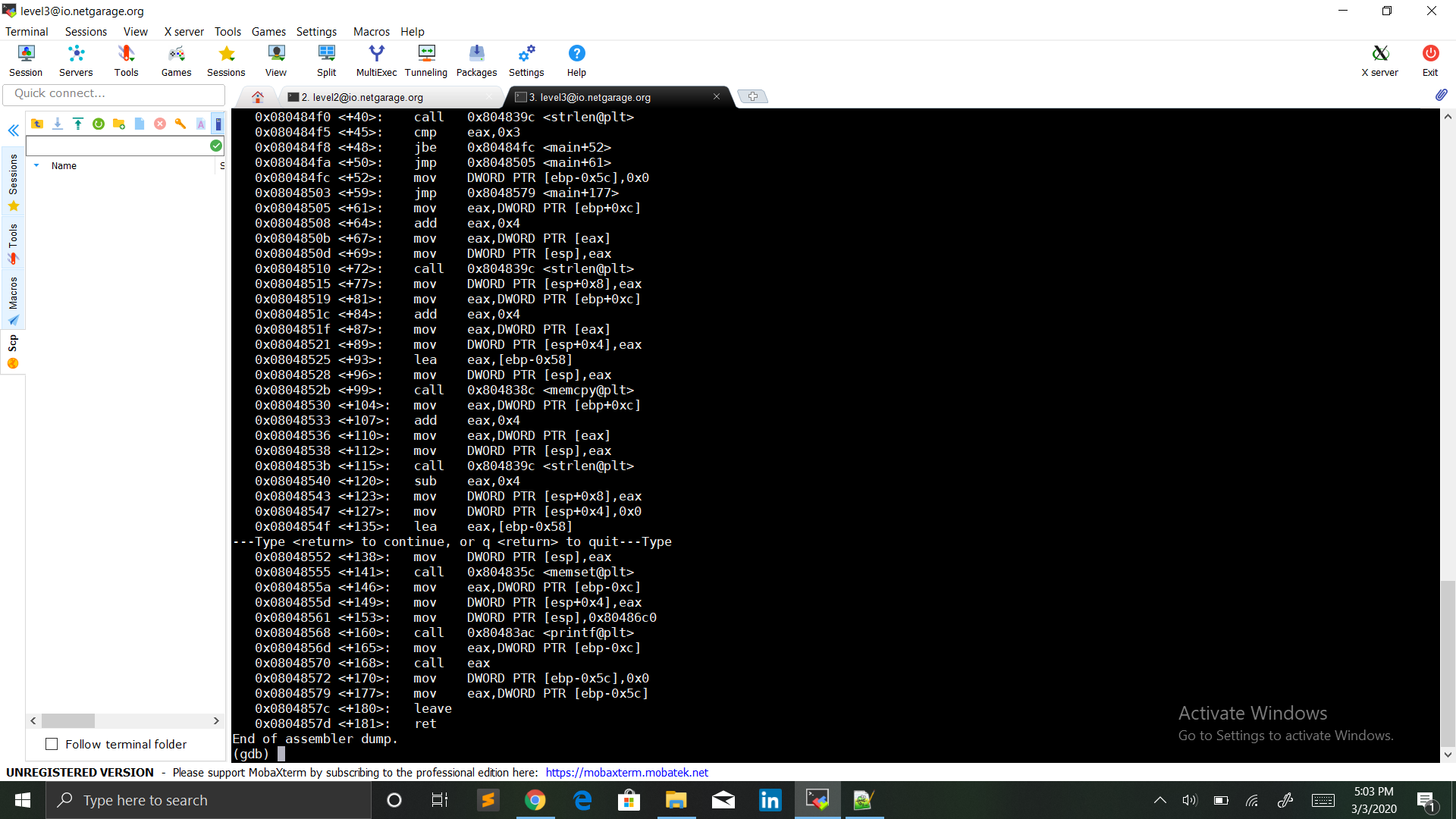
* Open the level03.c file



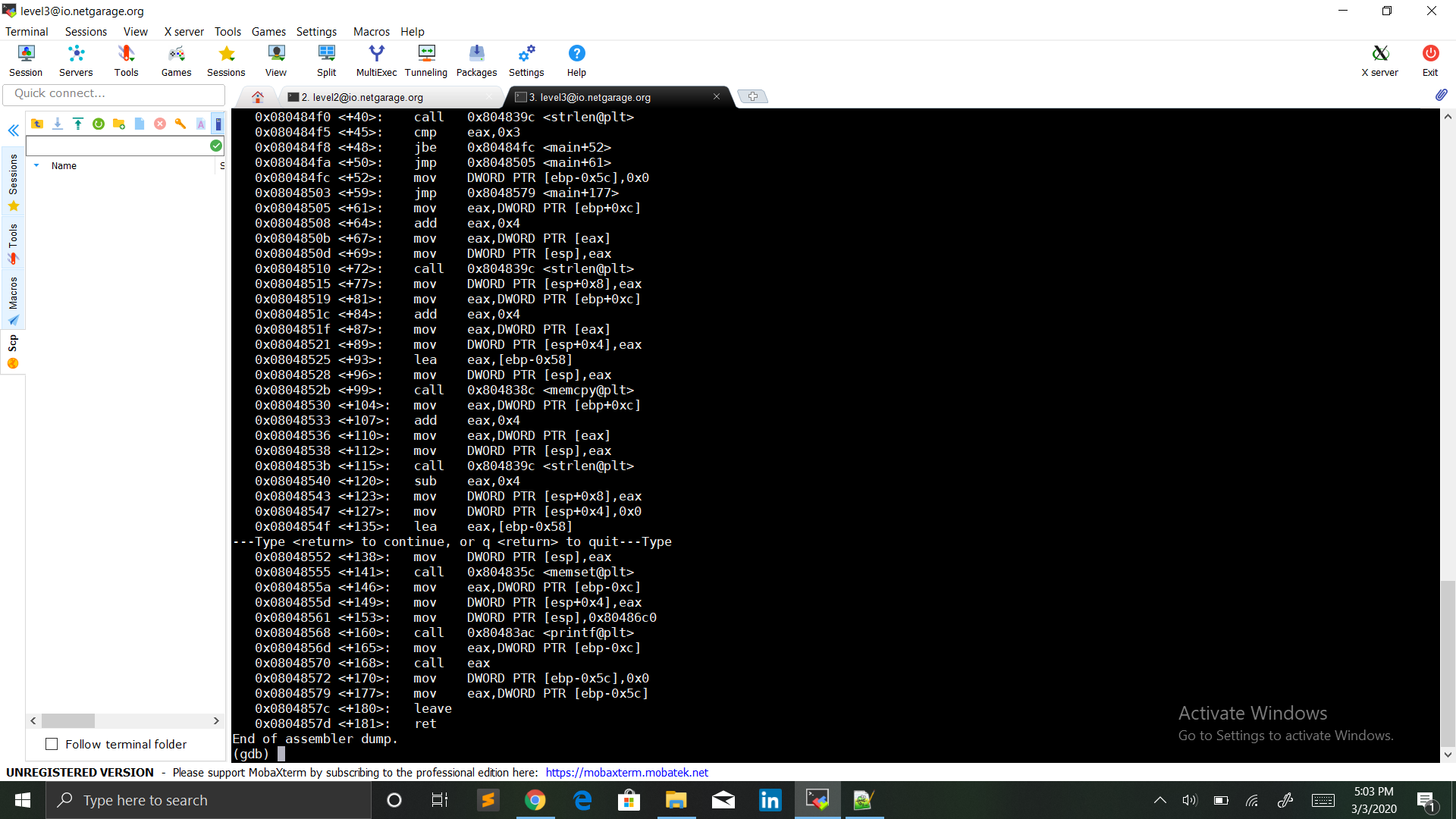
* We can see the source code above, the program is **vulnerable** to **Buffer Overflow** security holes , seen from the memcpy(buffer, argv[1], strlen(argv[1]));conditions where argc != 2 || strlen(argv[1] < 4, in void those who have functions with pointers and store strings in functions bad()
* In the function good(), which later will popup /bin/shand the function is bad()to save the message I'm so sorry, you're at %p and you want to be at %p\n
* Launching the program under **gdb**



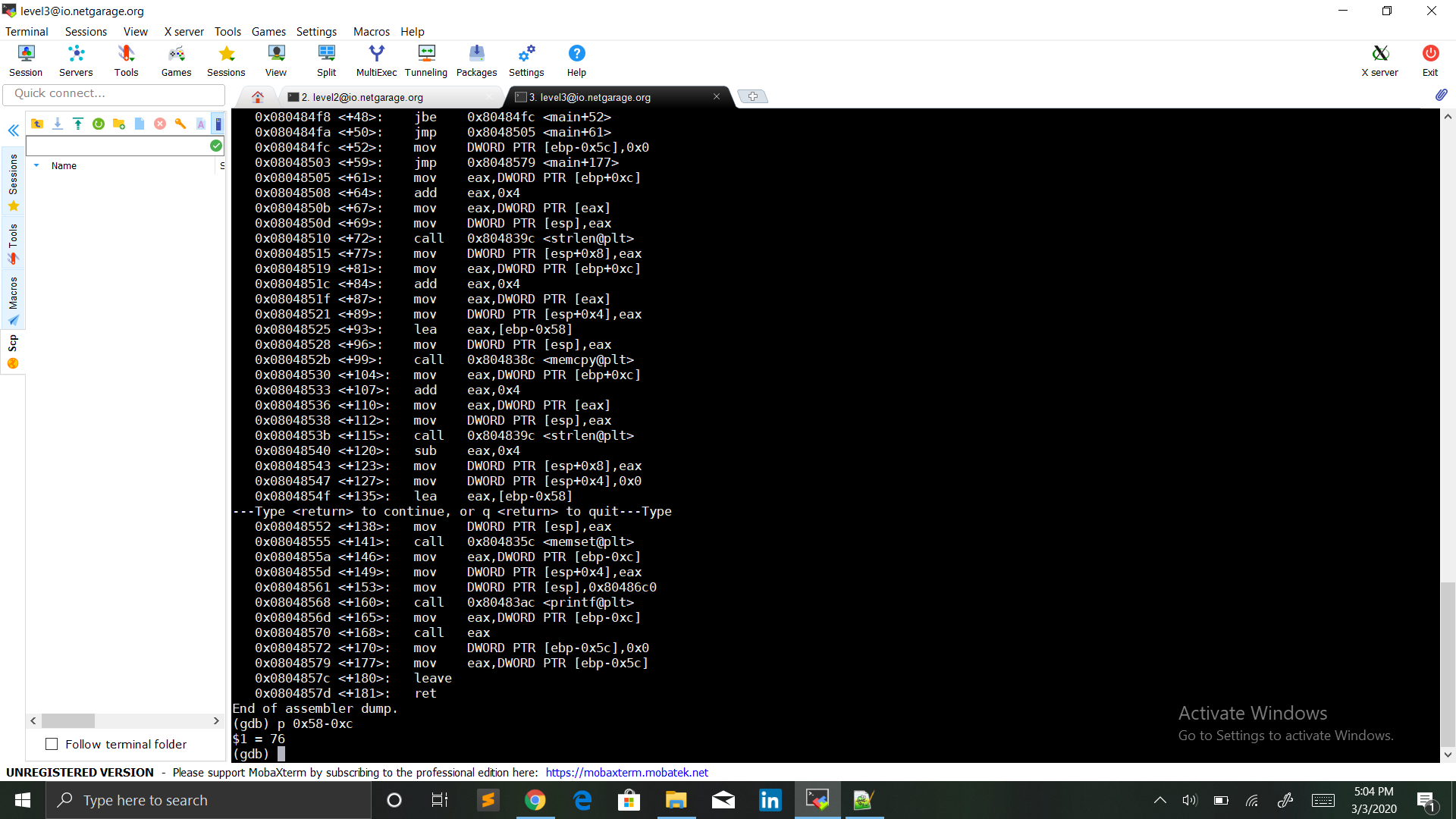
* With **gdb**, we can confirm that argument1 really contains the string.
* So, the function pointer is located at esp+0x4. However, I would recommend you find a reference that uses **ebp** instead of esp. This is because the value of **ebp** is constant in a stack frame whereas **esp** can vary.



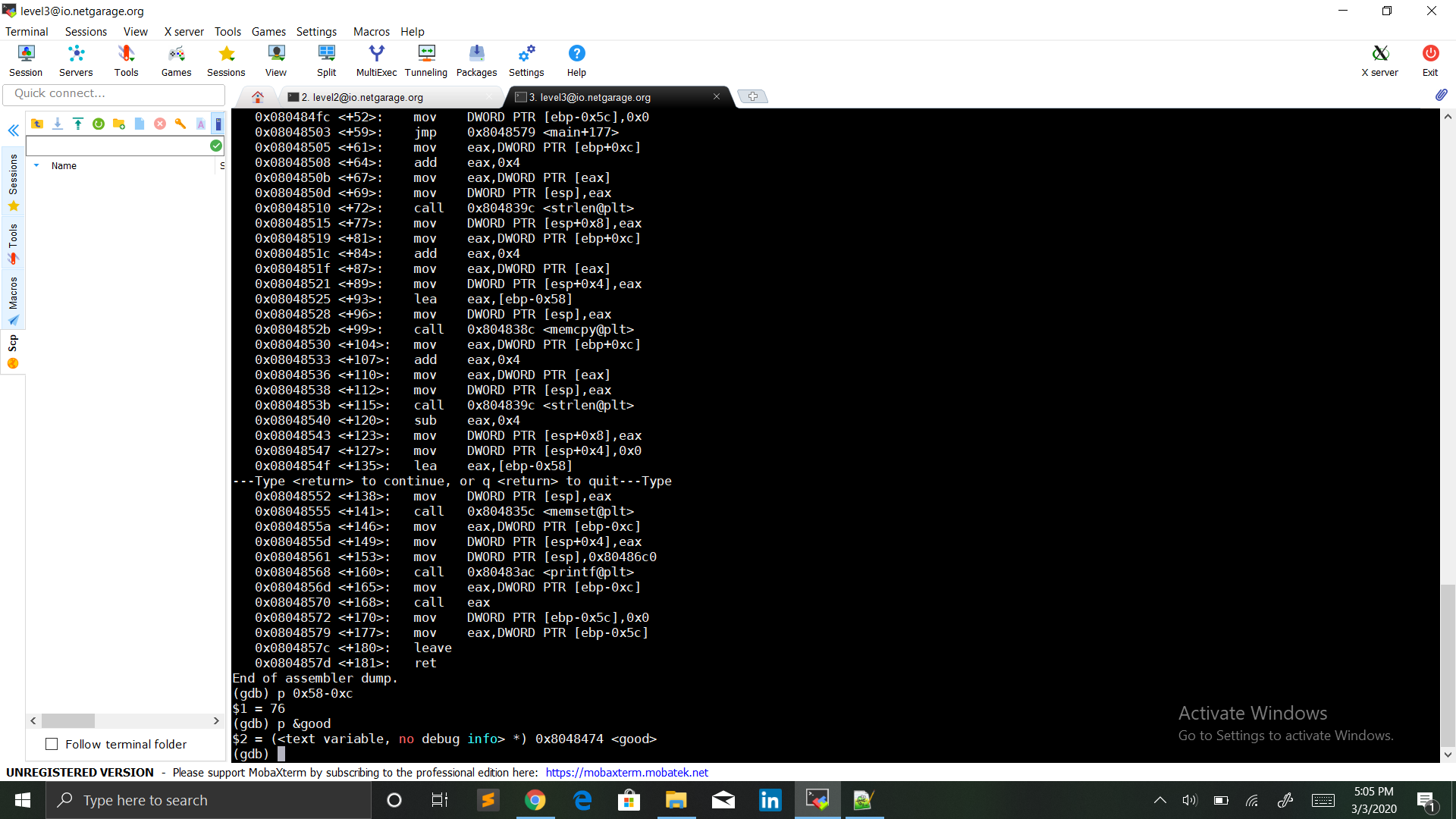
* This is another place that actually references our function pointer. Notice call **eax** From this, we know function pointer is at **ebp-0xc**
* We can do the same to find out where is the buffer.



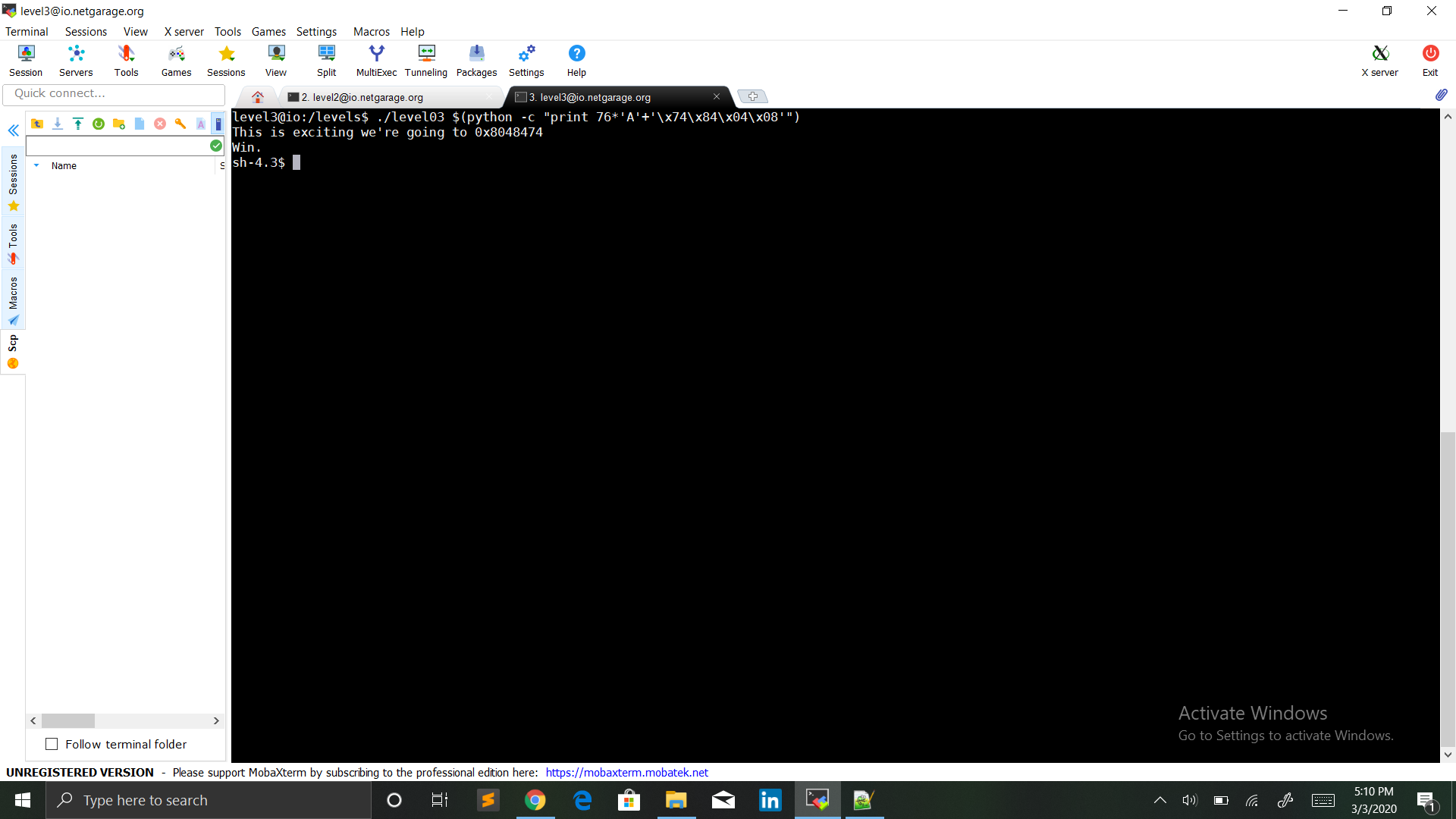
* From the above snippet, the buffer is at **ebp-0x58**
* Now, we just need to find the distance between buffer and function pointer



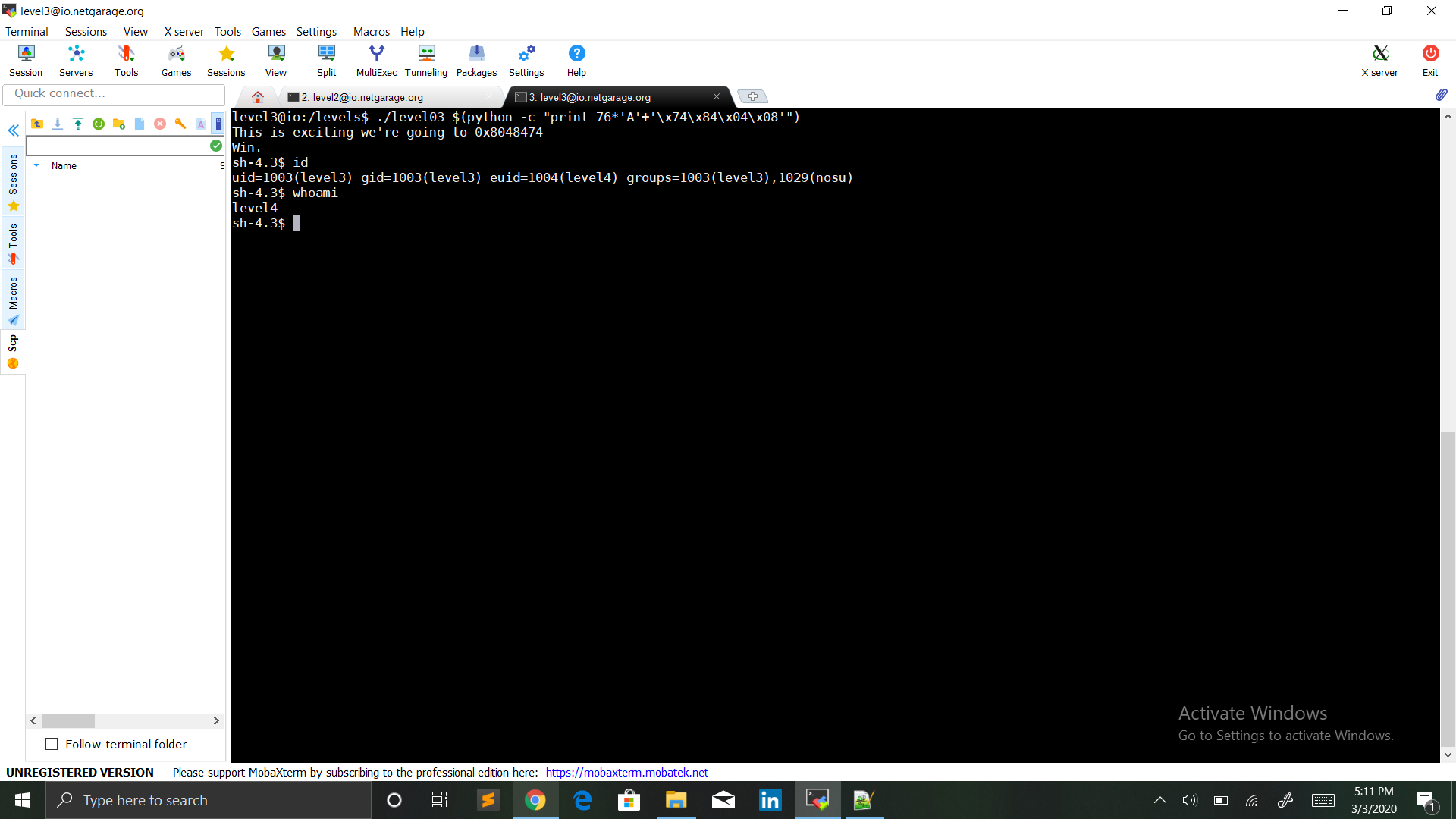
* From the above, we see it actually takes 76 bytes to reach the local variable function pointer instead of 50 bytes.
* The good address is present in the stack at address 0x8048474



* We can override the buffer with arbitrary data. In the current overflow, the last 4 bytes replace the function being executed.
* So, we know the 76 first bytes can be random, and the last 4 should form the address 0x080484a4
* As you can see the address is reversed, it's because the x86 processor is little-endian, so we need to transfer the address backward



* We successfully obtained a shell. We can see that our effective user-id became level 4. Thus, we have access to the password for the next level.



* Level 4 password: **7WhHa5HWMNRAYl9T**