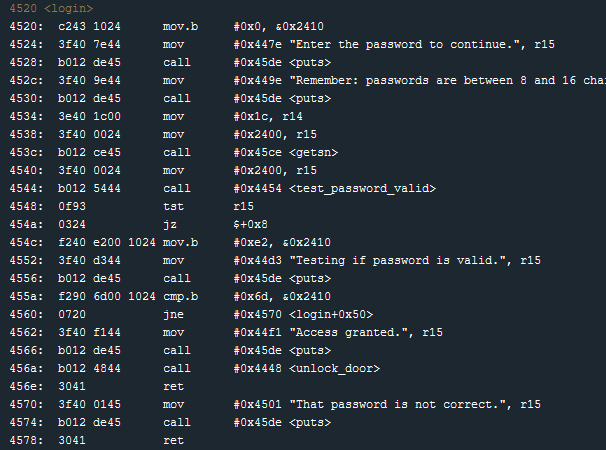
Micro Corruption – Hanoi

As with the past 2 challenges, the first thing to do when starting Hanoi was read the manual. This time they’ve taken some precautions to have no sign of the password inside of the code or memory. This will require a bit of investigating into the assembly to see if I can find a way to crack the lock.

The first thing would be to investigate the main function, but it’s a little short and all it really does is call a function called login. So that’s where I’ll be looking first.



So by this point after doing a few challenges, the first couple lines of code look familiar. Some output is shown to the user, a password is requested, and the memory address of the provided password is stored in register r15. After the password is obtained, there is a call to test\_password\_valid. The following line ‘tst r15’ seems to imply that there is some modification to r15 and its value is being checked for 0. This must be the flag that gets set for a valid password that is mentioned in the manual.

If the value was not 0, the byte 0xe2 is written to memory address 0x2410. If the flags value was 0, there is a jump to the print statement informing the user the password is being tested. After the print statement, there is a check to see if the byte in memory location 0x2410 is equal to 0x6d. This is the key right here. The test\_password\_valid function must set that memory location to 0x6d if the password is valid. Looking at line 4538, I see that the address 2400 is where the user provided password is stored. With only 16 bits between the start of the user provided password and the flag, this looks prime for an overflow. Let’s check that out.

To do this, I’ll set a breakpoint on line 4540 and for the password I’ll provide 17 bits to see if I can overflow that flags value. To make it easy to find the password, I’ll input 17 ‘Z’s. When the breakpoint is hit I’ll look to see what the memory at that location looks like.



Well it’s obvious to see that the memory location 2410 can be written to. So now I’ll try writing the value being compared for, 0x6d, 17 times and see if the comparison will pass.

Entering ‘6d6d6d6d6d6d6d6d6d6d6d6d6d6d6d6d6d’ as the password (make sure to select ‘enter as hex value) will overflow that memory location and grant access to the lock. With our operatives in the building it’s time for the next challenge.