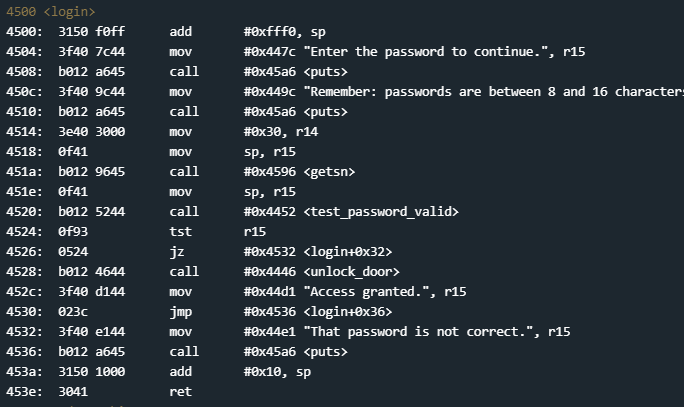
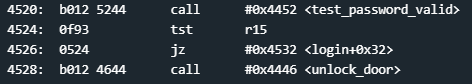
Classic first move, look at <main>:



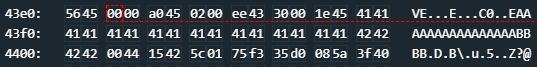
I think this is going to be a normal thing. Now let’s look at <login>:



Well, there is no cmp in sight but we do have a tst of r15 at line 4524. Let’s check out a couple of lines around that tst r15:



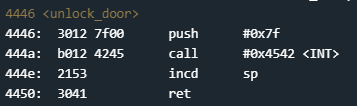
It looks like we call <test\_valid\_password> which should set r15 to a certain value, we test r15 and then, if it is zero, we jump past the call to <unlock\_door>. Hmm...how do we set r15 to the right value? Let’s back up for a second and see where values are being stored into memory:



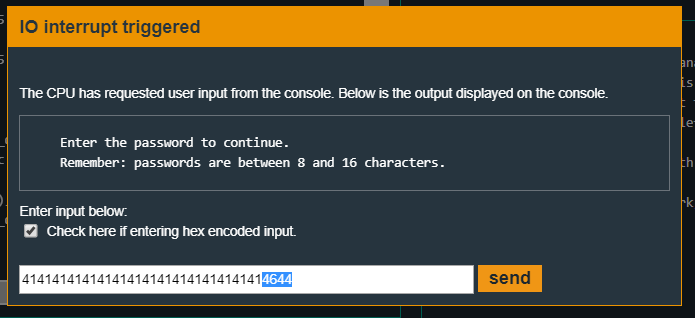
That string of 16 As and 2 Bs were the passed in input. If we continue through the program after we give our input we get this error:



This means that we have overwritten a return address. With this knowledge we could execute a classic stack overflow to redirect code flow to where we want it to go. This is similar to protostar stack0-4. Let’s find the memory location of where the door is unlocked:



The address of the <unlock\_door> function is 4446. Remember that the processor is little endian so we need to input this as 4644. We are expecting a 16 character password, so we will input 16 As and then 4644:



And this unlocks the door! Our first classic overflow challenge!

