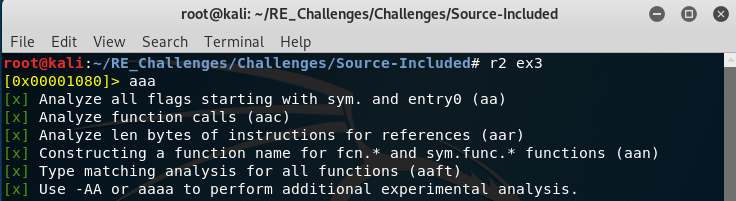
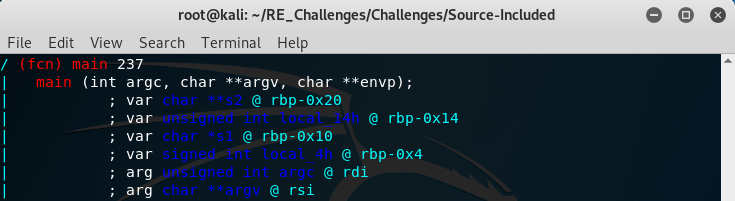
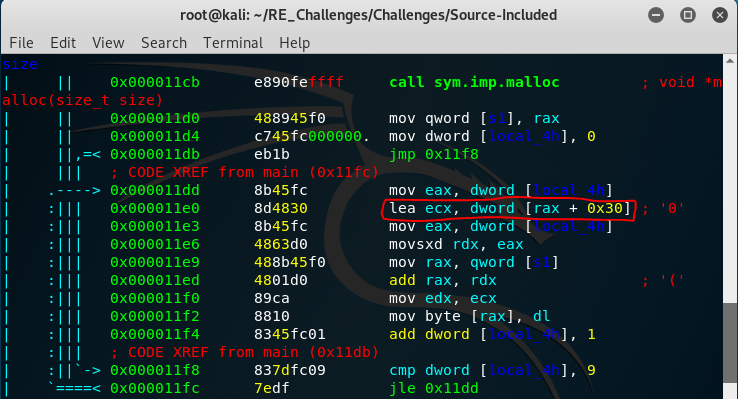
***Ex3***

In this challenge the input string is compared to a string that is dynamically built in memory.

Like always we run radare2 with the binary as its sole parameter and perform some analysis before we begin to dive into the inner workings of the program.

Run the “[pdf@main](mailto:pdf@main)” command that we have grown to know so well and do not forget to always look at the variable listing provided by radare2 before we dive into the instructions of the main function.



As we scroll through we will notice a call to the C function “malloc”. This function takes a size and returns a pointer that points to an address on the heap of the provided size. This is used as a dynamic chunk of memory, ram, for the program to use freely as the programmer specifies. After this we see that it moves this address into the local stack variable “s1” and 0 into the local\_4h variable (our loop counter). From here we see code that is going to add 0x30 to the loop counter and finally move this number into the loop counter index of our malloc’d memory location. Remember the ascii table, 0x30 is ascii ‘0’ and 0x31 is ‘1’, you should now see that we are building a readable string of the decimal numbers 0-9.

