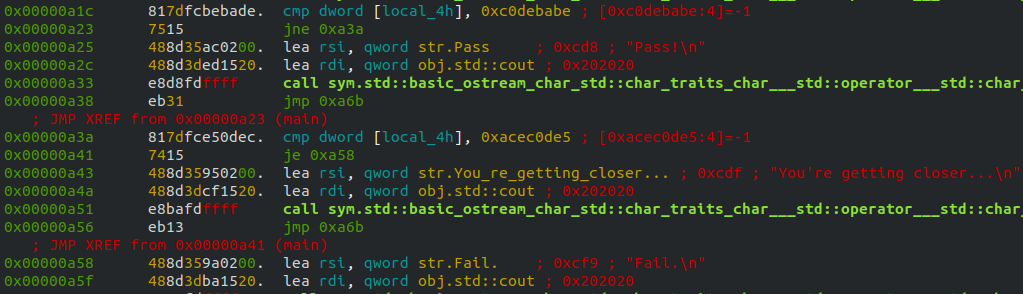
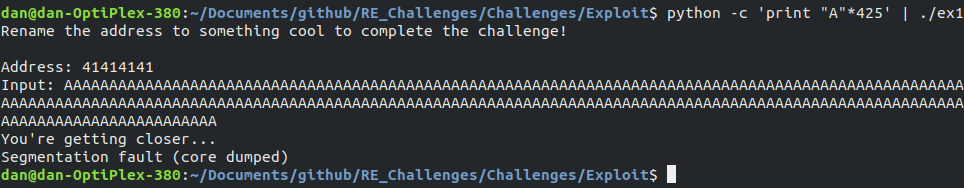
*Ex1*

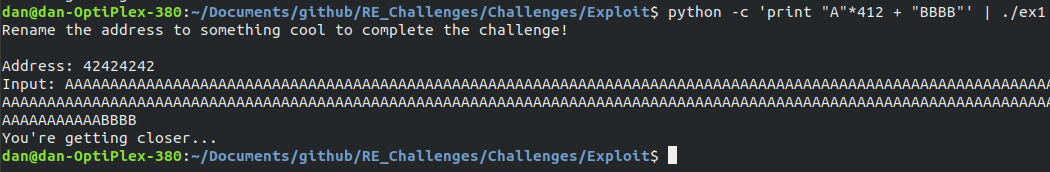
First, take a look at this challenge in radare2 to find the value the program is looking for instead of the current printed address. While you could change the value using static analysis, the point of this challenge is to use a buffer overflow to change it. The commands I used in radare2 were simply **aa** and [**pdf@main**](mailto:pdf@main).



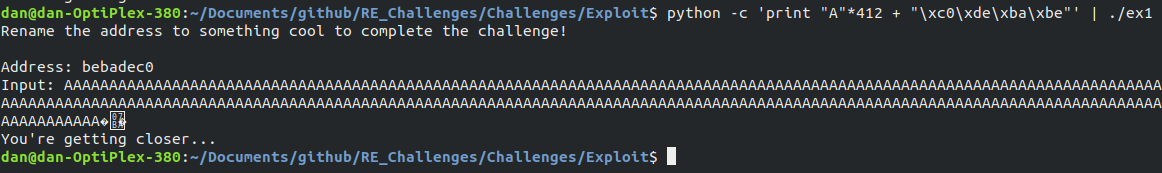
The current address printed to the string and the address it is expecting you to change it to is clearly visible. Now, let’s work on smashing the stack. Instead of running the program and copy-pasting a large amount of values to the input, we can create large strings in a language like perl or python and *pipe* the values into stdin (standard input). Look up terminal piping for more information.



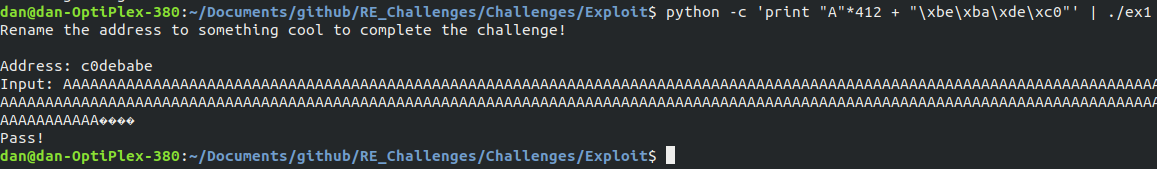
Now that we’ve overwritten into the address, lets pinpoint exactly where the address is:



Note that 41 is the char code for ‘A’ and 42 is the char code for ‘B’. Now, lets pop in the correct value where the B’s are.



Hrm. This appears to be backward… oh, addresses on my computer are little-endian, meaning that we need to feed these bytes in backward!



Mission accomplished.